## black hat EUROPE 2019

### X Aleph Research HCL AppScan

DECEMBER 2-5, 2019 EXCEL LONDON, UK

### Simplifying iOS Research: Booting the iOS Kernel to an Interactive Bash Shell on QEMU



#### How current iOS research is done

- Third party iOS emulator on a remote server
- Development fused iPhone
- Off the shelf iPhone jailbroken
- Off the shelf iPhone no jailbreak



#### iPhone panic log

- 1 {"bug\_type":"210","timestamp":"2019-01-31 00:00:31.72 +0100","os\_version":"iPhone OS 11.4.1 (15077)","incident\_id":"C8D49F8C-DBCD-4217-848A-4E3409C968D2"}
- 2 (
- 3 "build" : "iPhone OS 11.4.1 (15077)",
- 4 "product" : "iPhone9,4",
- 5 "kernel": "Darwin Kernel Version 17.7.0: Mon Jun 11 19:06:26 PDT 2018; root:xnu-4570.70.24~3\/RELEASE\_ARM64\_T8010",
- incident" : "C8D49F8C-D8CD-4217-848A-4E3489C968D2",
- 7 "crashReporterKey" : "bbd821d8854956ba61dd35c63d4c25144a58ad9b",
- 8 "date" : "2019-01-31 00:00:28.39 +0100",

```
"panicString" : "panic(cpu 0 caller 0xffffff01b3f8cac): Unaligned kernel data abort. (saved state:
        0xffffffe024a0b290)\n\t x0: 0xffffffe006405507 x1: 0xffffffe0005fb471 x2: 0x000000000000001 x3:
        exffffffelace5alc\n\t x8: exffffffee005fb470 x9: ex0000000000000001 x10: ex00000000001071ae x11:
        exe0000000000001071ad\n\t x12: 0x0000000000000000 x13: 0x000000000000000000000000000000 x15:
        0x00000000000000000 x18: 0xfffffff01b70a014 x17: 0x0000000000000000 x18: 0xfffffff01b2c1000 x19:
        0xffffffe0064054ff\n\t x20: 0x0000000000000000000 x21: 0x0000000000000 x22: 0xfffffff01b825000 x23:
        0xffffff01b2f87e8\n\t x24: 0x0000000000000000000000 x25: 0xffffff0007474000 x26: 0xfffffff0005fb470 x27:
        0xffffffe0003a5c00\n\t x28: 0x00000000000000000000 fp: 0xffffffe024a0b640 lr: 0xfffffff01b2f87a0 sp:
        0xffffffe024a0b5e0\n\t pc: 0xffffff01b301c7c cpsr: 0xa0400304
                                                                           esr: 0x96000021
                                                                                                  far:
        0xffffffe006405587\n\nDebugger message: panic\nMemory ID: 0x1\nOS version: 15077\nKernel version: Darwin Kernel
        Version 17.7.0: Mon Jun 11 19:06:26 PDT 2018; root:xnu-4570.70.24~3\/RELEASE_ARM64_T8010\nKernelCache UUID:
        C50E403D58F345EAD0FECCF458171791\niBoot version: iBoot-4076.70.15\nsecure boot7: YES\nPaniclog version: 9\nKernel
        slide: 0x000000014200000\nKernel text base: 0xffffff01b204000\nEpoch Time:
                                                                                     sec
                                                                                                 usec\n Boot
        0x5c522a5b 0x8008dae2\n Sleep : 0x00000000 0x00000000\n Wake : 0x000000000 0x00000000\n Calendar: 0x5c522c57
        0x0002666e\n\nPanicked task 0xffffffe0003elce8: 27417 pages, 234 threads: pid 0: kernel_task\nPanicked thread:
        exffffffeeeesfb47e, backtrace: exffffffee24aeaa9e, tid: 401\n\t\t lr: exfffffffeb3f96e8 fp:
        0xffffffe024a0abd0\n\t\t lr: 0xffffff01b2e15f4 fp: 0xffffffe024a0abe0\n\t\t lr: 0xfffffff01b313a44 fp:
        0xffffffe024a0af50\n\t\t lr: 0xfffffff01b313dd4 fp: 0xffffffe024a0afb0\n\t\t lr: 0xfffffff01b313c00 fp:
        exffffffe024a0afd0\n\t\t lr: 0xfffffff01b3f8cac fp: 0xffffff0024a0b130\n\t\t lr: 0xfffffff01b3f9dfc fp:
        0xffffffe024a0b270\n\t\t lr: 0xfffffff01b2e15f4 fp: 0xffffffe024a0b280\n\t\t lr: 0xfffffff01b301c7c fp:
        exffffffe024a0b640\n\t\t lr: 0xfffffff01b2f87a0 fp: 0xffffffe024a0b690\n\t\t lr: 0xfffffff01b2fffa0 fp:
        0xffffffe024a0b770\n\t\t lr: 0xfffffff01b306be8 fp: 0xffffffe024a0b850\n\t\t lr: 0xfffffff01b339988 fp:
        0xffffffe024a0b8e0\n\t\t lr: 0xfffffff01b634b84 fp: 0xfffffff0024a0b950\n\t\t lr: 0xfffffff01b654448 fp:
        0xffffffe024a0b980\n\t\t lr: 0xfffffff01b654124 fp: 0xffffffe024a0bab0\n\t\t lr: 0xfffffff01b654e1c fp:
        exffffffee24aebb30\n\t\t lr: exfffffffe1b652ea8 fp: exffffffee24aebc90\n\t\t lr: exfffffffe1b2ec500 fp:
        "panicFlags" : "0x2",
10
```

- "otherString" : "\n\*\* Stackshot Succeeded \*\* Bytes Traced 202128 \*\*\n",
- 12 "memoryStatus" :
  - {"compressorSize":0, "compressions":0, "decompressions":0, "busyBufferCount":0, "pageSize":16384, "memoryPressure":false
    ,"memoryPages":
  - {"active":46523,"throttled":0,"fileBacked":69049,"wired":32283,"purgeable":1282,"inactive":14971,"free":48793
    ,"speculative":29929}},



#### **Jonathan Afek**

- Aleph Research group manager at HCL/AppScan
- 15 years of experience in security research and low level development including vulnerability research, Linux kernel, storage systems, WiFi systems and FW, security systems and more.



### iOS on QEMU work done by @zhuowei (Worth Doing Badly)

FEHEL FIRLACK HATEVENTE



#### QEMU

From Wikipedia, the free encyclopedia

QEMU (short for Quick EMUlator)<sup>[2]</sup> is a free and open-source emulator that performs hardware virtualization.

FRHEU VIBLACK HATEVENTS



#### Past Research – Worth Doing Badly (@zhuowei)

- Chosen version is iPhone X iOS 12 beta 4
- Extracted the kernel image and the device tree from the software update package
- kernel, device tree and the kernel boot arguments were loaded in memory
- iOS RAMDisk was loaded in memory
- UART serial output was achieved
- Kernel was booted
- Launchd was executed



#### **Past Research – Worth Doing Badly (@zhuowei)**

BSD root: md0, major 2, minor 0
apfs\_vfsop\_mountroot:1468: apfs: mountroot called!
apfs\_vfsop\_mount:1231: unable to root from devvp <ptr> (root\_device): 2
apfs\_vfsop\_mountroot:1472: apfs: mountroot failed, error: 2
hfs: mounted PeaceSeed16A5327f.arm64UpdateRamDisk on device b(2, 0)
: : Darwin Bootstrapper Version 6.0.0: Mon Jul 9 00:39:56 PDT 2018; root:libxpc\_execu
boot-args = debug=0x8 kextlog=0xfff cpus=1 rd=md0
Thu Jan 1 00:00:05 1970 localhost com.apple.xpc.launchd[1] Notice>: Restore environm



#### **Goals of our project**

- Booting iOS on QEMU with no kernel patches
- Supporting hardware (disk, display, touch, sound, multiple CPUs, Interrupt controllers, etc...)
- Supporting different iOS versions
- Conducting iOS security research
- Learning about iOS and QEMU internals



#### **Status of our project**

- Booting Secure Monitor and the kernel (unpatched)
- Executing a user-mode app over launchd
- Running an interactive bash shell on an iOS kernel on QEMU
- Supporting only on iOS 12.1 for iPhone 6s plus

#### + iOSonQEMU

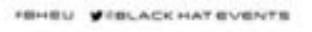
-JORNANNE.

x



#### Agenda

- Past public research on iOS on QEMU
- iOS kernel boot process
- Execution of non-apple executables with Trust Cache
- Bash execution with launchd
- UART interactive I/O
- Next steps



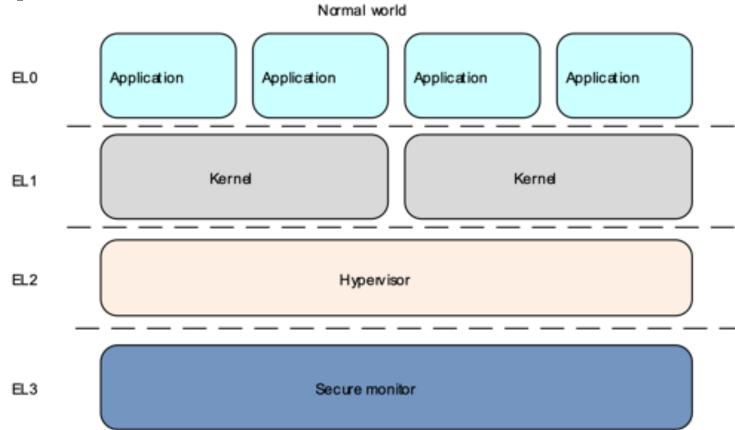


- Start booting the kernelcache code in EL1 as done by @zhuowei
- Crash on SMC instruction (Secure Monitor Call)

x0	0x800		2048	x1	8x478a5888	1191858176	x2	exe	•
x6 > >	0xff		255	x7 ->	0xad0	2768	x8	0x40000000	1073741824
12	0x1		1 10	x13	0x1800000	16777216	x14 24	8x3000000 11	58331648
x18	8x8		0	×19	0xffffff0075d1000	-68595937280	x20	0xfffffff0070a5088	-68601360248
24	0xf4248		1000000	x25	0xffffff00707d000	-68601524224	x26	0xfffffff007078440	-68601543616
30	<b>B</b> xfffffffee	31.72.57	-68600236896	50	0xfffffff00767fe10	0xffffff00767fe		0xfffffff0070a7d3c	exffffffee7e
WFR6_EL1_RES	ERVED 8x8		0	MVFR4_EL1_RESE	EVED Bx8	0	ID_AM64PFR1_EL	1 0x0	0
ID_AA64PFRS_E	L1_RESERVED @		ě	ID_AA64PFR6_EL	1_RESERVED @x@	÷		1_RESERVED @x0	÷
ID_AAG4DER3_E	1_RESERVED @	a)	io an	ID_AAGAAFR1_EL	1 0x0		TO_AAGAAFR2_EL	1_RESERVED @x0	
ID AA64ISARS	EL1_RESERVED 4	buð	e an	ID_AA64ISAR3_E	L1_RESERVED 0x0		ID_AA64ISAR1_E	L1 8x8	ě.
ID_AAGAAFRO_E			ě.	ID_AAGAMAFR1_E				L1_RESERVED 0x0	
	EL1.RESERVED	huði	ě.	ID ANG AMERICA E		4388		L1.RESERVED 0x0	
REVIDE_EL1	8x8		ě.	SCTLR	8x3454593d	877943181	ACTUR_EL1	0x0	ě.
SCTLR_EL2	8x8			HSTR_EL2	8x8	0	CPTR_EL2	8x8	
OCR_EL2	8x8		ě.	MDCR_EL3	0x0		SCTLR_EL3	0x30d5180d	819271693
PMCR_ELØ	0x41000000		1090519040	PHONTENCLR_EL0		•	PHONTENSET_EUR		0
PMCCNTR_ELØ	exe		0	PMCEID1_ELØ	0x0		TTBR0_EL1	8x18008c7c48000	2814777892536
MATR EL1	8x44f08bb44	ff	4737361200383	TTBR0_EL2	0.0		TOR_EL2	8x8	0
TTBRØ_EL3	8x418998c984		279172923392	MATR_EL2	0.0		TCR_EL3	8x1o511	107793
L2CTLR_EL1	0x0		0	L2ECTLR_EL1	0x0		DACR32_EL2	0x0	0
SP_ELØ	8.8		ě.	VBAR_EL2	0.0	i i	SP_EL1	0xfffffff007688000	-68595187712
SPSR_TRO	A-4		ē.	SPSR ART	exe		FPCR	8x8	0
DBXFFFFFFFB		ret	#0x11						
Bxffffff	87807644	.inst	0x00000000 ; unde	fined					
BATTITI			0x00000000 ; unde						
Bafffffff			0x00000000 ; unde						
Beffffff			8x80000000 ; unde						
a.rrrrrrr			0x00000000 ; unde						
Bxfffffff	87847458		0x00000000 ; unde						
Beffffff	070a7d5c		8x80000000 ; unde						
Bafffffff			0x83828188 ; unde						
Beffffff			8x87858584 ; unde						
Barreren		add	n8, n8, n10, 1sl						
Barrereren B	278a7d6c	.inst	0x8f8e8d8c ; unde						
Berrrrrr			0x00000000 ; unde						
Battittio			0x00000000 ; unde						
Bxfffffff			0x00000000 ; unde						
Bufffffff			0x00000000 ; unde						
Bxfffffff			x1, x1						
Bxfffffff			0xffffffff0070s7s2	M // h. first					
Battittit			0xfffffff0070g7de						
	er els relet	812	x29, x30, [sp, #-	1011					
Bxfffffffb Bxfffffffb	0.00-0-000	BOV	x29, sp						

FEHEL VIBLACK HATEVEN





from https://developer.arm.com/docs/den0024/a/fundamentals-of-armv8

F 551-4 10

BLACK HAT EVENTE



- Secure Monitor Loads at boot in EL3
- It resides in a secure memory location inaccessible from EL1 (kernel code)
- It services SMC calls from the kernel (similar to how system calls from user apps to the kernel are serviced)
- It is responsible for KPP (Kernel Patch Protection) in our system



• The kernel needs a secure monitor to service its SMCs





# Any ideas?

FEHEU FIBLACK HATEVENTS



- iPhone X uses KTRR (hardware mechanism to prevent patches) and no longer uses a Secure Monitor for KPP (Kernel Patch Protection)
- Loading the Secure Monitor image for iOS 12.1 for iPhone 6s plus in EL3 and start executing
  - Loading the image at its preferred address (secure memory) with the image's boot args at the next page and start execution at the entry point in EL3



- Loading the Secure Monitor image for iOS 12.1 for iPhone 6s plus in EL3 and start executing
  - Data abort for trying to parse the kernelcache Mach-O header to decide which areas need which protections





_	.i.			Base Base	LAB_ffff00410000550c		132	DAT_ffff004100013230 = 1Parm2;
					Con the second second		133	DAT_TTTT004300013250 = DAT_TTTT004100013240;
	1.		1.00		#5234 XREF [1]		134	DAT_FFFF804380013258 = DAT_FFFF0041000132481
				str	x8, [x27, #0x18]==0AT_ffff004100013248			LA8_ffff00410000523c:
				acr.	LAD_fff100430000558c		136	pp///ar8 = DAT_ffff004500013248;
					CAR_TTTTRPC200003305			
					#523c XPEF [2]		137	<pre>if (I+piNar13 as 0-101 00 ipports = 0.01 iffration100013240, +[longlong +)[piNar1 pine 1 = piNar29 + 2;</pre>
	-+		0.46	,11110641068	esesc Aner (e)		139	INARS - FUN_FFFFF004100000602x[pimar1,"TEXT_EXEC", 0x101:
				Mr	w8, (x23)		540	- ((rear = 0) ]] (iver = FUR_ffff0045000662(fer 1,PLK_TEXT_EXEC., #
					w8,49x19	0-1	541	second and the second sec
				cmp	LAB_ffff00430000550c		542	puVar7 = =[ulonglong ==](piVar19 = 6];
				bune ldr			50	puvarie = =(ulonglong +=)(pivarie = 6); puvarie = =(ulonglong +=)(pivarie = 6);
					x0, [x2), #0x20]			
		0438008524c 08 18 0 04380085258 fc 22 0		che	x8,LA8_111100410000558c		344	ppsWar8 = DAT_ffff004100013230 + 11
				add	x28,x23,#0x8		145	DAT_ffff004100013230 = DAT_ffff004100013230 + 4;
				BOV	w2,#0x10		546 547	<pre>ppuVar8 = puVar7:</pre>
				mov.	x0,x28			ppsWar12[2] = psWar18;
		0430000525c e1 e3 1		BOV	x1=>sTEXT_EXEC_ffff004500007064,x20		148	<pre>ppuWar12[3] = {ulonplong =}0x2;</pre>
	- 1			ы	FUN_ffff804380006e2c		549	1f (puVar7 < DAT_fff1004300013260) {
1.1	11			che			250	DAT_fffreedbeet3368 = puVar71
		04380005268 e2 e3 1		mov.	w2,#2x10		151	3
		0430000526c e0 03 1		BOV	x0,x28		152	<pre>uVar11 = [[langlong]puVar10 + [longlong]puVar7) - (longlong)OAT_ffff004100</pre>
		04300005270 e1 03 1		BOV	x1=>sPLK_TEXT_EXEC_FFFF994589887878;		253	if (0AT_ffff004300013268 < uNar133 {
	-	04300005274 ee 05 0		84.	FUN_FFFF80438000662c		254	DAT_ffffee4100013268 = wWar111
	57	04300005278 =0 88 0		che	.LAB_ffff00410000538c		155	3
	- 1	0430000527c e2 e3 1		BOV	w2,#8x18		156	if (DAT_ffff004300013248 == (ulonglong ==)8x0) {
	: 1			mov.	x#_x28		257	<pre>eppuVar12 = {ulonglong =}0x8;</pre>
		04380005284 e1 ef 0		adr	x1=>sTEXT_ffff804380087888,-0xffbeff			LA8_ffff0841888854#8:
	-	04300005288 1f 20 0		nop			159	ppuVar8 = 0ulonglong ++042AT_ffff004500013248;
		0430000528c e8 05 0		14	FUN_1111004300006e2c		560	3
1.53	11	04380005290 +0 00 0		che	.LA8_ffff00410000540c		261	else (
111	11	04380005294 e2 e3 1		BOV .	w2,#2x18		362	ppuVarB = Gulonglong ++08x8;
1.1.1	11			BOV	x0,x28		563	ppuVar5 = 0AT_ffff004100013240;
111	1.1	0430000529c 41 ef 0	8 78	adr	x1=>s PRELINK TEXT ffff804180007887		264	60 f



#### Kernel address space grows this way ->



FEHEU FIBLACK HAT EVENTS



- The kernel needs a secure monitor to service its SMCs
- The secure monitor requires the *base address* boot arg to point to the kernel Macho-O header

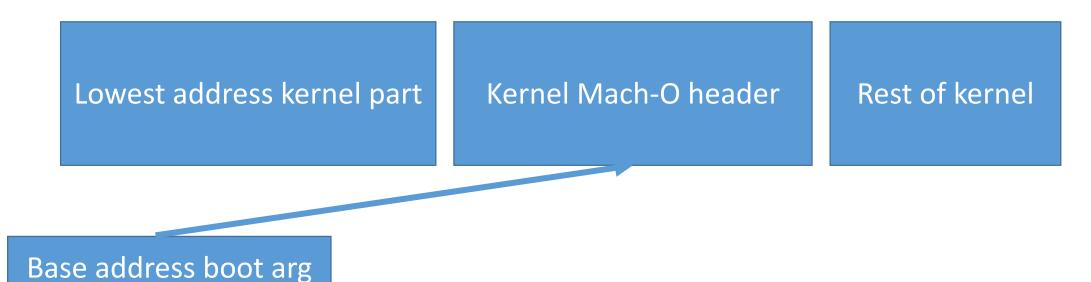


# Any ideas?

FEHEU FIBLACK HATEVENTS



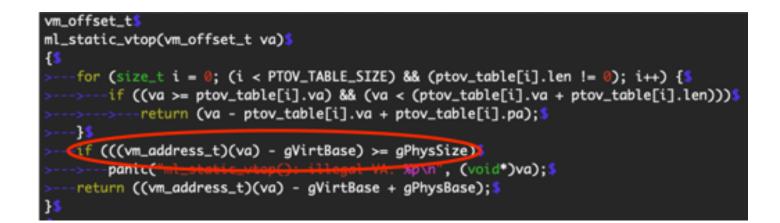
#### Kernel address space grows this way ->



FEHEL FIBLACK HAT EVENTS



- Loading the Secure Monitor image for iOS 12.1 for iPhone 6s plus in EL3 and start executing
  - Tried many different solutions such as changing the base address to the loaded Mach-O header address (above the lowest loaded section/driver)





- The kernel needs a secure monitor to service its SMCs
- The secure monitor requires the *base address* boot arg to point to the kernel Macho-O header
- The *base address* boot arg needs to point to the lowest kernel address in order for the kernel to operate properly

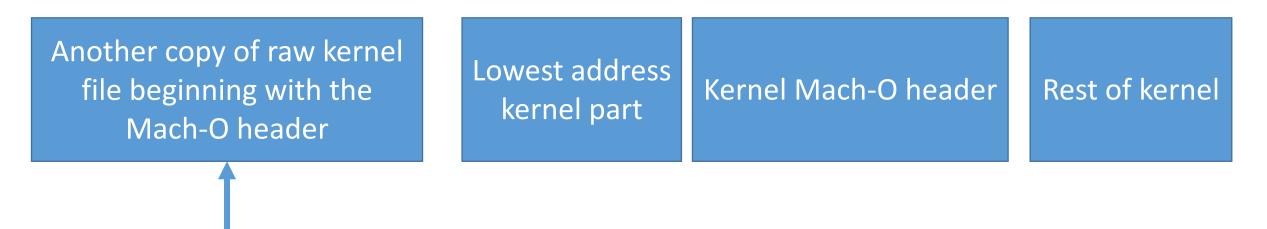


# Any ideas?

FEHEU FIBLACK HATEVENTS



Kernel address space grows this way ->



Base address boot arg

FEHEU FIELACK HATEVENTS



## And it works!

FEHEU FIBLACK HATEVENTS

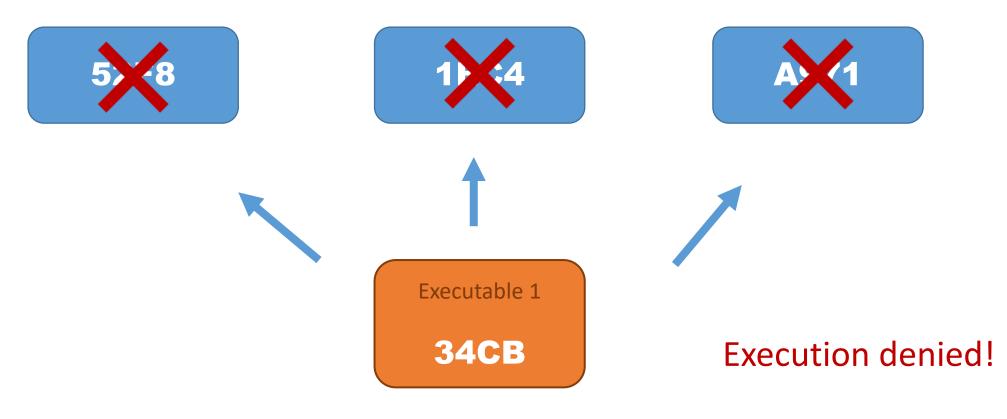


#### Agenda

- Past public research on iOS on QEMU
- iOS kernel boot process
- Execution of non-apple executables with Trust Cache
- Bash execution with launchd
- UART interactive I/O
- Next steps



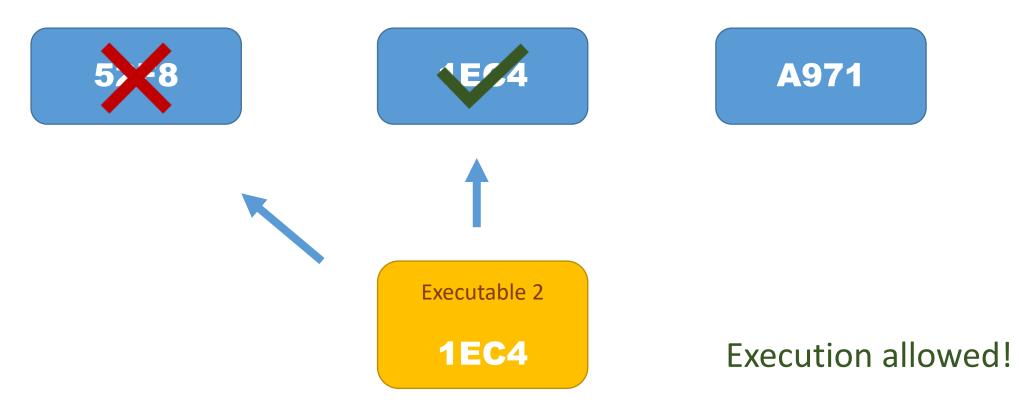
#### Trust Cache Executables Hash List



FEHEL FIBLACK HATEVENTS



#### Trust Cache Executables Hash List



FEHEU FIBLACK HATEVENTS



- iOS has 3 different types of trust caches
  - A list of hardcoded hashes approved in the kernelcache
  - A dynamic trust cache that can be loaded at runtime from a file
  - A static trust cache in memory pointed from the device tree





• Top level CoreTrust validation where execution is decided

	101 8t 99 90 800400 to:	WE, MERR, LAG_TITTTTTNNDSDAT/4	44	plStack272 = (longlong +)&v0	
: .	ff0063e4eec e0 5b 40 f9 Ldr	x8,[sp, #8x00]	67	War7 = FUN_ffffffffffffffffffffffffffffffffffff	
	f19963e4e19 fc 93 95 32 mov	w28,#2×4000000	44	if ((War7 6 1) == 0) {	
1.00	ff0063e4ef4 80 01 00 b4 cbz	x8,LA8_fffffff8063e4f24	1 102		
	ff0063e4ef8 e1 ef #2 91 add	x1, 10, #9xbb	1 1 1 2 2	FUX_FFFFFFFBB63e5e58	
	fribbiGelefic e2 eb #2 90 add	x2,50,00xba	70	(uParm1, uParm8, uParm9,	
	friee63e4fee es ra ff s7 bl	FUN_11111110065e2124	71		
	ff8863e4f84 68 82 48 b9 ldr	v0, [x29]	72		
				for validation: No"	
		v8, [x29]	73		
	ff8863e4f38 e8 ef 42 39 Ldrb	wf, [sp. #exbb]	24	gota LA8_fffffff8865e4f94j	
	ff0063e6f34 1f 01 00 71 cmp	vil, #0x8	75	2	
	ff8863e4f38 88 99 88 52 mov	vil. #0×0000	76	sWar5 = 8x4000000;	
		v0,#0x400, LSL #16	77	if (plStack272 t= (longlong =)#x#) {	
	ff0063e6f20 9c 03 00 1a csel	v01,v01,v0,eq	78	<pre>www.setup.com/www.setup.com setup.com/www.setup.com/www.setup.com/www.setup.com/www.setup.com/www.setup.com/www.setup.com/www.setup.com/www setup.com/www.setup.com/www.setup.com/www.setup.com/www.setup.com/www.setup.com/www.setup.com/www.setup.com/www setup.com/w</pre>	
	Contraction of the second second		79		
1.	LAB_FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	44734	34	uVar5 = 0x4000000;	
	f10063e4f24 e0 03 18 as mov	x8,x24	81	1f (05tack261 t= 0) {	
		_csblab_pet_cdhash	82	war5 = 8x4084008;	
: 1		×23,×0	8.3		
	ff8063e4f38 77 83 88 b4 cbz	x23,LA8_fffffff0065e4f9c	34	>	
	ff00i5e4f34 e0 03 17 aa mov	x8,x23	85	puVar23 = (undefined *)_csplob_get_cdhash(uPare4);	
	ff8863e4f38 65 f1 ff 97 b1		85	if (puVar23 == (undefined *)@w#) {	
	ff9963e4f3c 68 97 99 34 cbz	PUN_1111110065e134c	82	pcWar17 = "Internal Error: No odhash found.";	
511		-0,LA8_fffffff8063x5028	88	LA8_ffffffff8063e4fa4:	
		v0,[x29]	89	FUN_fffffff0063e5e50(uParm1, uParm8, uParm8, pc%ar17);	
	fribblige4f44 02 01 56 2a orv	w2_w8_w28	94	>	
	ff0063e4f48 62 02 00 b9 str	w2, [x29]	91	etse (	
	ff0063e4f4c c3 d5 ff b0 adrp	x3,-0xffa363000	92	iVar6 = FUN_fffffff0063e134cfpuWar230;	
	f10063e4158 63 30 17 91 add	x3xxs_in-kernel_fffffffMScMScc,	93		
	f19963e4154 c8 03 35 as mov	x8,x22	94	SVar6 = FUIL_fffffff065c4990(p/Var23);	
	ff0063e4f58 e1 03 18 aa mov	x1,x24	95	if (Start == #) (	
	fr0063e4f5c e4 03 54 aa mov	x4_x20	95	LAB_fffffff0063a50x0;	E VI
	f19963e4168 e5 93 15 aa mov	x5,x21	92	pullar23 = (undefined +)_IOMalloc(0x1000);	
	FRANCISCHER AN AN AN AN AN AN	FIR FEFFEFEEEEEEEEE		Provide - Andreas and Angeler	



• From there dive deeper into the static trust cache lookup

1		b LNR_TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT		<pre>i wlonglung _peep_lookup_int_static_trust_cachelbyte epiforni_reput_tootil i i i i i i i i i i i i i i i i i i</pre>
6		LAB_OTTOTOBETSEScal	AMERIZZ): Protector Transfer	8 tanglang (tan 14 6 tyte aptrox 74
	-friefladcad of fe of cl -friefladcad of fe of cl -friefladcad bi ic of is in the later of its of is	star cr., babbe-dat_rerrereenance st rat_rerrereerantance mer ct., enc		7 Byte Bearls 8 unbert sileris 9 ubert sileris been ratio perting 10 byte ephysic printer to increase part of hash tables
1			AMEFILS: POSTOT	<pre>iii wheegineg_onerig ii wheegineg_onerig ii wheegineg_onerig ii wheegineg_onerig ii wheelig iii wheelig</pre>
	arlangtong Byto e storetong Tyte e storetong storetong storetong storetong	<ul> <li>FUNCTION</li> <li>alonglangrdect _prop looksa, is_statis_trust_s</li> <li>alongrdect _proprdect _proprdect</li></ul>	acherbyt	<pre>17 #* ELAP_IFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</pre>
	anderforestill	Stark)-HelMick book_30	AMEP(3): AMEP(3	III (phfaret_input_hash[1] ++ phtteri_patienter_to_part_at_hase_totle_biset_inter_mast_natio + 8x84 + 121 { uteri + tubet_patientinput_mast[2]; uteri = tubet_bitteri_patienter_to_non-tubet_at_hase_totle } uteri = tubet_bitteri_patienter_to_totle + 8x24 + 3]; if (phfaret_input_hast[2] +=
		ets etc.cl, for final (1)	1	<pre>in prest activity we part of here limit limit limit here, table + 848 + 121 6 in prest a futation and here file </pre>



#### • Using XREFs we can see that the static trust cache is set from here

	arr	ch, alr, #-feet	La Contrata de
find the fund at at the ro	810	19, 1st, #0.1481	- Nell and ch. dorf the set
## de 11 58	adir	since and policy partners striffereningerste, -in-	- National Constant C
#10000efac8 31 28 80 db	incer .	Con December 11 and 1	- And all all and the second sec
	81.	PUR, 121717/METINGAIN	- Table addr
_fillefact at 25 be be	adra	ally-Buttellahoner	-Tel: adv - closed, contracted and - co
#10070efact 00 10 41 Ft	344	-0, bot, sectional contractions	- Ave and
_110074efad0 40 00 04 b4	without .	LAR FTYTTTTBUTTETAR	Conference of Co
	ale		-Aph 41 Hi parts hell pro-
			- Total And
	AL THITTORE	Infail MAPTELS PROVIDE	- Rade 1at - 10,000,FFFFFFFBBBBBBBBBBBBBBBBBBBBBBBBBB
	adra		Loss of the second s
	ALC: NO	27,49x1	
file fand en 21 ht 30	atrib	wh. bat. #Mchalloodaf fffffffallana	ALC: N
_findTdefant 68 of al 18	adr.	closed_peaks_co_ne_hepary_nc_ffffffffdd78178648	- The second sec
1100/0e/test 11 20 40 40	TOP	Constitution of advantage of a second second second	- Total and a contract and a contrac
_110070efant 61 75 c7 18	adr	c)==047_1717111087804e546v1181825e8	- This or reached, sharest reaction of the
	100	Contraction of the second s	and the second s
F19878efaf4 e2 43 te 32	-	wild start	LODE 1 LOD 1 LOD PERSON
_118876+faf8 45 60 34 55	-	-1, #1.0	
-Helliefafc ca ff 12 84	81.	PE parse boot ange	
FT0076+F500 17 20 83 45	map	urbanden fahr	propinsion of C. H.
	lar	column, terretelendere, static, transache, size	- All str
	alled	CTULLINE TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	- Not an among the particularity of the second seco
	now	staffing the state of the state	- NH IN DESCRIPTION
#100100-fb10 ett 82 +7 58	Law	CTL_DAT_TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	
	-		The second s
	110	<pre>sile_definitionses sile_(sile_definitionselfinities)s</pre>	And Description of the second se
		id, bid, and a second s	- BR 44 - (1,1,1)
FF88786478262 00 80 15 AB	100	(30,404)	and the second s
	h.ht.	LAB. TTTTTTMETHATILL	- The et al. (Charles and Charles and Char
F10670+f1028 (3 43 44 F5	str	still, forfestional, the	- Test and Annual A
_118876+fb20 +0 +0 +1 30	ade	cleve "shirt sectalized trust cache re fffffff	
_118674e7520 27 28 63 63		and an on a range of the case of the second	
TERTMETERS OF JA BA TT BT	-	sector in the sector is the se	permittative at 1.00 M
		Test to the second seco	- AND the starting of the second seco
	Plas Poterry	AT CALL STORE LLAS THREE THE	<ul> <li>Note and interpretational prostant.</li> </ul>
	CALCULATION OF THE OWNER	and the second se	- NA JANE

BLACK HAT EVENTS

### • RE on the previous function reveals this trust cache structure

Trust Cache

```
struct cdhash {
   uint8_t hash[20]; //first 20 bytes of the cdhash
   uint8_t hash_type; //left as 0
   uint8_t hash_flags; //left as 0
};
struct static_trust_cache_entry {
   uint64_t trust_cache_version; //should be 1
   uint64_t unknown1; //left as 0
   uint64_t unknown2; //left as 0
   uint64_t unknown3; //left as 0
    uint64_t unknown4; //left as 0
    uint64_t number_of_cdhashes;
    struct cdhash[];
};
```

```
struct static_trust_cache_buffer {
    uint64_t number_of_trust_caches_in_buffer;
    uint64_t offsets_to_trust_caches_from_beginning_of_buffer[];
    struct static_trust_cache_entry entries[];
```

```
};
```



### • Using XREFs we can see that this structure is read from the device tree

and the second s		of \$124 are only but considered as
	641	<pre>style10. #Exc[0]sob#1_fffffff80080828</pre>
	edrp.	ATT, -Buffielded
	Tate-	+10, (x21, #0-dc8)++047_0111110070700c8
	- 100	48,428,48
	adrp	130,-00170103000
	atr	v0,1x38, #8x2303x-0AT_111111980804028
	adrp	rd, -befillenaa
	#1r.	-0, [st, #busid)QAT_fffffffeetenade
	mirp	at, -6.770723068
	MIP.	<pre>six, Ext, #http://wikit_fffffffffffffffffffffffffffffffffff</pre>
	advp	
FTB871548461 e8 F1 FF 14	100	htts.[200, #bob040.edb#_ffffff/MMFND4678_statio_
	mdrp.	disch_fanden8x1185(a800
	#04 #04	ulters_randomins_chean/memory-aup_fffffffmells_
	W1	n1, mp.#Px38 BTLookustetry
- F100713/0671 #0 75 40 F1	hale .	
	-	<pre>disrit_random_bio_Wiscol_314_styreep_base} si8xff87xa404</pre>
	+04	Close TrustCache (TTTTTT00703000, c1, 40,000
	-	4/, 429, #0458
	and a	17, 129, 401.78
FT00/710/00 5a d7 87 34	11	differ Property
	CRM	aller's reside. Phyl
	10.00	LAS_FFFFFFFABTUSRDER
	auto .	10, 120, #0100
_FTMF71b9ball 1a Bd 4P FP	Lar	420, Ect. #14001.703
	Ler.	uters random, faint
	#1	phystoky
	adra	10,-0.710103000
	1010	ulture runtes, Lob, Andels]CAT_TTTTTTMENENDAL-
	late	si, Int. Mati
	8079	x388v118183888
FF8871690ec 21 18 85 F9	atr	AL. LOTI, MINING GAT_PPPPPPPBCHETHERE statis
PIBS71506c8 e3 83 88 32	incre :	41,801
	-	w1.#14.7
	NO.	VI. #111
- REPORTATION DO DO DO		

1.818 1.0	
which are welling and an array	
- Bold and any con inclusion	
. Bod ter united, Mussioner, Pr.,	
- Boll and challent	
while all incluin, Machinesed, even.	
while while the distribution	
-BOR HAT IS, AND BRANDWARD PROFESSION	
- Bally safety rds, shifting the same	
- Ball and min. 14, Minandianal Josep.	
- Bar and the lot have been	
added and simplify the blocked bits	
- the stip cold, index, acceluate	
-BOT ME VIET, DECKING, DEAL RES.	
white and interfaces	
UND 1. PRODUCTS	
which the street, manufactory without down	
LINE AND LUCKTONIAM	
- Add and - Long Building Doctores.	
WHEN AND VIOLENCE	
- BOR IL ENGLISHING	
- Buff and " Antal a contra Mail	
man a re las divertings tanked	
Contra the contract of the second	
	- No. 40 (1)-10-10-10 - No. 40 (1)-10-10-10 - No. 40 (1) (1)-10-10-10 - No. 40 (1) (1)-10-10-10 - No. 40 (1) (1)-10-10-10 - No. 40 (1)-10-10-10-10 - No. 40 (1)-10-10-10-10-10 - No. 40 (1)-10-10-10-10-10 - No. 40 (1)-10-10-10-10-10-10-10 - No. 40 (1)-10-10-10-10-10-10-10-10-10-10-10-10-10-
1.1	
Manager Street, and store, or other land	
18 COLUMN TO A	
when the start instan 200, maniput	
- Ban tay in the statisticant with	
wheel and an address of the second	
- Bad and clubble	
-800 803 11,800	
ulter my visite	
LINE MY ILLEN	
CHARLES IN THE WORK PLANTING THAT	

ELACK HAT EVENTS

• Which Apple released the source code for

```
Trust
Cache
```

```
DTEntry memory_map;
MemoryMapFileInfo *trustCacheRange;
unsigned int trustCacheRangeSize;
int err;
```

```
err = DTLookupEntry(NULL, "chosen/memory-map", &memory_map);
assert(err == kSuccess);
```

```
err = DTGetProperty(memory_map, "TrustCache", (void**)&trustCacheRange
if (err == kSuccess) {
    assert(trustCacheRangeSize == sizeof(MemoryMapFileInfo));
```

```
segEXTRADATA = phystokv(trustCacheRange->paddr);
segSizeEXTRADATA = trustCacheRange->length;
```

arm\_vm\_page\_granular\_RNX(segEXTRADATA, segSizeEXTRADATA, FALSE



- Always works when only 1 hash in the list
- Only some items work when more than 1 item is in the list





# Any ideas?

FEHEU FIBLACK HATEVENTS



 Reversing this code revealed a binary search code which means the hashes are expected to be sorted in this list

	Underlie all a		-cast b	LAR_TYTTTTTBET255-34			<pre>//winging _peop_laship_id_static_trust_cachelbyte epideral_unput_text8</pre>
<b>i</b>			LAB_CONTINUES		AME#1231	******	e i byte autom/21
	Uplicat of a Uplicat at a Uplicat of a	c. 68 34	BL BL MARY	er, balledar, rerrernersaate Fal, rerrernersaate et, etc.	9		and a second sec
	100-04 N 2 100-04 N 4	1 41. #	1.40	adeau attentacio, attente, fact, emiliti attente, fact, effecte, fatt	AMEFILIS	mm	
	labele all e			and or family summarian			11 whit where; 13 If the later presentation is 40 K
1	stangton		- slanglangrate	FUETIN CL _ prog_ lanksg_ ik_statis_trust_ vietijas-	cacheribyt-		10 vesct_heat_table_pert_size = Colonglongletubst_siThAt_retretrendstablet = Bellig 10 vesct_pertret_size_retretrendstable = Bellig = Bellig 10 petert_pertret_pert_st_peak_tests = Dayte =10007_FTTTTTMentetedst = Bellig 11 es 4
	Batta a scoregitory Tarta a scoregitory		sh.8 sh.8 sh.8	Alfarel_sout_fach offict_back_tails_part_size alfar5_painter_ts_tartert_part_ offict5_painter_ts_tartert_part_	at here.	APTEN:	22 check, base, hash, hather + unert, hash, hathe, part, hits == 12 23 check = (uint)-parter, part, or hash, hathe [start, base, hash, hather = \$6(6); 24 biter) = pillert, part, or hash, hather[start, base, hash, hather = \$6(6); 25 check = (uint) = (uint);
	ander" (.res		\$7m(8)+8+(8) \$7m(8)+8+(8)	d Strat_34	399	071211 -	28 If (epidfared_light
	and and and		State]-BelW			Bynry P	Interf_interit_inter_table + Bitt + Bit           II         If (phfwrst_input_Next[1]) ++           III         phterit_inter_ta_privat_part_it_hase_table_biterit_inter_part_table + 1           III         phterit_inter_ta_privat_part_it_hase_table_biterit_inter_part_table + 1           III         phterit_inter_table_biterit_inter_part_table_biterit_inter_part_table + 1
	505cr# 17 a			10.10. Ph.0		Terrate	
	225014 14 4	r et al	etg.	alloid, by, Final 18			and a function for the board of

FEHEU VIELACK HAT EVENTE

121-4

+ 125 1



## And it works!

FEHEL FEBLACK HAT EVENTS



### Agenda

- Past public research on iOS on QEMU
- iOS kernel boot process
- Execution of non-apple executables with Trust Cache
- Bash execution with launchd
- UART interactive I/O
- Next steps





- Mount the RAMDisk image on OSX
- Remove all files in /System/Library/LaunchDaemons/
- Add a single file there for running bash (com.apple.bash.plist)
- Add the bash executable to the RAMDisk
- Add the bash executable to the Trust Cache
- Unmount the RAMDisk and run QEMU

<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" <plist version="1.0"> <dict> <key>EnablePressuredExit</key> <false/> <key>Label</key> <string>com\_apple\_bash</string> <key>POSIXSpawnType</key> <string>Interactive</string> <key>ProgramArguments</key> <array> <string>/iosbinpack64/bin/bash</string> </array> <key>RunAtLoad</key> <true/> <key>StandardErrorPath</key> <string>/dev/console</string> <key>StandardInPath</key> <string>/dev/console</string> <key>StandardOutPath</key> <string>/dev/console</string> <key>Umask</key> <integer>0</integer> <key>UserName</key> <string>root</string> </dict> </plist>



- System tries to execute bash
- Logs show missing libraries required for bash





# Any ideas?

FEHEU FIBLACK HATEVENTS



- The RAMDisk image comes without the dynamic loader cache on it, which is a file that holds most of the common runtime libs for iOS
- Copy this file into the RAMDisk at the correct path from the full disk images





# Any ideas?

FEHEU FIBLACK HATEVENTS



 Debug /usr/lib/dyld (the dynamic loader) which is responsible for loading the dynamic loader cache



```
// map in shared cache to shared region
int fd = openSharedCacheFile();
if ( fd != -1 ) {
       uint8_t firstPages[8192];
        if (::read(fd, firstPages, 8192) == 8192 ) {
                dyld_cache_header* header = (dyld_cache_header*)firstF
#if __x86_64__
                const char* magic = (sHaswell ? ARCH_CACHE_MAGIC_H : #
#else
                const char* magic = ARCH_CACHE_MAGIC;
#endif
                if ( strcmp(header->magic, magic) == 0 ) {
                        const dyld_cache_mapping_info* const fileMappi
                        const dyld_cache_mapping_info* const fileMappi
                        shared_file_mapping_np mappings[header->mappi
                        unsigned int mappingCount = header->mappingCou
```



### • Stepping through the execution path with gdb showed the error was in here

```
if (_shared_region_map_and_slide_np(fd, mappingCount, mappings, codeSi
        // successfully mapped cache into shared region
        sSharedCache = (dyld_cache_header*)mappings[0].sfm_address;
        sSharedCacheSlide = cacheSlide;
        dyld::gProcessInfo->sharedCacheSlide = cacheSlide;
        //dyld::log("sSharedCache=%p sSharedCacheSlide=0x%08lX\n", sSh
        // if cache has a uuid, copy it
        if ( header->mappingOffset >= 0x68 ) {
                memcpy(dyld::gProcessInfo->sharedCacheUUID, header->uu
else {
        throw "dyld shared cache could not be mapped";
        if ( gLinkContext.verboseMapping )
                dyld::log("dyld: shared cached file could not be mappe
```

FEHEU FIBLACK HAT EVENTS



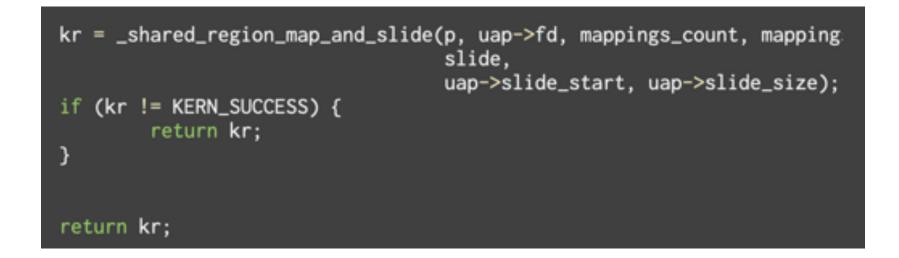
 Since we have a kernel debugger in gdb we can step into the system call in the kernel

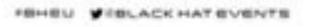
```
shared_region_map_and_slide_np(
        struct proc
                                                *р.
        struct shared_region_map_and_slide_np_args
                                                        *uap,
        ___unused int
                                                        *retvalp)
        struct shared_file_mapping_np
                                        *mappings;
        unsigned int
                                        mappings_count = uap->count;
        kern_return_t
                                        kr = KERN_SUCCESS;
                                        slide = uap->slide;
        uint32_t
#define SFM_MAX_STACK 8
        struct shared_file_mapping_np
                                        stack_mappings[SFM_MAX_STACK]
        /* Is the process chrooted?? */
        if (p->p_fd->fd_rdir != NULL) {
                kr = EINVAL;
                goto done;
```

FEHEU VIELACK HATEVENTS



• Stepping through this function we see that the call to \_\_\_\_\_\_shared\_region\_map\_and\_slide() is the part that fails





### • Stepping in that function reveals the error here

```
/* make sure vnode is owned by "root" */
VATTR_INIT(&va);
VATTR_WANTED(&va, va_uid);
error = vnode_getattr(vp, &va, vfs_context_current());
if (error) {
        SHARED_REGION_TRACE_ERROR(
                ("shared_region: %p [%d(%s)] map(%p:'%s'): "
                  "vnode_getattr(%p) failed (error=%d)\n",
                 (void *)VM_KERNEL_ADDRPERM(current_thread()),
                 p->p_pid, p->p_comm,
                 (void *)VM_KERNEL_ADDRPERM(vp), vp->v_name,
                 (void *)VM_KERNEL_ADDRPERM(vp), error));
        goto done;
if (va.va_uid != 0) {)
        SHARED_REGION_TRACE_ERROR(
                ("shared_region: %p [%d(%s)] map(%p:'%s'): "
                  "owned by uid=%d instead of 0\n",
                 (void *)VM_KERNEL_ADDRPERM(current_thread()),
                 p->p_pid, p->p_comm,
                 (void *)VM_KERNEL_ADDRPERM(vp),
                 vp->v_name, va.va_uid));
        error = EPERM;
        goto done;
```

### Bash on Launchd



- The code validates that the cache file is owned by root
- Mount the RAMDisk image in a different way to allow permission editing
- Copy the cache file and chown to root





## And it works!

FEHEL FEBLACK HAT EVENTS

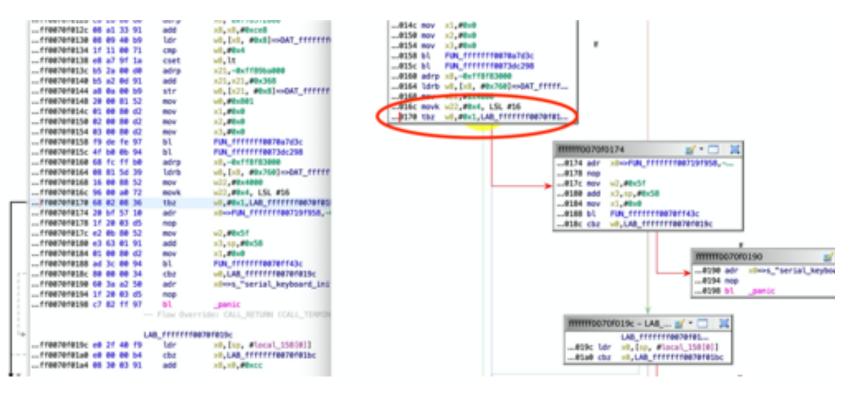


### Agenda

- Past public research on iOS on QEMU
- iOS kernel boot process
- Execution of non-apple executables with Trust Cache
- Bash execution with launchd
- UART interactive I/O
- Next steps



- UART output only was already possible with previous research
- Found where UART input is decided on in the kernel



VIBLACK HAT EVENTS



- Enabling UART input is decided based on bit #1 of a global var
- The global var is read from the "serial" kernel boot arg

	ff9971bed54	85	d9	ef.	94	b1
	f19973bad58	48	27		10	adrp
	f10073bad5c	88	a1	84	91	add
	ff9071bad68	69	54	12	10	adr
	f19973bad64	11	20	80	65	nop
	f19973bad68	14			71	ORD
	ff0071bad6c	28	41	88	94	csel
	ff9071bad78	49	22		bē	adrp
	f19973bad74	28	51	63	19	str
	f19973bad78	01	16	11	10	adrp
	ff0071bad7c	21	80	34	91	add
	f19971bad88	зr			69	str
	ff8073bad84	eŧ	13	11	94	adrp
	ff9071bad88	66	eŧ	28	91	add
	ff0071bad8c	e2	83	3e	32	<b>BOV</b>
	f19973bad98	03			52	BOV
-	ff8073bad94	24	43	er.	94	84
	ff9071bad98	48	82	88	34	chr
	ff0071bad9c	13	16	ff.	10	adrp
	f19971bada8	68	62	47	69	ldr .
	f19973bada4	89	81	3e	12	and
	ff9971bada8	e9	53	88	69	str
	ff0073badac	68	*1	28	37	tiona
	f19973badb#	eŧ	13	**	94	adrp
	ff9071badb4	66	fc	28	91	add
	ff9971badb8	e2	83	3e	32	BOV
	f19973badbc	e1	43	65	91	add
	ff8073badc8	03	88	88	52	<b>BOV</b>
	ff0071badc4	18	d3	ef.	94	61
-	ff0071badc8	c#			34	cha
	ff0073badcc	eß	53	48	69	<b>Mar</b>
-	ff9071badd8	88	88	88	34	cbz

x8,-0x118a41000		1834	2
x8, x8, #9x128		1015	DAT
x9,-0xff8e5fca0		1835	DA1
xy-extraction		1817	Det
War9_randon,#bx8	ш.	1058	294
x8,x9,x8,eq	п.	1019	11
x9,-0x11091d000		1828	
		1821	>
x8xx_serial_putc,[x9, #bx766]xx_PE_kput		1822	194
x10x178783000		1823	
x1==0AT_fffffff887676766,x1,#8x768		180.4	11
war, [x1]+>6#J_fffffff00707d760		1825	
uVar9_random,-0xf18fca000		1826	3
uVar9_randomos_serial_fffffff007836831		1827	· .
w2,#0x4		183	100
w3,#9x8		1829	11
PE_parse_boot_argn		1838	
uVar9_random,LAB_fffffff8071bade0		1831	
x19,-0xff0f83000			
w8,[x19, #8x768]+>DAT_fffffff887874768		1032	
w9, w8, #2x4		1033	
w9, 1sp, #local_150_physmap_base]		1834	
ul_eex2_LAB_fffffffee71badd8		1035	
uVar9_random0xf18fca000	ы.	1036	
uVar9_randomos_drain_uart_sync_ffffffig	ο.	1837	
w2.#8x4	п.	1038	>
x1, sp,#8x50		1039	1.5
w3,#0x0	ч.	1040	
PE_parse_boot_argn		1041	
uVar9_random,LAB_fffffff8871bade8		1842	
w8, [sp. #local_150_physmap_base]		1843	
LAB_fffffff0071bade0	11	1844	

serial init

1812	/* WWWEING: Subroutine does not return */
1013	_panic("\"Platform Expert not initialized\"");
1834	>
1015	DAT_fffffff0005742c8 = 0x11;
1835	BAT_11111110005742:0 = 0;
1017	DataNenoryNarv Ser (2, 30)
1018	<pre>IVar9 = Pt_parse_boot_argn("debug",&amp;local_150_physmap_base,4,4);</pre>
1819	if ((War9 != 0) 66 (((byte)local_150_physmap_base >= 3 6 1) != 0)) (
1828	DAT_ffffff007095cf0 = 0;
1821	>
1822	<pre>War9 = _serial_init();</pre>
1823	<pre>PE_kputc = FUN_fffffff0071x0360;</pre>
1824	ar damars to do d
1825	_PE_kputc = _serial_putcs
1825	>
1827	ALT_CONTINUEVOR = 0
183	
1829	IT THINK I IN ALL ALL
1838	((local 150 phytmap base =
1031	(longlong +)
1832	Elonglong +) E(ulonglong)local_150_physmap_base & #xffffffff000000000
1032 1033	<pre>Clonglong +) C(ulonglong)local_150_physmap_base &amp; 0xffffffff00000000   Uulonglong)_DAT_fffffff00000004),</pre>
1832 1833 1834	<pre>(longlong +) ((ulonglong)local_150_physmap_base &amp; 0xfffffffff00000000   (ulonglong)_DAT_fffffff00000004), (_DAT_ffffff00000004), &gt;&gt; 2 6 3) != 0   </pre>
1032 1033 1034 1035	<pre>(longlong +) ((ulonglong)local_150_physnap_base &amp; 0xfffffffff00000000   (ulonglong)_DAT_fffffff000000004), (_DAT_fffffff000000000000000000000000000000</pre>
1032 1033 1034 1035 1035	<pre>(longlong +) ((ulonglong)local_150_physmap_base &amp; 0xfffffffff000000000   (ulonglong)_DAT_fffffff000000004), (_DAT_fffffff000000000000000000000000000000</pre>
1802 1803 1804 1805 1805 1805	<pre>(longlong *) ([ulonglong]]ocal_150_physmap_base &amp; 0xfffffffff000000000   Ulonglong]_DAT_ffffffff00000004, (_DAT_ffffff00000004,argn("drain_wart_sync",&amp;local_150_physmap_base,4,0), IN ([Uuintlines]_150_physmap_base 5= 00)]])) ( DAT_ffffff0000004,argn("drain_wart_sync",&amp;local_150_physmap_base,4,0), IN ([Uuintlines]_150_physmap_base 5= 00)]]) ( DAT_ffffff0000000000000000000000000000000</pre>
1032 1033 1034 1035 1035 1035 1035 1038	<pre>(longlong +) ([ulonglong]]ocal_150_physmap_base &amp; 0xfffffffff000000000   (ulonglong]_DAT_fffffff000000000 &amp; 0xffffffff000000004), (_DAT_fffffff00000000000 &gt;&gt; 2 &amp; 6 3) 1= 0    ((Uvar9 = Mt_parse_boot_argn("drain_wart_sync",&amp;local_150_physmap_base,4,0), I) ((wint)local_150_physmap_base t= 00))))) ( DAT_fffffff00000000000 = _DAT_fffffff00000000000   4; ))</pre>
1802 1803 1804 1805 1805 1805 1805 1808 1809	<pre>(longlong +) ([ulonglong]]ocal_150_physmap_base &amp; 0xfffffffff000000000   (ulonglong]_DAT_fffffff00000004,), (_DAT_ffffffff00000000 &gt;&gt; 2 &amp; 3) != 0    ((Uvar9 = Pt_parse_boot_argn("drain_wart_sync",&amp;local_150_physmap_base,4,0), I) ((uint)local_250_physmap_base != 00))))) ( _DAT_fffffff000000000 = _DAT_ffffffff000000000   4; ) if (DAT_ffffff00000000000 == 0) (</pre>
1832 1833 1834 1835 1835 1835 1835 1839 1848	<pre>(longlong +) ([ulonglong]local_150_physnap_base &amp; 0xffffffff000000000 ] ([ulonglong]loAT_fffffff00000000 &amp; 0 2 6 3) 1= 0 [] ([Ulonglong]_DAT_fffffff00000000 &amp; 0 2 6 3) 1= 0 [] ([Ulonglong]_DAT_ffffff000000000, argn("drain_warf_sync",&amp;local_150_physnap_base,4,0), [] ([Ulonglong]10cal_150_physnap_base = 0)]))) ( [DAT_fffffff00000000000 = _DAT_ffffffff00000000000000000000000000000</pre>
1802 1803 1804 1805 1805 1805 1805 1808 1809 1848 1849	<pre>(longlong +) ((ulonglong)local_150_physmap_base &amp; 0xffffffff000000000000000000000000000</pre>
1802 1803 1804 1805 1806 1807 1808 1809 1848 1841 1841 1842	<pre>(longlong +) ((ulonglong)local_150_physmap_base &amp; 0xffffffff000000000   (ulonglong)loAT_fffffff00000000 &amp; 0 2 6 3) t= 0  ] ((Uvr9 = PE_parse_bood_argn("drain_sart_synt",&amp;local_150_physmap_base,4,0), IN ((uurt)local_150_physmap_base t= 00))))) ( _DAT_fffffff0000007000 == 0AT_ffffffff00000000   4; ) if (DAT_ffffff0000007000 == 0) ( local_150_physmap_base = (longlong +)((ulonglong)local_150_physmap_base &amp; 0xffffff iVar9 = PE_parse_boot_argn("validation_disables",&amp;local_150_physmap_base,4,0); local_150_physmap_base0_4_ = DAT_fffffff000000000000000000000000000000</pre>
1802 1803 1804 1805 1805 1805 1805 1808 1809 1848 1849	<pre>(longlong +) ((ulonglong)local_150_physmap_base &amp; 0xffffffff000000000000000000000000000</pre>



• Setting the "serial" boot arg to 2





## And it works!

FEHEL FEBLACK HATEVENTS

```
and prove a second s
tatic boolean_t
hread_invoke(
                                            self,
        thread_t
        thread_t
                                            thread.
       ast_t
                                            reason)
        if (__improbable(get_preemption_level() != 0)) {
                 int pl = get_preemption_level();
                 panic("thread_invoke: preemption_level %d, possible cause: %s",
                     pl, (pl < 0 ? "unlocking an unlocked mutex or spinlock" :</pre>
                          "blocking while holding a spinlock, or within interrupt context"));
        }
        thread_continue_t
                                   continuation = self->continuation;
       void
                                   *parameter = self->parameter;
        processor_t
                                   processor:
       uint64_t
                                   ctime = mach_absolute_time();
ifdef CONFIG_MACH_APPROXIMATE_TIME
        commpage_update_mach_approximate_time(ctime);
lendif
if defined(CONFIG_SCHED_TIMESHARE_CORE)
```

((thread->state & TH\_IDLE) == 0)



### **Demo – Research a vulnerability – voucher\_swap**

- Research done by Brandon Azad
- iOS 12.1 jailbreak
- Trigger the vulnerability while debugging



### ios\_command\_line\_tool.m

### int main(int argc, const char \*argv[]) { kern\_return\_t kr; mach\_port\_t thread; mach\_port\_t uaf\_port; mach\_port\_t discloser\_mach\_port = MACH\_PORT\_NULL;

### printf("stort Pacha");5

kr = thread\_create(mach\_task\_self(), &thread);

```
usf_port = create_voucher();
```

kr = thread\_set\_mach\_voucher(thread, uaf\_port);

```
voucher_release(uof_port);
```

```
mach_port_destroy(mach_task_self(), uaf_port);
```

```
thread_get_mach_voucher(thread, 0; &discloser_mach_port);
return 0;
```

ios\_command\_line\_tool.m:

186-1 188% 88

### [1 bash] [2 bash] bash [4 bash] [5 bash] [6 bash] [7 bash]

Sunday 17:29-17/11/2019



### Agenda

- Past public research on iOS on QEMU
- iOS kernel boot process
- Execution of non-apple executables with Trust Cache
- Bash execution with launchd
- UART interactive I/O
- Next steps



### **Next steps and challenges**

- IP communication
- Non RAMDisk disk support
- More hardware devices (disk, screen, touch, sound, comms, etc..)
- Load all the regular iOS services in the original launchd dir instead of just bash
- More than a single CPU and an interrupt controller
- More iOS versions and devices including KTRR, PAC and other features
- More gdb scripts (allocation zones info, objects info, etc...)
- Security research



### **Black Hat Sound Bytes**

- Use the the project and contribute! https://github.com/alephsecurity
- Check out our blog: https://alephsecurity.com
- Follow us on twitter: @alephsecurity @JonathanAfek
- Questions?

### HCL AppScan