

00001160	8d eb 6b 46 bf e2 2b 0e	13 f5 5e 93 85 9c 1f da	00001160	30 00 41 28 00 00 00 61	65 61 62 69 00 01 1e 00	IO.A(...seabi....)
00001170	5e c3 e8 1f 9e cc d2 86	79 13 07 1e 4d f4 63 c7	00001160	00 00 05 35 54 45 00 06	04 08 01 09 01 12 04 14	....5TE.....
00001180	99 b1 90 61 96 46 93 d3	33 4f 0d 0c 1b aa 0e 85	00001170	01 15 01 17 03 18 01 19	01 1a 02 00 2e 73 68 73	.....
00001190	61 08 a1 9e 99 5e d9 19	47 81 ba 54 55 05 92 8d	00001180	74 72 74 61 62 00 2e 69	64 75 62 70 00 2e 68	trtab.interp..h
000011a0	fb 21 24 c2 bf 26 0d 02	22 9a 8e aa 76 2e 1c 8f	00001190	61 73 68 00 2e 64 79 6e	73 79 6d 00 2e 64 79 6e	trtab.dynsym..dyn
000011b0	b4 03 1d 7d 12 3f 0f 6e	54 37 57 c6 4b 76 36 cf	000011a0	73 74 72 00 2e 72 65 6c	2e 70 6c 74 00 2e 74 65	istr..rel.plt..tel
000011c0	8d 45 94 9f 47 4b be da	55 c7 63 c0 99 ce a5 77	000011b0	78 00 00 2e 72 6f 64 61	74 61 00 2e 70 72 65 69	xt..rodata..prel
000011d0	43 38 f7 2e 50 89 c0 92	1 f f2 ae af af 96 44 af	000011c0	6e 69 74 5f 61 72 72 61	79 00 2e 69 6e 69 74 5f	init_array..init
000011e0	49 96 c2 61 61 66 c6 40	e8 fe 6a 44 2c 54 88 e7 d8	000011d0	61 72 72 61 79 00 2e 66	69 6e 69 5f 61 72 72 61	lannaj..fini.lannaj
000011f0	17 aa c1 92 7e 9b 4c f8	05 67 19 53 0c 06 0f	000011e0	79 00 2e 63 74 6f 72 73	00 2e 64 79 6e 61 6d 69	ly..ctors..dynsym
00001200	c1 b4 d0 33 46 9e da 73	49 e0 05 ef 9d 9e fc 0c	000011f0	63 00 2e 67 6f 74 00 2e	62 73 73 00 2e 63 6f 69	lc..got..bss..com
00001210	a3 93 e5 b5 38 79 96 f0	22 43 7f 5c f8 da 12 b6	00001200	6d 65 6e 74 00 2e 41 52	4d 2e 61 74 74 72 69 62	ment..ARM.attrib
00001220	de 67 3b 6b 02 b3 70 4a	90 18 04 86 03 20 01 1e	00001210	75 74 65 73 00 00 00 00	00 00 00 00 00 00 00 00	lutes.....
*			00001220	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00001240	76 29 3c 26 e6 ee 85 32	02 11 e1 5a 4a 1f 3a 78	00001240	0b 00 00 00 01 a1 00 00	02 00 00 00 d4 80 0e 00	.....
00001250	36 42 f1 5a 24 86 1f 9e	33 35 4f 82 b3 06 d5 2e	00001250	04 00 00 00 13 00 00 00	00 00 00 00 00 00 00 00	.....
00001260	ea be f6 20 c4 b2 44 8d	1f 90 c0 f4 a8 d2 2f 8f	00001260	01 00 00 00 00 00 00 00	13 00 00 00 00 05 00 00	.....
00001270	16 89 92 e0 ef 25 70 ff	9d 06 05 4b a9 58 c4 b2	00001270	02 00 00 00 00 00 00 00	08 00 00 00 00 00 00 00	.....
00001280	03 27 d1 ae 25 70 ff	9d 06 05 4b a9 58 c4 b2	00001280	02 00 00 00 00 00 00 00	08 00 00 00 00 00 00 00	.....
00001290	f4 97 3a 13	8d 06 05 4b a9 58 c4 b2	00001290	02 00 00 00 00 00 00 00	08 00 00 00 00 00 00 00	.....
000012a0	df 9e 38 84	8d 06 05 4b a9 58 c4 b2	000012a0	04 01 00 00 00 02 00 00	04 04 00 00 00 01 00 00 00	.....
000012b0	f5 0a 83 b4 3a b5 de 83	68 32 4d 50 c3 b6 9a de	000012b0	b4 00 00 00 10 00 00 00	21 00 00 00 03 00 00 00	.....
000012c0	0c 24 e7 68 23 75 83 83	1e 8a e2 ad c5 17 ac 56	000012c0	02 00 00 00 b4 83 00 00	b4 03 00 00 05 01 00 00	.....
000012d0	1a 17 83 0a b6 9c 17 0c	84 4d 3c 2d 03 ff 1a 82	000012d0	00 00 00 00 00 00 00 00	01 00 00 00 00 00 00 00	.....
000012e0	f1 e9 f9 c9 fb e6 2e a5	66 97 71 99 f4 6e c3 88	000012e0	29 00 00 00 09 00 00 00	02 00 00 00 bc 84 00 00	.....
000012f0	8b 3a 33 88 67 9a 84	af ab f8 14 56 1d 40 82	000012f0	bc 04 00 00 98 00 00 00	03 00 00 00 06 00 00 00	.....
00001300	ef 28 b7 3c 8d c1 a0 08	b7 be 8b fd a2 86 96 73	00001300	04 00 00 00 08 00 00 00	22 00 00 00 01 00 00 00	.....
00001310	e2 c3 a7 58 07 9a 9c 81	07 f9 4a f5 94 92 2b 67	00001310	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00001320	04 b7 46 ef af 9a 17 0c	d7 99 e5 d5 1e c9	00001320	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00001330	e2 74 73 f4 0a 33 ca c2	8d 06 05 4b a9 58 c4 b2	00001330	02 00 00 00 00 00 00 00	06 00 00 00 50 86 00 00	.....
00001340	64 1b ec dd b8 ab 1e 12	50 4f 4d e5 17 8c 65 ab	00001340	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00001350	8f 59 1b 80 4a c7 d8 69	bb 41 48 a2 e8 e3 63 8a	00001350	05 06 00 00 e8 02 00 00	00 00 00 00 00 00 00 00	.....
00001360	85 a7 1c 32 55 24 de d1	e2 84 cc b9 1e 7d 91 de	00001360	10 00 00 00 00 00 00 00	38 00 00 00 01 00 00 00	.....
00001370	a5 55 9f 5e 0b 9a c8 3e	89 64 21 61 62 16 e6 d6	00001370	32 00 00 00 38 89 00 00	38 09 00 00 e0 02 00 00	.....
00001380	df 6f cf 51 e2 1d 9b c8	32 15 2b 09 42 c5 53 91	00001380	00 00 00 00 00 00 00 00	04 00 00 00 01 00 00 00	.....
00001390	01 09 2a c2 9e 1e 89 93	55 99 e5 ef 18 7d 20 29	00001390	40 00 00 00 10 00 00 00	03 00 00 00 00 90 00 00	.....
000013a0	4b 6e 1c d8 cb f2 7f 66	54 aa a3 84 91 45 f7 95	000013a0	00 10 00 00 08 00 00 00	00 00 00 00 00 00 00 00	.....
000013b0	5e 1f 5a f6 fc fb de 05 60	fd d5 6f 4f 6e bc 1a fe	000013b0	01 00 00 00 00 00 00 00	4f 00 00 00 0e 00 00 00	.....
000013c0	1a 17 83 0e b6 9c 17 0c	84 ad 3c 2d 03 ff 1a 82	000013c0	03 00 00 00 08 90 00 00	08 10 00 00 08 00 00 00	.....
000013d0	93 ed 67 7c 55 ed d0 ba	6e 70 58 08 bf 09 77 8b	000013d0	00 00 00 00 00 00 00 00	01 00 00 00 00 00 00 00	.....
000013e0	e1 dd 6b 66 77 27 1e d5	22 4f 4c b5 66 95 79 7b	000013e0	00 00 00 00 0f 00 00 00	03 00 00 00 10 90 00 00	.....
000013f0	0e 3a 48 71 28 81 d6 11	74 48 5e a1 f3 e4 e9 e9	000013f0	01 00 00 00 08 00 00 00	00 00 00 00 00 00 00 00	.....
00001400	6d 8e 44 5d ea bb c4 42	a5 ba 9e 67 68 1e f5 ab	00001400	03 00 00 00 00 00 00 00	07 00 00 00 01 00 00 00	.....
00001410	3a 17 83 0e b6 9c 17 0c	84 ad 3c 2d 03 ff 1a 82	00001410	00 00 00 00 00 00 00 00	18 10 00 00 08 00 00 00	.....
00001420	ff c9 33 80 73 3b 30 33	ce bc 0c 0c 0c 2f da 4c	00001420	00 00 00 00 00 00 00 00	01 00 00 00 00 00 00 00	.....
00001430	da 21 47 c5 af 03 de e0	8a 2e 8c 91 70 d8 db da	00001430	00 00 00 00 06 00 00 00	03 00 00 00 20 90 00 00	.....
00001440	48 f9 9e 18 d8 e8 26 b9	f7 aa 0e 47 01 5c d2 43	00001440	20 10 00 00 c8 00 00 00	04 00 00 00 00 00 00 00	.....
00001450	fd 57 0f ca ee cc c3 df	d2 59 95 15 31 55 0d 51	00001450	04 00 00 00 08 00 00 00	77 00 00 00 01 00 00 00	.....
00001460	04 b7 46 ae af ab 9a 1f	d4 79 ed 97 d5 95 1e c2	00001460	03 00 00 00 e8 90 00 00	e8 10 00 00 58 00 00 00	.....
00001470	4c 8d 75 fc 6c 36 f7	d9 66 a5 ad 66 8b e1 8d	00001470	00 00 00 00 00 00 00 00	04 00 00 00 04 00 00 00	.....
00001480	a5 70 93 a7 03 7d 04 98	05 a9 8b 8b b7 8c 8e 8e	00001480	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....

# Pawn Storm: Malware on iOS devices?

Axelle Privrille - FortiGuard Labs, Fortinet

Insomni'hack, Geneva  
March 2015





- ▶ Security Researcher at **Fortinet, FortiGuard Labs**
- ▶ Focus: (strange) malware not on desktops/laptops
- ▶ E.g mobile malware, Internet of Things...
- ▶ Twitter: @cryptax



**Are there malware on iOS?**



**Are there malware on iOS?**

**Answer: Yes**  
*but not many*

**They're all for jailbroken phones, aren't they?**

# They're all for jailbroken phones, aren't they?



Find & Call

**No - but very rare**

iOS/FindCall (2012)

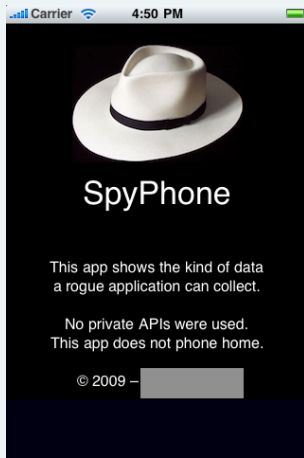
Found (and removed) in Apple Store

- ▶ Spams all your contacts
- ▶ Sends your (email/skype/...) passwords and location in clear text

# Other samples for **non jailbroken** iPhones

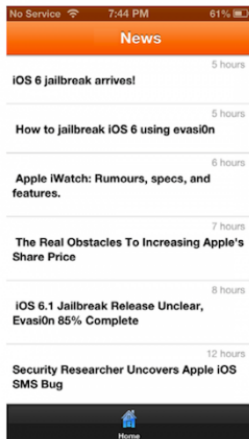


Adware/LBTM!iPhoneOS (2010)



iPhoneOS/Toires.A!tr.spy  
Nicolas Seriot, CH - 2009 - **PoC**

# PoC Jekyll malicious app on non jailbroken iPhones

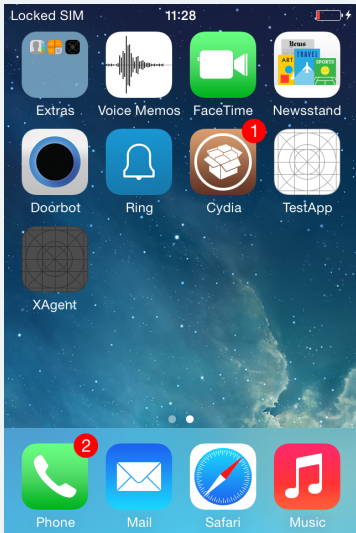


Credits: Tielei Wang, Kangjie Lu, Long Lu, Simon Chung, and Wenke Lee - Georgia Tech

*"Jekyll on iOS: When Benign Apps Become Evil"*,  
USENIX Security 2013



# Does PawnStorm run on non jailbroken iPhones?



Yes and No ;)

- ▶ Version A: will work, but with limits.
- ▶ Version B: requires jailbreak.

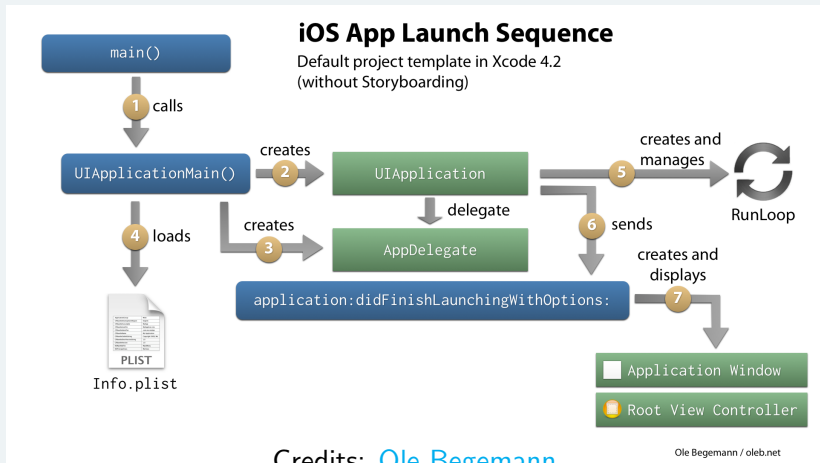
## The Operation

- ▶ Cyber espionage operation
- ▶ Discovered by [Trend Micro](#) in October 2014
- ▶ Targets military officials, government, defense industries

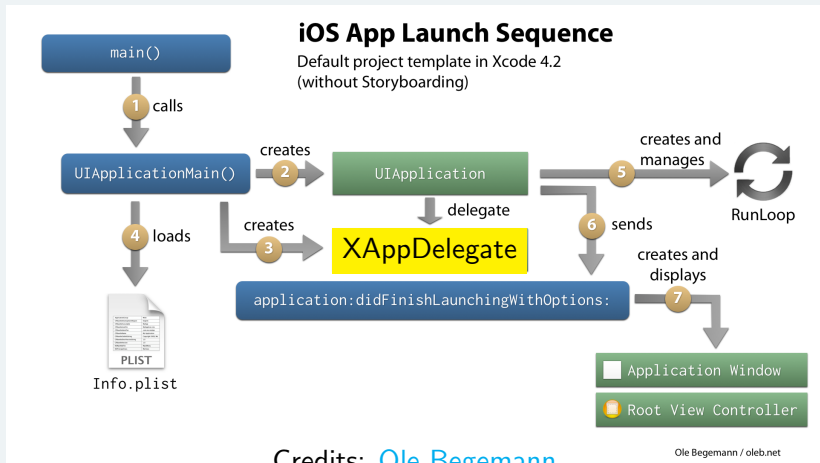
## iOS malware

- ▶ **Feb 4, 2015:** Trend Micro discovers two iOS samples
- ▶ **Version A:** XAgent - hidden trojan spyware
- ▶ **Version B:** madcap.dylib - malicious Cydia Substrate extension

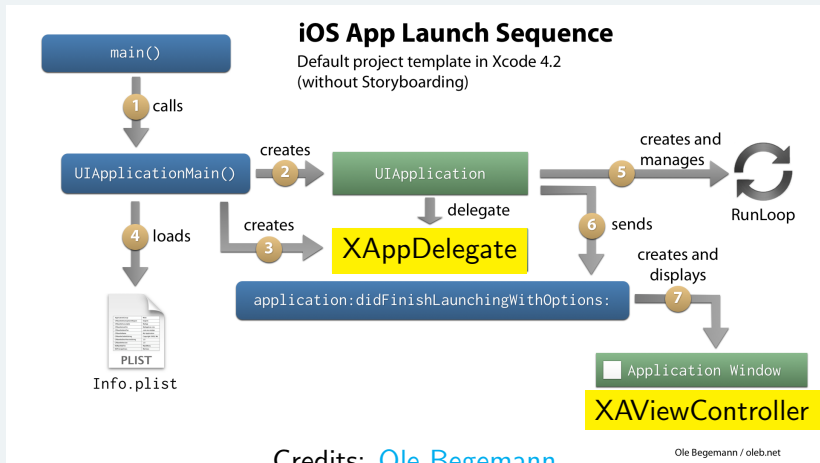
# What happens when iOS/PawnStorm.A!tr is launched?



# What happens when iOS/PawnStorm.A!tr is launched?




# What happens when iOS/PawnStorm.A!tr is launched?



# didFinishLaunchingWithOptions: Background Fetching

```
1 // XAAppDelegate - (char)application:(id) didFinishLaunchingWithOptions:(id)
2 char __cdecl __XAAppDelegate_application_didFinishLaunchingWithOptions__(struct XAAppDelegate *self,
3 {
4     id v4; // ST0C_4@1
5     int v11; // ST04_4@1
6     int v14; // [sp+14h] [bp-18h]@1
7     int v15; // [sp+14h] [bp-18h]@1
8     SEL v16; // [sp+14h] [bp-18h]@1
9     struct XAAppDelegate
10 {
11     v17 = self;
12     v16 = a2;
13     v15 = 0;
14     v4 = a4;
15     objc_storeStrong(&v15, a3);
16     v14 = 0;
17     objc_storeStrong(&v14, v4);
18     _objc_msgSend(&OBJC_CLASS__UIApplication, "sharedApplication");
19     _R0 = objc_retainAutoreleasedReturnValue();
20     _R3 = &UIApplicationBackgroundFetchIntervalMinimum;
21     __asm { ULDR          D16, [R3] }
22     v11 = _R0;
23     __asm { UMOV          R2, R3, D16 }
24     _objc_msgSend(_R0, "setMinimumBackgroundFetchInterval:");
25     objc_release(v11);
26     objc_storeStrong(&v14, 0);
27     objc_storeStrong(&v15, 0);
28     return 1;
29 }
```

[[UIApplication sharedApplication]  
setMinimumBackgroundFetchInterval:  
UIApplicationBackgroundFetchIntervalMinimum];



## Background Fetching in Info.plist

```
<key>UIBackgroundModes</key>  
  <array>  
    <string>fetch</string>
```

## Multi-tasking

- ▶ applicationWillResignActive
- ▶ applicationDidEnterBackground
- ▶ applicationDidEnterForeground
- ▶ applicationDidBecomeActive
- ▶ ...

Next method called **viewDidLoad**:

1. Instantiate **XA\_HTTP\_Chanel**: calls **getAgentID**. Retrieves a UUID.

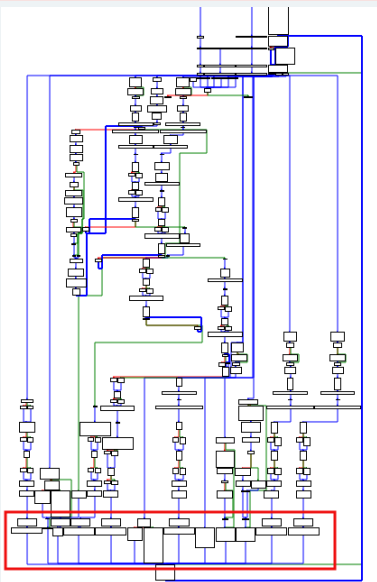
```
NSUUID *vendorIdentifier = [[UIDevice currentDevice]
    identifierForVendor];
uuid_t uuid;
[vendorIdentifier getUUIDBytes:uuid];
```

2. Creates a thread : **modulesThread**

```
_thread = [[NSThread alloc] initWithTarget:self
    selector:@selector(modulesThread:) object:nil];
```

modulesThread calls **cycleLoop** of **XAINfolphone**





It's a big switch

- 0 | Get Info Device
- 1 | Start Record
- 2 | Get Audio File
- 3 | Get Contact List
- 4 | Current Location
- 5 | Get Installed Apps
- 6 | Wifi Status
- 7 | Get All Pictures From Lib.
- 8 | List a given directory
- 9 | Get a given file
- 10 | Get process list
- 11 | Get SMS

## Example: Get All Pictures from Photo Library

```
break;
case 7:
    v265 = *((_DWORD *)v591 + 400);
    v743 = -1;
    v389 = -1;
    v388 = (int)&_objc_msgSend;
    v387 = &OBJC_IVAR__PhotoLibrary_http_param;
    v386 = (void (__fastcall *)(int, int, int))&_objc_msgSend;
    v385 = (int)seize_dequeue;
    v384 = &OBJC_IVAR__XA_HTTP_Chanel_out_array;
    _objc_msgSend(v265, "getAllPicturePhotoLibrary");
    v266 = *((_DWORD *)*((_DWORD *)v591 + 400) + *v387);
    v267 = *((_DWORD *)*((_DWORD *)*((_DWORD *)v591 + 400) + *v387) + *v384);
    v268 = *((_DWORD *)v385);
    v743 = v389;
    v386(v266, v268, v267);
    v743 = v389;
    v269 = objc_retainAutoreleasedReturnValue();
    objc_release(v269);
    break;
case 8:
```

## From disassembly

```
alasset_obj = &OBJC_CLASS___ALAssetsLibrary;
...
v18 = (void*alassetslib->library;
...
objc_msgSend(v18,
    "enumerateGroupsWithTypes:usingBlock:failureBlock:" ...);
```

## User authorization is not requested

```
if ([ALAssetsLibrary authorizationStatus])
{
    //Library Access code goes here
}
```

nowhere to be seen in the malware's code

# Get GPS coordinates

```
callocnan_obj = _objc_msgSend(&OBJC_CLASS__CLLocationManager, "alloc");
v4 = _objc_msgSend(callocnan_obj, "init");
locman = v25->locationManager;
v25->locationManager = (struct CLLocationManager *)v4;
objc_release(locman);
if ( (unsigned __int8) _objc_msgSend(&OBJC_CLASS__CLLocationManager, "locationServicesEnabled") )
{
    _objc_msgSend(v25->locationManager, "setDelegate:");
    _R2 = &kCLLocationAccuracyBest;
    _asm { ULDR    D16, [R2] }
    _R3 = (int)selRef_setDesiredAccuracy;
    _asm { UMOV    R2, R3, D16 }
    ((void (__fastcall *)(struct CLLocationManager *, _DWORD, int)) objc_msgSend)(
        v25->locationManager,
        "setDesiredAccuracy:", ← GPS accuracy is set to "best"
        _R2);
    _R2 = &kCLLocationDistanceFilterNone;
    _asm { ULDR    D16, [R2] }
    _R3 = (int)selRef_setDistanceFilter;
    _asm { UMOV    R2, R3, D16 }
    _R9 = &objc_msgSend;
    ((void (__fastcall *)(struct CLLocationManager *, _DWORD, int))_objc_msgSend)(
        v25->locationManager,
        "setDistanceFilter:", ← Report location for any movement
        _R2);                                     (no distance filter)
    _objc_msgSend(v25->locationManager, "startUpdatingLocation");
}
_objc_msgSend(v25->locationManager, "location");
v23 = objc_retainAutoreleasedReturnValue();
if ( v23 )
    objc_msgSend_stret(&v21, (const char *)v23, "coordinate");
else
    memset(&v21, 0, 0x10u);
v16 = objc_msgSend(&OBJC_CLASS__NSString, "alloc");
    _asm
    {
        ULDR    D16, [SP,#0x90+var_30]
        ULDR    D18, [SP,#0x90+var_28]
        USTR    D18, [R2,#0x90+var_8C]
        UMOV    R3, R9, D16
    }
v28 = objc_msgSend(v16, "initWithFormat:", CFSTR(" latitude:%F longitude:%F"), _R3, _R9);
```

Are location services enabled ?

GPS accuracy is set to "best"

Report location for any movement  
(no distance filter)

Format GPS coordinates

Since iOS 8, an additional requestAlwaysAuthorization must be requested

# Get SMS messages

```
v28 = 0;
v27 = objc_retain(CFSTR("<font size=4 color=blue><pre>"));
if ( !sqlite3_open("/var/mobile/Library/SMS/sms.db", &ppDb) )
{
    v2 = _objc_msgSend(&OBJC_CLASS__NSMutableDictionary, "alloc");
    v24 = _objc_msgSend(v2, "init");
    if ( !sqlite3_prepare_v2(ppDb, "SELECT id,rowid from handle", -1, &v25) )
    {
        while ( sqlite3_step(v25) == 100 )
        {
            v3 = _objc_msgSend(&OBJC_CLASS__NSString, "alloc");
            sqlite3_column_text(v25, 0);
            v4 = _objc_msgSend(v3, "initWithCString:encoding:");
            v5 = _objc_msgSend(&OBJC_CLASS__NSString, "alloc");
            sqlite3_column_text(v25, 1);
            v6 = _objc_msgSend(v5, "initWithCString:encoding:");
            ((void (__Fastcall *) (int, _DWORD, int, int))_objc_msgSend)(v24, "setValue:forKey:", v4, v6);
            objc_release(v6);
            objc_release(v4);
        }
    }
    if ( !sqlite3_prepare_v2(ppDb, "SELECT text,handle_id,is_from_me,account from message order by date desc", -1, &v26) )
    {
        while ( sqlite3_step(v26) == 100 )
        {
            ++v28;
            v7 = _objc_msgSend(&OBJC_CLASS__NSString, "alloc");
            sqlite3_column_text(v26, 0);
            v23 = _objc_msgSend(v7, "initWithCString:encoding:");
            v8 = _objc_msgSend(&OBJC_CLASS__NSString, "alloc");
            sqlite3_column_text(v26, 1);
            v22 = _objc_msgSend(v8, "initWithCString:encoding:");
            ((void (__Fastcall *) (int, _DWORD, int, int))_objc_msgSend)(v24, "setValue:forKey:", v23, v22);
            objc_release(v22);
            objc_release(v23);
        }
    }
}
```

Format HTML log

Read the SMS database  
jailbreak required

Perform SQL queries

# Get Installed Applications

```
MOV R2, (selRef_stringByAppendingPathComponent - 0xBFBC) ; selRef_stringByAppendingP
ADD R2, PC ; selRef_stringByAppendingPathComponent
MOV R3, (__InstalledApp_getInstalledApp__.$cacheFileName - 0xBF6) ; -[InstalledApp ge
ADD R3, PC ; -[InstalledApp getInstalledApp:].cacheFileName
LDR R3, [R3] ; "com.apple.mobile.installation.plist"
LDR R2, [R2] ; "stringByAppendingPathComponent:"
MOV R9, R0
STR R0, [SP,#0xBC+var_3C]
MOV R0, R9
STR R1, [SP,#0xBC+var_40]
MOV R1, R2
MOV R2, R3
LDR R3, [SP,#0xBC+var_40]
BLX R3
MOV R7, R7
BLX _objc_retainAutoreleasedReturnValue
STR R0, [SP,#0xBC+var_18]
LDR R0, [SP,#0xBC+var_3C]
BLX _objc_release
BLX _NSHomeDirectory
MOV R7, R7
BLX _objc_retainAutoreleasedReturnValue
MOV R1, (cfstr_____ - 0xBF6) ; "../.."
ADD R1, PC ; "../.."
MOV R2, (_objc_msgSend_ptr_0 - 0xC006) ; _objc_msgSend_ptr_0
ADD R2, PC ; _objc_msgSend_ptr_0
LDR R2, [R2] ; _imp_objc_msgSend
MOV R3, (selRef_stringByAppendingPathComponent - 0xC012) ; selRef_stringByAppendingP
ADD R3, PC ; selRef_stringByAppendingPathComponent
LDR R0, [R0]
```

installation.plist file

plist is expected to be found at  
../.. from sandbox directory

To get outside the sandbox → jailbreak

## Pseudo Objective C code

```
static NSString *const cacheFileName =
    @"com.apple.mobile.installation.plist";
NSString *relativeCachePath = [[@"Library"
    stringByAppendingPathComponent:
    @"Caches"] stringByAppendingPathComponent:
    cacheFileName];
path = [[NSHomeDirectory() stringByAppendingPathComponent:
    @"../.."] stringByAppendingPathComponent:
    relativeCachePath];
```

## List Directory

```
nsfileman_obj = &OBJC_CLASS__NSFileManager;
defaultMan = "defaultManager";
v105 = objc_msgSend(nsfileman_obj, defaultman);
...
v104 = objc_msgSend((void*)v202,
    "contentsOfDirectoryAtPath:error",
    path,
    &error);
...
v84 = objc_msgSend(&OBJC_CLASS__NSString,
    "stringWithFormat:",
    CFSTR("<table><caption color=blue> Directory:
    %@ </caption>"),
    path);
```

Sandboxing limits to /private/var/mobile/Applications/THEAPP



## Pseudo decompiled code in XAInfophone getInfoDevice

```
telephony_obj = objc_msgSend(  
    &OBJC_CLASS__CTTelephonyNetworkInfo,  
    "alloc");  
v1223 = objc_msgSend(telephony_obj, "init");  
...  
subscriberProvider = "subscriberCellularProvider";  
...  
v1449 = objc_msgSend(v1153, subscriberProvider);  
...  
v1448 = objc_msgSend((void*)v9, "mobileNetworkCode");  
...  
v1447 = objc_msgSend(v12, "mobileCountryCode");
```

## Later in XAInfolphone getInfoDevice

Phone number is read from /private/var/wireless/Library/Preferences/com.apple.commcenter.plist

Out of sandbox → **Requires jailbreak**

Get the "PhoneNumber" key

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>CarrierBundleName</key>
  <string>208[REDACTED]</string>
  <key>ICCID</key>
  <string>8933[REDACTED]2</string>
  <key>LASDNextUpdate</key>
  <date>2015-02-22T10:19:13.915174Z</date>
  <key>NextUpdate</key>
  <date>2015-01-30T17:26:37.694534Z</date>
  <key>PhoneNumber</key>
  <string>+3361[REDACTED]</string>
</dict>
</plist>
```

The application icon does not appear on the home screen:

```
<key>SBAppTags</key>  
  <array>  
    <string>hidden</string>  
  </array>
```

## Limitations

- ▶ Known not to work on iOS 8
- ▶ Hidden tag is easy to detect → Apple bans it from Apple Store

## Requires jailbreak

- ▶ Read **SMS** database
- ▶ Read com.apple.commcenter.plist for **phone number**
- ▶ **Hiding** icon

## Limited without jailbreak

- ▶ List content of **directory**
- ▶ Retrieve **file**

## Malware does not ask these authorizations

Will not work (unless granted from elsewhere)?

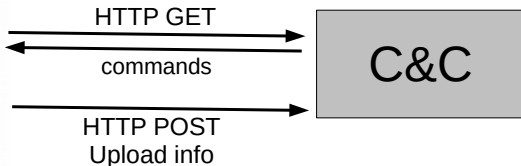
- ▶ **Get Photos** from library. Should request `requestRecordPermission`
- ▶ **Geolocation**. Authorization needed for iOS 8.
- ▶ **Record voice**. Should request `requestRecordPermission`

## Malware asks for these authorizations

**Read lists of contacts**. Code requires authorization via `ABAddressBookRequestAccessWithCompletion`

- ▶ Get **model**, name, systemName, systemVersion, localizedModel via UIDevice
- ▶ Test for existence of **jailbreak** via `/private/var/lib/apt`
- ▶ List **running process** via call to Unix command `sysctl`
- ▶ **WifiStatus** via calls to Reachability API
- ▶ Background fetching of C&C URLs
- ▶ Take **screenshots**??? (not called)

iPhone infected  
With Pawn Storm

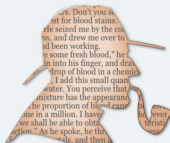


`hxxp://198.27XXXXXX/watch/?ai=<BASE 64 of RC4 data>`

`close/? text= 50 bytes key`  
`search/? from=`  
`find/? utm=`

...

# Who wrote Pawn Storm?



## Indications

- ▶ **BuildMachineOSBuild** 13E28: 10 possibilities: iMac, MacBook Pro, MacBook Air...
- ▶ /Users/mac/Desktop/work/IOS\_PROJECT
- ▶ XAgent-azeuhvvhelifolbyqbjqwuwimdho?
- ▶ **DTXcodeBuild** 5B1008: Xcode 5.1.1
- ▶ **Implementation.** Intended for jailbroken iOS 7.x?
- ▶ **Many typos:** XA\_HTTP\_Chanel, runningProcecces, generateUrlParametrs
- ▶ **Grammar:** "Host not exist" ...



# Am I safe from iOS/PawnStorm?

**YES** ... probably!

```
def safeFromPawnStormA():
    if (military official or defense contractor): #etc
        if (iOS >= 8):
            print "Do you have an XAgent icon?"
        elif (iOS >= 7.1):
            if (jailbroken iPhone):
                print "Check your iPhone"
            else: # only parts of XAgent can work
                print "Probably safe, check to be sure"
        else:
            print "Probably ok"
    else:
        print "You're not targeted, so probably safe"
```



# It's a **TARGETED** malware! Not for the *masses*

Quoting myself from [Fortinet's blog](#)

*"it is unlikely you'll be affected because the malware probably wasn't distributed massively, but only to targeted victims"*

*"it is very unlikely the malware could have been on the Apple Store "*

Not my fault if anything concerning iOS gets more attention in the press, is it? ;P

It's a **TARGETED** malware!

Not for the *masses*

Its **importance** depends ... on the **target!**

Quoting myself from [Fortinet's blog](#)

*"it is unlikely you'll be affected because the malware probably wasn't distributed massively, but only to targeted victims"*

*"it is very unlikely the malware could have been on the Apple Store "*

Not my fault if anything concerning iOS gets more attention in the press, is it? ;P

Am I infected with Pawn Storm?

Am I infected with Pawn Storm? **Probably not**

Am I infected with Pawn Storm? **Probably not**  
Was Pawn Storm on the Apple Store?

Am I infected with Pawn Storm? **Probably not**  
Was Pawn Storm on the Apple Store? **No**

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for?



Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones?

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

... And on jailbroken iPhones?

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

... And on jailbroken iPhones? **Yes !!!**

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

... And on jailbroken iPhones? **Yes !!!**

Can I spot it?

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

... And on jailbroken iPhones? **Yes !!!**

Can I spot it? **On iOS 8, yes, otherwise difficult**

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

... And on jailbroken iPhones? **Yes !!!**

Can I spot it? **On iOS 8, yes, otherwise difficult**

Who coded it?



Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

... And on jailbroken iPhones? **Yes !!!**

Can I spot it? **On iOS 8, yes, otherwise difficult**

Who coded it? **We don't know**

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

... And on jailbroken iPhones? **Yes !!!**

Can I spot it? **On iOS 8, yes, otherwise difficult**

Who coded it? **We don't know**

Is iOS safe from malware?

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

... And on jailbroken iPhones? **Yes !!!**

Can I spot it? **On iOS 8, yes, otherwise difficult**

Who coded it? **We don't know**

Is iOS safe from malware? **No !!!**

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

... And on jailbroken iPhones? **Yes !!!**

Can I spot it? **On iOS 8, yes, otherwise difficult**

Who coded it? **We don't know**

Is iOS safe from malware? **No !!!**

Is Android less safe than iOS?

Am I infected with Pawn Storm? **Probably not**

Was Pawn Storm on the Apple Store? **No**

What version is it for? **iOS 7.1 and +**

Does it *run* on non jailbroken iPhones? **Yes but  
with limits**

... And on jailbroken iPhones? **Yes !!!**

Can I spot it? **On iOS 8, yes, otherwise difficult**

Who coded it? **We don't know**

Is iOS safe from malware? **No !!!**

Is Android less safe than iOS? **Perhaps. Difficult  
question**

# Thank You !

## Contact info

**@cryptax** or aapvrille (at) fortinet (dot) com

## References and interesting links

- ▶ [Blog post from Trend Micro](#)
- ▶ [Blog post on Fortinet](#)
- ▶ Wang et al, [Jekyll on iOS, USENIX Security 2013](#)
- ▶ C. Livitt, [Rethinking & Repackaging iOS Apps: Part 1](#), Feb 2015
- ▶ Zheng et al, [Enpublic Apps: Security Threats Using iOS Enterprise and Developer Certificates](#), ASIA CCS 2015

Thanks to : Claud Xiao, Ruchna Nigam, Nicolas Seriot, Trend  
Micro

PowerPoint? No way! This is [Lobster](#)

