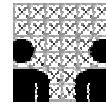




University of Hamburg
Fachbereich Informatik
Arbeitsbereich AGN



Risk Analysis of Mobile Devices with Special Concern of Malware Contamination

Diploma Thesis

Henrich C. Pöhls

Alsterweg 13
22339 Hamburg
Germany

Phone: ++4940 611 64 378

E-mail: Henrich.aVTC@Poehls.com

Matrikel-Nr.: 4964746

First Supervisor: Prof. Dr. Klaus Brunnstein

Second Supervisor: Prof. Dr. Norbert Ritter

August 2003

This page is intentionally left blank

Erklärung

Ich versichere, die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe angefertigt zu haben.
Für die Arbeit habe ich keine außer den angegebenen Quellen und Hilfsmitteln benutzt.
Ich bin mit einer Einstellung in den Bestand der Bibliothek des Fachbereichs einverstanden.

Hamburg, den 31.8.2003

(HENRICH C. PÖHLS)

This page is intentionally left blank

Index

1	Introduction	1
2	Terms and Definitions	3
2.1	Towards a "Risk Analysis" of mobile devices	3
2.2	The Term "Mobile Device" and Related Terms	3
2.2.1	The Term "Desktop"	4
2.2.2	The Term "MS Pocket PC 2002"	4
2.2.3	Why the Pocket PC 2002 Platform?	5
2.3	The Connection State of a Mobile Device	8
2.3.1	Connection State: Disconnected	8
2.3.2	Connection State: Connected	8
2.3.3	Connection State: Wirelessly Connected	8
2.3.4	Connection State: Cradled (Home System & Pairing Process)	8
2.4	The Term "Malware" and Related Terms	9
2.4.1	The Term "Virus"	9
2.4.2	The Term "Worm"	9
2.4.3	The Term "Trojan Horse"	9
2.4.4	The Term "Hostile Applet"	10
2.4.5	The Term "Malware Distribution"	10
2.4.6	The Term "Malware Contamination"	11
2.4.7	The Terms "Testbeds", "Test Sets" and "Samples"	12
3	Pocket PC 2002 Hardware	15
3.1	Pocket PC 2002 Hardware Specification	15
3.1.1	ARM Processor	15
3.1.2	ARM Architectures and Instruction Sets	16
3.2	Available Pocket PC 2002 mobile devices	16
3.2.1	Short list of mobile devices	16
3.2.2	Categorization by processor architecture	17
3.2.3	Categorization by CPU clock and size of RAM	18
4	Pocket PC 2002 Operating System	19
4.1	Windows CE Versions	19
4.1.1	Windows CE Version 1.0	19
4.1.2	Windows CE Version 2.0	19
4.1.3	Windows CE Version 3.0	19
4.1.4	Windows CE.net	19
4.2	Versions build on Windows CE 3.0	20
4.2.1	Pocket PC 2002: Professional, Premium & Phone Edition	20
4.2.2	Smartphone 2002	20
4.3	Pocket PC 2002 core functions	21
4.3.1	Process and Thread Management	21
4.3.2	Synchronizing Processes and Threads	23
4.3.3	Memory Types	25
4.3.4	Virtual Memory	27
4.3.5	Files, Databases and Persistent Storage	30
4.3.6	Interprocess Communications	33
4.3.7	User Interface Services	34
4.3.8	Multimedia	36
4.4	Pocket PC 2002 network functions	36
4.4.1	Overview of Network Components	37
4.4.2	Serial Connections [client/server]	38
4.4.3	Telephony API (TAPI) [client]	38
4.4.4	Remote Access Service (RAS) [client]	38
4.4.5	Windows CE Sockets (WinSocks) [client/server]	38
4.4.6	Windows Networking (WNet) [client]	39
4.4.7	Windows CE Internet API (WinInet) [client]	40
4.4.8	TCP/IP stack in Windows CE 3.0 [client/server]	41
4.4.9	Network Driver Interface Specification (NDIS) [client/server]	41
4.4.10	Point-to-Point Protocol (PPP) [client/server]	42
4.4.11	Virtual Private Network (VPN) [client]	43

4.4.12	Terminal Services [client].....	43
4.4.13	ActiveSync	43
4.4.14	Connection Manager	48
4.5	Pocket PC 2002 power management	50
4.6	Pocket PC 2002 security functions	51
4.6.1	Power-On Protection	51
4.6.2	Security Support Provider Interface (SSPI)	53
4.6.3	Cryptography API (CAPI).....	54
4.6.4	Digital Certificate Handling	54
4.6.5	Smartcard Support.....	55
4.6.6	Trust-Model [not implemented in Pocket PC 2002].....	55
4.6.7	Policy Restriction [undocumented Pocket PC 2002 function]	57
5	Pocket PC 2002 Applications.....	61
5.1	File Explorer	61
5.1.1	File Extension not Displayed	61
5.1.2	Display the file extension using “Beam File ...”	61
5.1.3	Hidden Files Not Listed by Default	62
5.2	Pocket Word.....	62
5.3	Pocket Excel	63
5.4	Pocket Internet Explorer	63
5.4.1	ActiveX.....	64
5.4.2	Java Virtual Machine	64
5.4.3	JScript Version 3.0.....	64
5.4.4	Tel-URLs.....	65
5.4.5	SSL Connections	65
5.4.6	Settings	67
5.5	Pocket Outlook.....	68
5.5.1	Connections.....	68
5.5.2	Attachments.....	70
5.5.3	HTML-Content in E-mails	71
5.6	eMbedded Visual Basic, eMbedded Visual C++.....	71
5.6.1	eMbedded VisualBasic (eVB).....	72
5.6.2	eMbedded Visual C++ (eVC).....	72
6	Risks of Malware Contamination and Distribution.....	73
6.1	General Comments on Malware	73
6.1.1	No Pocket PC 2002 specific Malware known today	73
6.1.2	Reasons for Pocket PC 2002 specific Malware in the future	73
6.2	Requirements for Malware Distribution.....	74
6.3	Requirements for Malware Contamination	75
6.3.1	Virus Requirements: Invocation, Execution, File Write Access.....	75
6.3.2	Worm Requirements: Invocation, Execution, Network Access	76
6.3.3	Trojan Horse Requirements: Invocation, Execution, Network Access (Online) / File Write Access (Offline), Hiding	77
6.3.4	Hostile Applet Requirements: Invocation, Execution.....	78
6.4	Points of Entry for Malware on Pocket PC 2002.....	78
6.4.1	Entry Point: File System	79
6.4.2	Entry Point: Documents	80
6.4.3	Entry Point: Internet Web Pages	80
6.4.4	Entry Point: E-mails	81
6.5	Pocket PC Functions and Properties Exploitable by Malware.....	81
6.5.1	Operating System Vulnerability: Autostart Functionality	82
6.5.2	Operating System Vulnerability: No Registry Access Protection	82
6.5.3	Operating System Vulnerability: No Restriction on Application Execution.....	82
6.5.4	File System Vulnerability: Auto-Run from Removable Storage Media	83
6.5.5	File System Vulnerability: No File Access Protection.....	83
6.5.6	File System Vulnerability: Overloading ROM Files.....	84
6.5.7	Connection Manager: Auto Establish Pre-Defined Connections.....	84
6.5.8	ActiveSync Vulnerability: RAPI.....	84
6.5.9	Standard Application Set Vulnerability: No List of All Running Processes.....	85
6.5.10	Pocket Outlook Vulnerability: Shortened Attachment’s Filename	85
6.5.11	Pocket Internet Explorer Vulnerability: Scripting	85
6.5.12	File Explorer Vulnerability: Hidden Files Not Shown by Default.....	86

6.5.13	File Explorer Vulnerability: File Extension Hidden.....	86
6.5.14	Pocket Word and Excel Vulnerability: Embedded Content Hidden.....	87
6.6	Risk Evaluation.....	87
6.6.1	Risk of Malware Contamination.....	88
6.6.2	Risk of Malware Distribution: High.....	97
7	Malware Defence on Pocket PC 2002.....	99
7.1	Build-in Defences.....	99
7.1.1	Disable Auto-Run from Removable Storage Media Functionality.....	99
7.1.2	Enable Undocumented Policy Restrictions.....	99
7.1.3	Remove Association between vb Extension and eVB Interpreter.....	100
7.1.4	Check "View all ..." in File Explorer.....	100
7.1.5	Do Not Save Passwords in the Connection Manager.....	100
7.1.6	Enable the Power-On-Protection.....	100
7.1.7	Disable Automatic Download in Pocket Internet Explorer.....	100
7.1.8	Disable Pocket Internet Explorer Scripting and ActiveX.....	101
7.2	Test of Anti-Malware Products for Pocket PC 2002.....	101
7.2.1	Test Goals.....	101
7.2.2	Test Conditions.....	101
7.2.3	Test Measurements.....	103
7.2.4	Test Procedures.....	104
7.3	Anti-Virus Products for Pocket PC 2002.....	110
7.3.1	AVP: Kaspersky Anti-Virus for Windows CE [tested].....	111
7.3.2	BDF: Softwin BitDefender for Windows CE [not tested].....	114
7.3.3	FSE: F-Secure Anti-Virus for Pocket PC Version 1.5 [tested].....	114
7.3.4	INO: Computer Associates eTrust Antivirus 7.0 for Pocket PC [tested].....	118
7.3.5	McAfee VirusScan Wireless [not tested].....	121
7.3.6	PCC: PC-cillin for Wireless [tested].....	121
7.4	Test Results for AV-Products for Pocket PC 2002.....	124
7.4.1	Evaluation of Test Results for the Macro In-the-Wild testbeds.....	125
7.4.2	Evaluation of Test Results for Script In-the-Wild Testbeds.....	129
7.4.3	Comparison of Pocket PC Anti-Malware Products and Desktop Products.....	133
7.4.4	Automatic Scan on Removable Storage Media Insertion.....	135
8	Conclusion.....	139
8.1	General Pocket PC Security.....	139
8.2	Risk of Malware Contamination and Distribution.....	139
8.3	Anti-Malware Products for Pocket PC.....	140
8.4	Suggestions for Future Work.....	141
	Bibliography.....	143
	List of Figures.....	167
	List of Tables.....	169
	Acknowledgement.....	171
	Appendices.....	173
	Appendix A: Abbreviations	
	Appendix B: Details of Pocket PC 2002 compatible devices	
	Appendix C: Details of ActiveSync Packets	
	Appendix D: Source Code of Tools and other Code	
	Appendix E: Details of the Testbeds	
	Appendix F: Results of Pocket PC Scanner Test 2003-05	

Legend

Throughout this work I will use the following type formatting for indication purposes:

`Text` Normal text

`Text` Program code, program output, function and variable names, path and file names

Text Examples

Cross references to other chapters are given in brackets:

... in eMbedded Visual basic (see 5.6) as they are also ...

References to books, papers, and online materials are given in squared brackets, and the details can be found in the bibliography at the end:

... from the ASUS A600 manual [A600MANUAL].

I also use coloured text in tables or figures:

Red This indicates that this will support malware or malicious functions can facilitate it.

Green This indicates that this will not support malware or malicious functionality is blocked or limited with this.

1 Introduction

More and more mobile devices are used in today's computing, and even more will be used in the future. In the year 2000, about 11,000 units of personal digital assistants (PDAs) were sold worldwide [ETFORECAST2002]. By the year 2004 the same analysts estimate the number of PDAs sold worldwide to be already over 25,000 units [ETFORECAST2002] (also see Figure 4 in 2.2.3.1).

These mobile devices will enable the user to do more and more of his or her daily work being mobile. Fixed desktop computers will still be used, but who would want to be bound to a desk to achieve the task using the desktop computer, if he or she could achieve the same with a mobile device but sitting at a chosen place? In work environments, workers could do their work while travelling or would always have their data with them, when they go to meetings.

But this mobile computing also has risks, there are new risks that come from the mobility, but this work will concentrate on an old threat, that a lot of users are already aware of: **The threat of malware.**

The desktop environment knows this threat, and countermeasures have been deployed and tested. The number of malware grows yearly. The following figure shows the number of malware that is used in the anti virus test center (aVTC) to assess the quality of desktop anti-malware products (numbers are taken from [SEEDORF2002] and [VTC2002-12]).

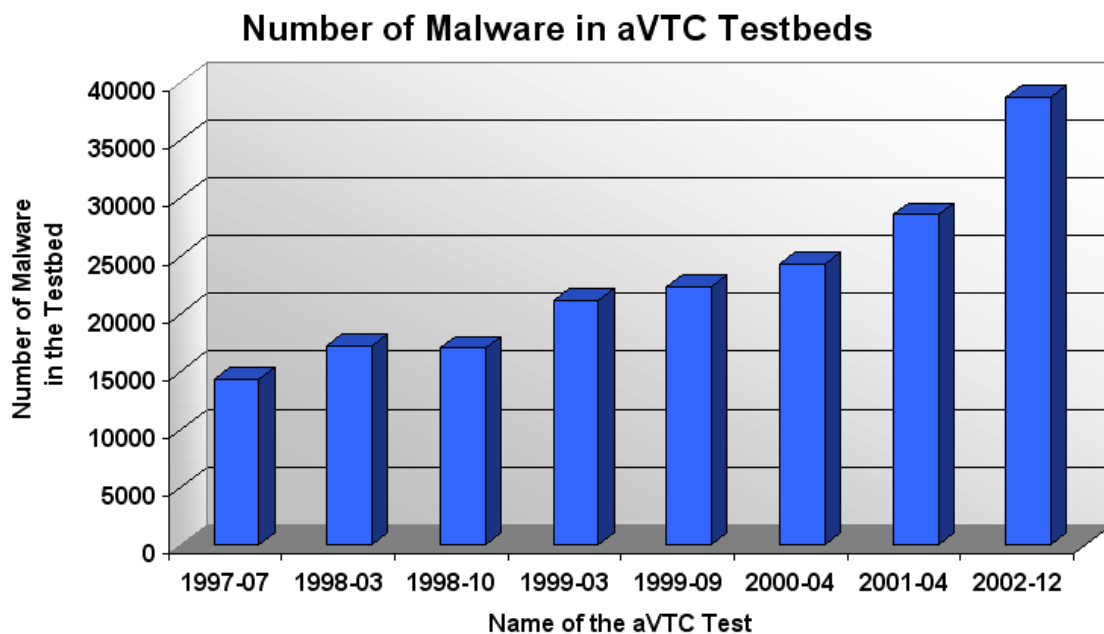


Figure 1: Number of malware used in the testbeds of recent aVTC tests

This growing number of desktop malware will also reach mobile devices. As a result, mobile devices will also be exposed to desktop malware, but they can also be targeted by especially crafted "mobile malware".

That really harmful things can happen when mobile devices become the target of malicious code shows an incident that occurred in Japan in the summer of 2001: A malicious e-mail send to users of i-mode mobile phones called the emergency number, once the user opened it. This at least caused disruption of the emergency service. For more details, please see a press release of Japan's mobile operator NTT DoCoMo [NTTPR].

This work will look at the security risk of malware contamination and malware distribution on mobile devices that are used without deploying additional third-party applications or tools. It will show which defences exist against malware on the mobile device, how build-in defences can be enabled and especially assesses the detection rate and detection quality offered by additional installable anti-malware products for mobile devices. This will be done in a way that it will be possible to contrast the detection rates of mobile device's anti-malware products with that of desktop products, for which the aVTC has a long history of tests. For more information about the aVTC, please see www.avtc.info [VTCWEBSITE].

The following chapter will further introduce and define the terms used throughout this work.

2 Terms and Definitions

2.1 Towards a “Risk Analysis” of mobile devices

According to an ISO standard risk analysis is defined as “the process of identifying security risks, determining their magnitude, and identifying areas needing safeguards” [ISO13335-1]. Risk in this standard is defined as “the potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to an organization” [ISO13335-1].

To identify the vulnerabilities of mobile devices, their technical aspects (hardware, operating system, applications) are analysed and their security problems, shortcomings and weaknesses are identified. This work will not look at organisational problems.

These security problems, shortcomings and weaknesses might make mobile devices also vulnerable to general security threats, but the threats looked at in detail in this risk analysis are malware contamination (defined in 2.4.6) and malware distribution (defined in 2.4.5). Therefore, the mobile device’s vulnerabilities that can be exploited to make malware contamination or distribution possible are explained in detail (see 6.5). Finally, a quantitative analysis on the risk of malware contamination and distribution on mobile devices is made and the risk’s magnitude is identified as high, medium or low.

This work cannot perform an economical risk analysis; it will not value individual assets and classify potential losses in monetary figures. The values of assets vary from organization to organization, and so I will use a more general definition of asset, again from the ISO standard 13335 part 1, which defines an asset as “anything that has value to an organization” [ISO13335-1].

This work will first identify the points through which malware could enter the mobile device. In addition, the mobile device’s vulnerabilities that can be exploited by the two threats are identified. Then evaluate how the entry points and the exploitation of vulnerabilities will allow the threat of malware to cause harm. As each threat uses certain entry points and certain functionality must be obtainable by exploiting vulnerabilities, I will evaluate the risk’s magnitude by checking the level of fulfilment of the threat’s requirements on mobile devices.

After the magnitude of risk has identified the areas that need protection, several safeguards are analysed, that could help to reduce the risk of malware contamination or malware distribution.

2.2 The Term “Mobile Device” and Related Terms

In this work, a “mobile device” is a hardware device that is able to perform sufficient computing operations and allows user interactions while being moved, not only a device that is “moving”¹ or has the ability to move.

Looking at today’s communication and computing devices a range of devices can be moved from one location to another, some examples are:

- Laptops
- Notebooks
- Sub-Notebooks
- Personal Digital Assistants (PDAs)
- Mobile Cellular Phones
- Transponders

All of the above listed devices can be taken from one place to another, but some of them are not easy to operate while being moved or do not have enough computing power to be considered a “mobile device”.

So to be really considered a mobile device the device shall be easy to interact with, while being moved around by the user, which will eliminate Laptops and Notebooks as they are too big and too heavy to be easily operated. Moreover, it shall have a sufficient amount of computing power to perform computing operations, which will eliminate transponders and some mobile cellular phones.

¹ A look into a thesaurus [ROGETS5] shows that the adjective “mobile” is found just under “moving”.

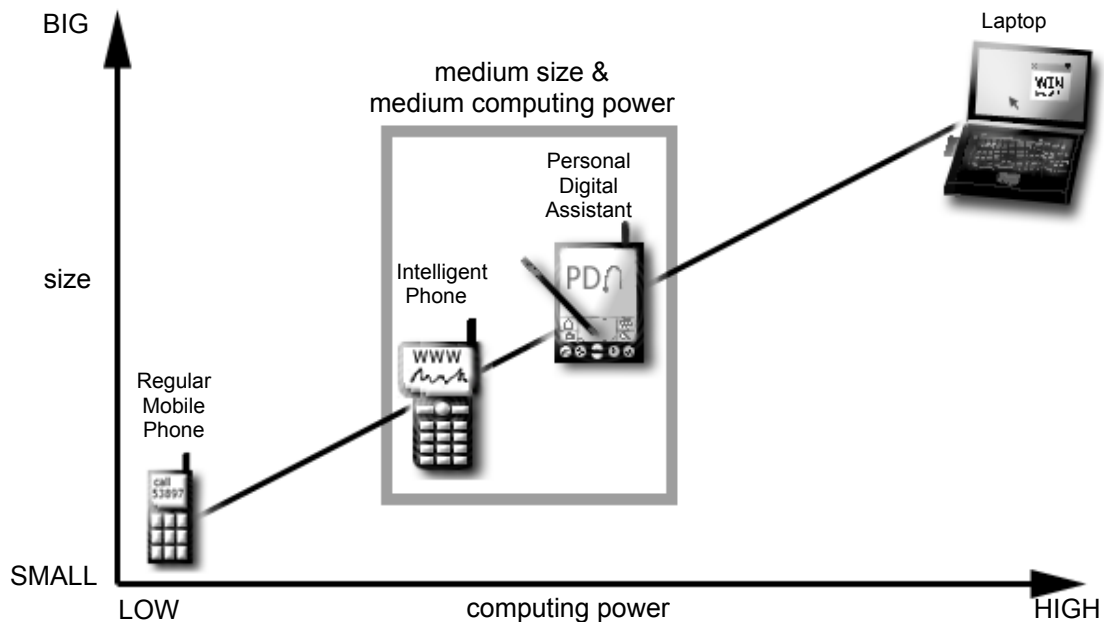


Figure 2: Computing power versus size of mobile device

The smaller the devices are the more mobile they get in the above-mentioned sense, but the smaller they get the less computing power they possess. The trend from this years CeBIT has shown that computing power is getting smaller and smaller. The term Notebooks as shown in Figure 2 shall represent the high-end class of Notebooks, having processors with the same or nearly the same computing power as found in desktop PCs. The other end of the scale depicted in Figure 2 shows a cellular phone of the 2G GSM phones, which has SMS capabilities and a limited amount of memory to store names and telephone numbers.

The devices that balance the two aspects size and computing power best are the Personal Digital Assistants (PDA) and Intelligent Phones, sometimes called “smart phones”. This is the class of medium size and medium computing power.

This work will concentrate on mobile devices with enough computing power to implement safeguards against malware on the mobile device itself, so PDAs and Intelligent Phones are considered to be mobile devices, where all functionality can be used with ease while the device is moved by the user.

There are a lot of different devices in today’s market that fit into the class of medium sized medium computing powered mobile devices, therefore the aspect of the operating system was taken into account to narrow the focus: This work has chosen that the mobile device shall be a Microsoft Pocket PC 2002 device.

So for this work a “mobile device” will be a PDA sized mobile hardware device running Microsoft Pocket PC 2002.

2.2.1 The Term “Desktop”

“Desktop” or “desktop system” is the name used for all the fixed non-mobile computer systems. These computers have high computing power, are not battery powered and are normally constantly connected to a network.

2.2.2 The Term “MS Pocket PC 2002”

“MS Pocket PC 2002” stands for a set of MS Windows CE 3.0 operating system components and application components, which are specially tailored for personal digital assistants (PDAs) [MSDNFAQ].

Therefore, the Pocket PC 2002 operating system (OS) is not equal to the Windows CE 3.0 operating system. There are components, application programming interfaces (APIs) or applications that are only available in the Pocket PC 2002 environment. The next problem is that whilst the same subset of MS Windows CE 3.0 components and applications is installed, the original equipment manufacturers (OEMs) can include special additional applications or drivers in the Pocket PC 2002 in their devices. Comparing one Pocket PC 2002 device with another, they might not be equal in their applications and their OS that reside in their read only memory (ROM).

When I use the term “Pocket PC 2002“, I will refer to Pocket PC 2002 as a platform. As defined in [DEVTOOLS], a platform is “a certain defined hardware plus a set of programs, modules, user interface components, and an operating system” [DEVTOOLS, page 2].

Pocket PC 2002 is specifically targeted to run on PDA devices, so special care has been taken to support the hardware restrictions that are typical for PDAs, such as limited screen estate and battery power to name a few. In [DEVTOOLS] this is seen as “a well-defined hardware (defined by Microsoft and implemented by OEMs such as Compaq and Hewlett-Packard)” [DEVTOOLS].

So additionally to the operating system and the applications, Pocket PC 2002 also specifies some hardware requirements for the mobile device.

Summarizing the Pocket PC 2002 platform can be seen as three parts:

- well-defined Hardware
- Windows CE 3.0 operating system components
- Applications

I will use the term “Pocket PC 2002” as equivalent to “MS Pocket PC 2002” or “Microsoft Pocket PC 2002”.

There are, to make it even more complicated, different versions of Pocket PC 2002, they are described in more detail in chapter 4.2.

2.2.3 Why the Pocket PC 2002 Platform?

There are three main operating systems for mobile devices on market:

- Palm OS
- Windows CE
- Symbian

I will shortly outline why this work has chosen Pocket PC 2002 (based on Windows CE 3.0) as the mobile device platform to analyse for general security and malware contamination risks.

2.2.3.1 Large number of devices and significant growth

According to research by canalys.com [CANALYS2002], the Palm OS still leads in shipped units in the European, Middle East and African region (EMEA) in the first Quarter 2002. But compared to the previous year’s figures (Q1-2001), where Palm OS dominated the market with over 50%, Windows CE devices come much closer to the still leading Palm OS devices in the first quarter of 2002. Figure 3 also shows that only Windows CE based devices show an increase in the number of shipped units from Q1-2001 to Q1-2002.

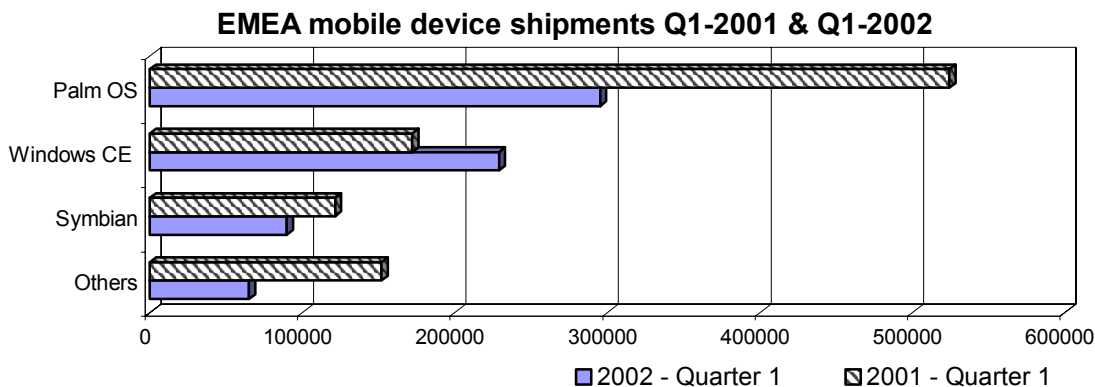


Figure 3: EMEA mobile device shipments for Q1-2002 and Q1-2001 [CANALYS2002]

Pocket PC 2002 devices are just the newest strain of mobile devices using the Windows CE operating system; some of the figures might still include older mobile devices running previous versions of MS Pocket PC.

Other market analysis predict that the mobile devices running on Windows CE will outnumber the devices running Palm OS by the year 2004 [ETFORECAST2002]. The importance of mobile devices running on Windows CE is growing, as is the number of units sold (see Figure 4).

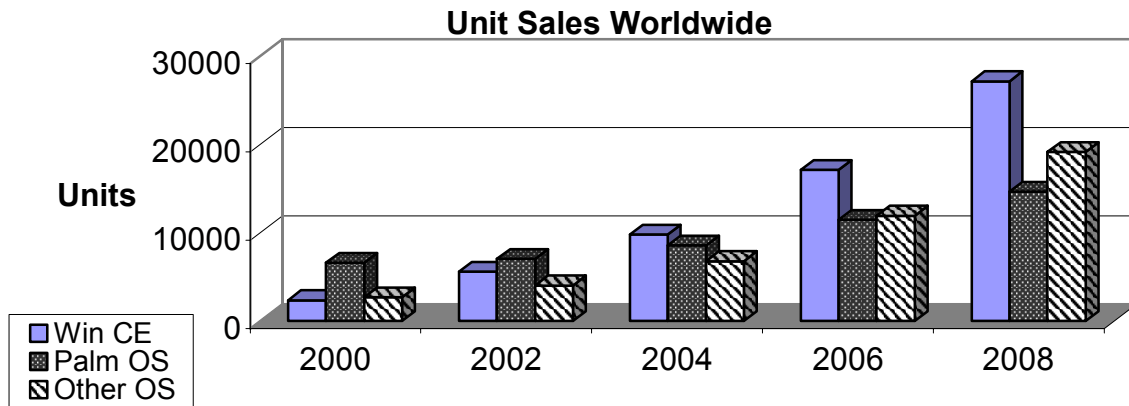


Figure 4: Worldwide Unit Sales Estimates [ETFORECAST2002]

Yet another market analysis by Gartner [GARTNER2003] estimates that 55.2 percent of the worldwide-shipped PDAs in the year 2002 are Palm OS devices, while the share of Windows CE devices is at 25.7 percent.

According to another research by Gartner [GARTNER2002] in the second quarter of 2002, Palm OS PDAs had only 50 percent of the worldwide PDA market and Windows CE PDAs were at 28 percent, which was a 2.6 share points gain compared to the second quarter 2001. It shall be mentioned that sometimes the different analysts define the term PDA in their statistics differently.

So, all market data and forecasts indicate that Windows CE mobile devices are fast growing and their number will increase over the next years.

Today's number of PDAs with Windows CE as an operating system base is quite low compared to the number of PDAs with Palm OS. Nevertheless, it shall be noted, that many of the same operating system components that are in the Pocket PC 2002 mobile device can also be used to build the operating system for embedded devices, as they can run MS Windows CE 3.0 as well.

One example of a device that you would not expect to run Windows is a sewing machine [ARTISTAWEB]. And if a vision of Microsoft [BRAGINKSI2000] comes true the refrigerator will be running an embedded Windows CE web server, allowing the user to check the temperature over the network.

Therefore, the Pocket PC 2002 platform and their security functions, mostly based on Windows CE 3.0, are of interest also beyond the use in personal mobile devices, such as the PDAs. To stay with the picture of a web-enabled refrigerator, this means that if there are security risks in the operating system of MS Pocket PC 2002, this will have implications on the security of the web-enabled refrigerator as well.

However, as the Windows CE refrigerator is not reality today, this work will focus on mobile devices running Pocket PC 2002.

2.2.3.2 Hardware compatibility

As seen before many PDAs that run Microsoft Pocket PC 2002 are in the market today, additionally there are a lot of accessories and extensions available for these mobile devices.

This makes it also an interesting platform, because a lot of modern PC or laptop hardware is already compatible or can be made compatible with the mobile device's hardware and operating system. The Pocket PC 2002 hardware allows connecting to USB, serial or PC CARD² devices, which can be bought from various vendors. Good examples of extensibility with standard PC hardware are the Compaq iPAQ models.

The following example³ outlines the above mentioned hardware compatibility:

With the Compaq PC CARD extension, a Compaq iPAQ can be equipped with an additional PC CARD. An existing Intel WLAN PC CARD hardware can be inserted into the expansion, and, as Intel offers driver support for Windows CE 3.0 [INTELWLANCE], it can be used to connect the iPAQ to a wireless network.

This hardware compatibility is not possible on Palm OS devices, as direct PC CARD connections are not possible. Also newer, especially for mobile devices targeted, hardware, such as SecureDigital or CompactFlash based hardware devices, are compatible with the Pocket PC 2002.

Hardware compatibility makes it possible to integrate MS Pocket PC 2002 based mobile devices into many existing environments, where mobility needs to be added.

For a list of Pocket PC 2002 devices and their specification please refer to chapter 3.2.

2.2.3.3 Software development compatibility

As can be deduced from the number of software available for the MS Windows operating systems, the number of developers with skills to program software for MS Windows operating systems is great.

For an existing Windows developer the MS Pocket PC 2002 platform looks familiar, a lot of the APIs are nearly the same and development tools, like Visual Basic are available for free from Microsoft.

Microsoft offers the following tools for development of Pocket PC 2002 software for download, these tools have been used throughout this work to demonstrate or test certain parameters:

- Pocket PC 2002 Software Development Kit (SDK) [PPC2002SDK]
- Microsoft eMbedded⁴ Visual Tools 3.0 [EVT3]

The eMbedded Visual Tools include eMbedded Visual Basic 3.0 and eMbedded Visual C++ 3.0; these dialects of the well-known programming languages are tailored to develop for mobile or embedded devices running on Windows CE 3.0.

This makes it easy for developers to write code that runs on the mobile device or to port existing software to the new target platform Pocket PC 2002, allowing it to make use of the mobility of the mobile device.

So there is a lot and there will be even more software for Pocket PC 2002 mobile devices.

2.2.3.4 "Windows Look&Feel" User compatibility

Another factor, why I chose Pocket PC 2002 is that it is based on the look and feel of the known MS Windows desktop operating systems. So users that are already familiar with Microsoft Windows operating systems or MS Windows based applications can use a mobile device based on MS Pocket PC 2002 as it offers similar features.

This is very subjective, and some users might still find it hard to cope with an application running on a mobile device, but I expect that the same users that use Windows based desktop computers will use a mobile device once they need the mobility.

Therefore, I assume that the security awareness of mobile users is nearly the same as the security awareness of today's desktop users.

² PC CARD formerly known as PCMCIA, these two terms will be used exchangeable.

³ Just an example, it has not been tested explicitly, but illustrates what shall be possible.

⁴ "eMbedded" with a capital "M" is the way Microsoft writes the name, which has been adopted here for easier recognition of the Microsoft related use.

2.3 The Connection State of a Mobile Device

The mobile device's network connection can be different; to allow an easy distinction I will define four different connection states:

Not connected to any network:

- disconnected

Connected to a network:

- connected
 - wirelessly connected
 - cradled

2.3.1 Connection State: Disconnected

Whilst the mobile device is in "disconnected" state it has no outside connections, it cannot exchange any data with other devices or systems.

As a result, the mobile device can use only self-provisioned functions or data that resides on local storage in this state. If software is able to function in this state, it can run stand-alone on the mobile device.

2.3.2 Connection State: Connected

In this connection state the mobile device is able to communicate with other devices or systems by being connected to a network.

The mobile device can use any form of connection medium to establish the connection. And from the moment were it is able to send any form of data to another device or system the mobile device will have reached the connected state, even if it just initiated a handshake, transferred login data.

2.3.3 Connection State: Wirelessly Connected

This connection state is a refinement of the connection state "connected", as the mobile device is able to communicate with other devices or systems over wireless networks. Any wireless network that is available to a mobile device can be facilitated to be "wirelessly connected".

For example: *Most of the mobile devices have an infrared connection build-in, others have GSM connectivity, wireless LAN (IEEE 802.11) or Bluetooth either build-in or it can be used through add-on cards.*

2.3.4 Connection State: Cradled (Home System & Pairing Process)

While being "cradled", the mobile device is in a special "connected" state. It is not just connected to any system, but to a "home system". This is the standard connection of a mobile device used to transfer data (contacts, files, e-mails, appointments etc.) and install applications.

A mobile device's "home system" is the computer or system with which the mobile device has established a special relationship and with which it synchronizes its data and applications on a regular basis. The mobile device is normally connected to the home system via a cable connection (USB or serial connection) and inserted into a docking station, which is often named cradle. Nevertheless, a cable connection is not required; a wireless connection can also be used.

To facilitate a cradled connection, an initial step is required, which I will call the "pairing process". The "pairing process" is done once a mobile device is first connected to the system that will later become the home system. During the pairing process, a relationship is established between the mobile device and the home system, which is stored on both sides. Mobile devices can have more than one home system, as one system can act as a home system to different mobile devices.

In the Microsoft Pocket PC 2002 environment, the home system is normally a desktop PC running the MS ActiveSync application. More about the ActiveSync connection can be found in chapter 4.4.13.

2.4 The Term “Malware” and Related Terms

I will use the term “Malware” in the most general case as defined in [BRUNNSTEIN1999] as: “Malware may be developed from a given (functional) software or module by intentionally contaminating it with unspecified (hidden) functions. Such malware may consist of combinations of self-replicating or propagating part, or both, which may be triggered by some built-in condition. Malware may include hidden Trojanic functions, which may also activate upon some built-in condition (trigger). The development of malware (in the contamination process, namely the Trojanization) may be observed in cases of self-reproducing software, but it is (at present) difficult to anticipate the malicious Trojanic behaviour before it materializes.” [BRUNNSTEIN1999, page 10]

For this work, I will classify four different types of malware:

- Virus
- Worm
- Trojan horse
- Hostile Applet

2.4.1 The Term “Virus”

A Virus is, in accordance with [BRUNNSTEIN1999], any software that self-replicates on one system. The virus writes its malicious code to normally previously not infected other objects, infecting more and more objects on the local system. For a more detailed definition see [BRUNNSTEIN 1999].

2.4.2 The Term “Worm”

A Worm is any software that propagates itself to other parts of a network, as again defined in [BRUNNSTEIN1999]. Therefore, a “Worm” needs access to a network to reach other systems.

2.4.3 The Term “Trojan Horse”

A Trojan horse is defined in [BRUNNSTEIN1999] as “a software or module that, in addition to its specified functions, has one or more additional hidden functions (called “Trojanic functions”) that are added to a given module in a contamination process (“trojanization”) usually unobservable for a user. These hidden functions may activate depending upon specific (trigger) conditions.” [BRUNNSTEIN1999, page 9]

To emphasize: Trojan horses do not self-replicate like viruses and worms. This definition does not include software, which includes functions that are intentionally harmful, but are not hidden.

As the Pocket PC 2002 class devices are also used by users, not familiar with the inner workings of their device, I would like to include software with only harmful functions or functions that a general user will not expect, and can therefore be seen as “hidden” from the user, into the category of Trojan horses.

A good example for software with an unexpected behaviour is given in [BRUNNSTEIN1999]: *Software that will, when executed, immediately formats the hard drive and which name is “help.exe”.*

I will consider the above software as a Trojan horse, because for nearly all users there is no chance of understanding what help.exe really does other than what can be falsely derived from the name. Therefore, it will be included into the category of Trojan horses, even if [BRUNNSTEIN1999] would not and defines it as a renamed “critter”.

Even though I will include the above-mentioned software into the malware category of Trojan horses, software, which describes what they do to the user (i.e. have a warning screen) will not be included into malware.

For this work I want to further distinguish Trojan Horses, depending on whether the “Trojanic functions” need network access or not:

- Online Trojan Horses: Network access needed
- Offline Trojan Horses: No Network access needed

Online Trojan Horses:

Examples of online Trojan horses are password stealing Trojan Horses, which find and decode passwords, log passwords entered or trick users to reveal a password to them. Then they use an outside connection and send the passwords to the attacker. Also common are so-called Backdoor Trojan Horses, which need a network connection to function properly. These Trojan horses listen for commands from the Internet. This allows the attacker to remotely control the system or trigger other malicious functions. Backdoor Trojan Horses consist of two parts, a server, which is installed and run on the attacked system, and a client, which is used by the attacker.

Offline Trojan Horses:

For example, the above-mentioned help.exe does not require network access, but it must be allowed to access the hard drive in order to format it. Offline Trojan horses existing in the desktop environment often attempt to delete files or create deep directory structures.

In today's networked computing environment the Trojan horses that make use of the network functionality are more dangerous, as they allow the attacker to remotely influence the Trojan horse's behaviour, and to extract information from the attacked system. Offline Trojan horses carry out functions that were pre-defined by the attacker at the time of trojanization

2.4.4 The Term "Hostile Applet"

A hostile applet is software, that is automatically downloaded and executed by Internet browsers and performs malicious tasks, unwanted by the user or makes the user perform such tasks. This definition is taken from [VTCINFO].

The name applet is often only used for Java programs, so called Java-Applets [JAVAAPPLET]. However, this definition of hostile applets includes all programming languages, which allow producing software that is automatically downloaded and executed by Internet browsers. Examples of such programming languages are Java for Java-Applets or C++ for ActiveX-Applets. Additionally the browser's scripting languages are used (i.e. JScript, JavaScript or VBScript). Different from other malware is that hostile applets make use of the browser's functionality to interpret or execute code rather than directly using OS functions.

2.4.5 The Term "Malware Distribution"

If the malware never executes on the platform, but the platform is used as a gateway for transfers (either a transfer to another network or to removable storage media) letting malware in or out without being detected, this will be called "malware distribution".

An example of malware distribution from the Pocket PC environment:

Alice gets an e-mail with a virus-contaminated attachment in Pocket PC 2002 Outlook, for example an executable infected with a virus. She saves the contaminated executable to the storage media on the device for later retrieval. Then later on, Alice sends the saved executable as an attachment in another e-mail to Bob.

Therefore, the Pocket PC device has facilitated the distribution of malware from Alice to Bob.

Distribution does not need to be an automatic task, as illustrated in the above example the virus is spread only because of Alice's actions. As it was not detected when it was saved to the device, the manual distribution by Alice was made possible, but malware distribution can also be an automatic task (i.e. automatic forwarding of e-mail).

Prevention of Malware Distribution

To prevent the distribution of malware it is essential that the malicious objects can be recognized as malware and that they can be differentiated from non-malicious objects.

The scan engine or scanner of an anti-malware product normally performs this recognition. The scan engine scans objects for malicious signs. The information what signs are malicious is stored in so-called virus definitions or patterns. These files are updated regularly by the anti-malware product vendor whenever new viruses are discovered and their particular malicious signs are identified.

The scan finally indicates if malicious signs have been found or not. If malicious signs were found the product also tries to identify the malware found with a name. This is called the scan process or just the scan.

This scan process can be initiated automatically, without user interaction, or manually on the user's request.

After the scan process, appropriate actions must be taken in order to prevent harm. The access to a malicious object can be blocked or the object can be destroyed, to name just a few actions.

The steps of malware distribution prevention shown in Figure 5 can be initiated automatically, every time an object is accessed. So-called "on-access" scanners scan the objects each time an access to an object is requested by the operating system. This access request is fulfilled only after the on-access scanner has taken the appropriate steps. If the scan process is manually started by the user, every object must be scanned at least before it is distributed to other devices. This is called an "on-demand" scan, as objects are scanned on the user's demand.

To effectively prevent malware distribution, these steps need to be carried out before any objects are forwarded to the network (for example by sending an e-mail) or permanently stored on removable media.

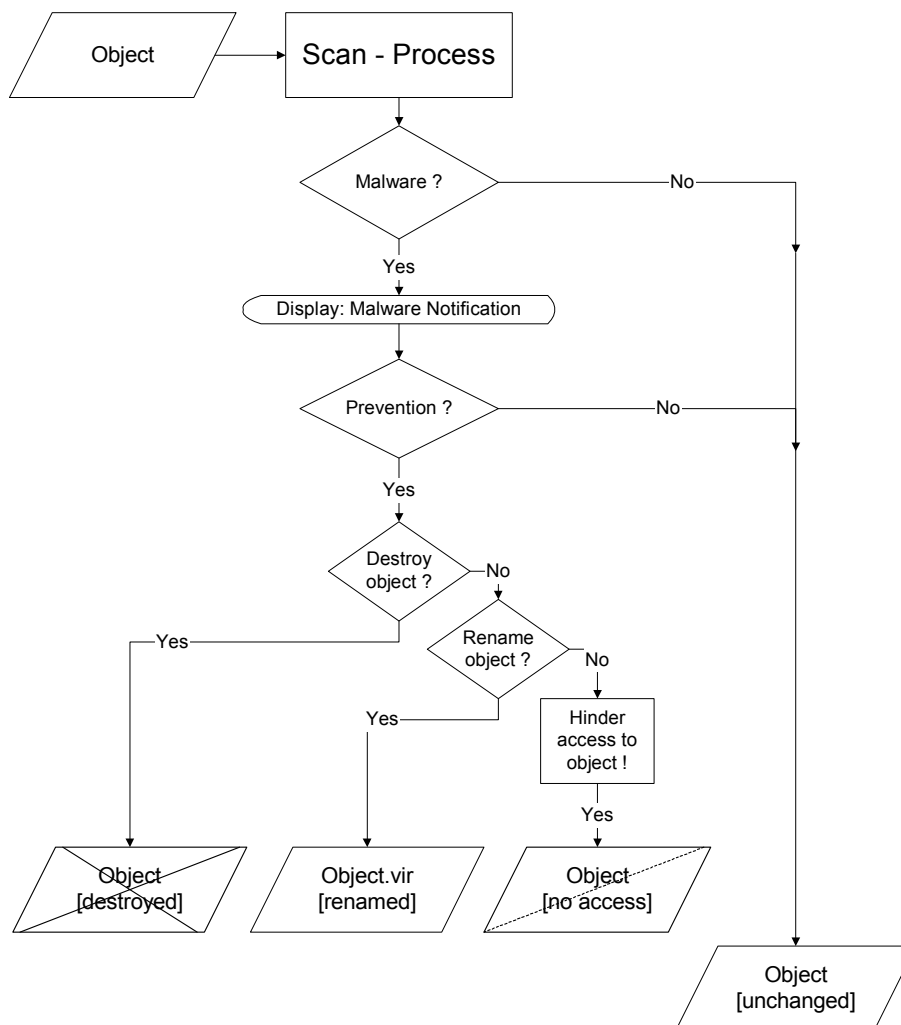


Figure 5: Steps for Prevention of Malware Distribution or Contamination

2.4.6 The Term “Malware Contamination”

The term malware contamination will be used in contrast to malware distribution, when malware executes and does intentionally harmful things on the platform without being detected. If the malicious code of a virus initiates an “infection”, or a worm initiates the “propagation” (as defined in [BRUNNSTEIN1999]), then the mobile is contaminated. If the malicious code is a Trojan horse, then executing the Trojan horse is also seen as contamination, not only the “trojanization”. For hostile applets, the execution of the applet’s malicious code in the browser is the contamination.

The result of an execution of malware is a contaminated mobile device.

To give an example of malware contamination in the Pocket PC world:

Alice again receives an e-mail from Bob, containing a malicious attachment. If the attachment is executed it will delete all Pocket Word documents found on the Pocket PC device and then send itself to all known users from the address book, and install itself, if run for the first time, as a program called PinBall on the device⁵.

Alice executes the malware when she was tricked by the text of the e-mail to click the attachment. Alice's device will from then be contaminated with that malware, as Alice has received no signs of warning that the action has taken place.

Prevention of Malware Contamination

To prevent malware contamination the objects have to be identified by a scan process before they are executed. If an object is classified as malicious, the execution has to be prevented. To prevent execution the steps shown in Figure 5 must be carried out before the file is executed.

As not only executable program files can contain malicious code, but also documents or Internet pages, nearly every object needs to be scanned before it is opened and embedded executable content is executed.

For example: Active content, which is downloaded from the Internet is often automatically executed for display by the browser, it is hard for users to initiate the scan process manually before execution.

So to best prevent malware contamination it shall be an automatically on-access scan process, as an on-access scanner can identify known malware automatically before it is invoked and executed.

2.4.7 The Terms “Testbeds”, “Test Sets” and “Samples”

When testing the quality of anti-malware products, the products will have to show if they can identify malware. A single object containing any form of malware will be called a “malicious sample”. As test are made also using non malicious objects, to see if the anti-malware products do not give false alarms, these objects are simply referred to as “samples”. This work has not performed false-positive tests so all samples in the testbeds used are malicious.

For example: One Word document infected with a Word macro virus is one malicious sample.

The different samples are grouped together into “testbeds”, if they belong to the same group of malware. This work will distinguish between malware written in macro languages and malware written in scripting languages, which will produce at least two different testbeds. To give an example, the above-mentioned infected Word document would be a malicious sample in the macro testbed, whilst a worm written in Visual Basic script would be in the script testbed.

If using all the samples in a testbed at once is unsuited for the task at hand, for example due to the size, then a testbed is divided into two or more “test sets”. Additionally, two test sets do not need to be even in numbers of samples or in their size.

Figure 6 on the next page shows a testbed that is split into two test sets.

⁵ This is pure fiction, as no Malware for Pocket PC 2002 devices is known yet.

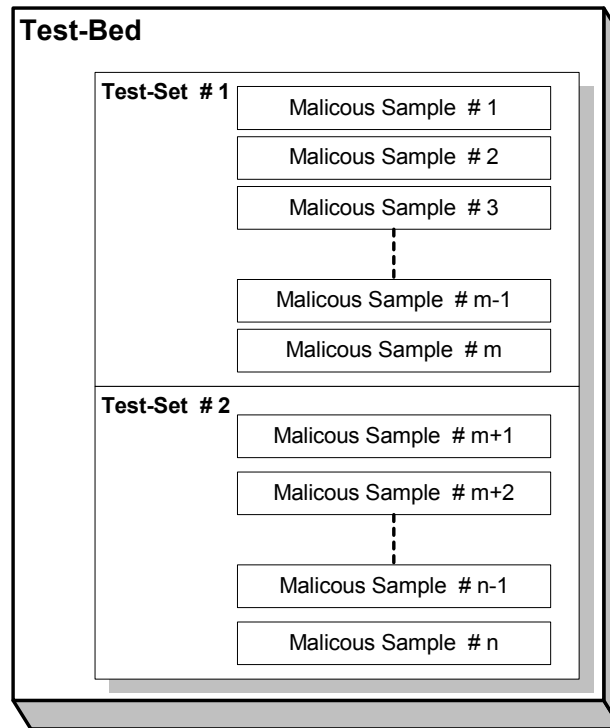


Figure 6: A testbed divided into two test sets, with m and $n-m$ samples in each test set

3 Pocket PC 2002 Hardware

In chapter 3, I will look at Pocket PC 2002 hardware. The hardware is one of the three parts of the Pocket PC 2002 platform. A lot of mobile devices are in the market today, details, also containing pictures, of more than 20 devices are in Appendix B.

Rather than going into details of each device this chapter will provide a minimum hardware specification, list the mobile devices available at the beginning of 2003, and categorize them.

3.1 Pocket PC 2002 Hardware Specification

Although Microsoft does not publish a Pocket PC 2002 minimum hardware specification, all the mobile devices in the market have gone through a compliance test, and then got the logo “Microsoft Windows Powered” [MSWINPOWEREDLOGO].

Original equipment manufacturers (OEMs) have many options when designing the Pocket PC 2002 hardware (see for example [HP3800SPECS] for the iPAQ 3800 hardware), but by comparing different device, it can be reduced to the minimum hardware requirements.

Additionally to the comparison, I have used mainly Microsoft sources to compile the following hardware specification for mobile devices running MS Windows Pocket PC, and having the “Microsoft Windows Powered” logo:

- **CPU:** ARM V4 or ARM V4T [EVT3HELP], [MSWINPOWEREDFAQ]
- **Memory:** At least 24 MB of ROM (flash able) and 16 MB of RAM [EVT3HELP]
- **Buttons:** Power-Button, Up/Down-Control & Action-Button [EVT3HELP]
- **Screen:** Touch-sensitive LCD screen with a size of 240 x 320 pixel in portrait orientation up to 16bit color, the dot pitch of the LCD must be 0.22 to 0.24, the resistive touch panel must have a refresh rate of at least 100 samples per second. [EVT3HELP], [MSHHPFAQ], [MSWINPOWEREDFAQ]
- **Input:** Stylus as the main input device. [EVT3HELP]
- **Communication:** IrDA, Serial Port [EVT3HELP]
- **Notification:** Audio Notification (Beep) [EVT3HELP]

The following external storage devices are supported by mobile devices [DEDO2001]:

- PCCARD (or PCMCIA)
- Multimedia Card (MMC) or Secure Digital (SD)
- Compact Flash (CF)

Earlier versions of Pocket PC, like Pocket PC 2000, supported different hardware platforms. This made it hard for developers and users, because different versions of the same software were needed for the different processors in each device. With Microsoft Pocket PC 2002 only an ARM compatible processor architecture is supported and Windows CE 3.0 uses the ARM instruction set version 4. This allows to consistently develop and use applications across all Pocket PC 2002 mobile devices from different vendors and with different, but instruction compatible, processors.

3.1.1 ARM Processor

The processors of the mobile devices running the MS Pocket PC 2002 operating system are required to be compatible to the ARM architecture, which I will shortly describe here.

The ARM architecture is an enhanced reduced instruction set computer (RISC). Enhancements have been made to allow a high performance, low power consumption, low code size and low silicon size [ARMREFERENCE]. The following additions have been made by ARM to enhance the RISC architecture [ARMREFERENCE]:

- control over the arithmetic logical unit (ALU) and the shifter unit
- auto increment and decrement addressing modes (loop optimisation)
- multiple instruction load/store
- conditional execution of every instruction

ARM processors have developed over the years, since the first ARM processor the architecture and especially the instruction set has undergone development changes. The different architectures with different instruction sets are looked at next.

3.1.2 ARM Architectures and Instruction Sets

The different ARM architectures can be differentiated by their instruction set [ARMREFERENCE]. Six ARM architectures exist today; the sixth version will probably be incorporated into future mobile devices, as the product rollout of ARM v6 has begun in the year 2002 [BRASH2002].

I will shortly list some information on the different versions of the instruction sets from version 1 to version 6 and their variants (see [ARMREFERENCE] and [BRASH2002]):

- Version 1 and
- Version 2 of the ARM instruction set are obsolete today.
- Version 3 introduced 32bit registers and new exception handling registers.
 - Version 3 variant M: support of 32bit multiplications with 64bit results (included in all higher versions)
- Version 4 added among other features a privileged processor mode
 - Version 4 (and above) variant T: “Thumb”, a set of shortened (16bit instead of 32bit long) instructions to achieve a higher code density
- Version 5 adds, among others, functions for faster integer division
 - Version 5 variant TE⁶: enhanced digital signal processing functions
 - Version 5 variant J: improved execution of JAVA ByteCode
- Version 6 includes the Version 5 TEJ, and adds:
 - Version 6 SIMD: Single instruction multiple data (SIMD) instructions, for increased multimedia application speed.
 - Version 6 variant Trusted Zone: Adds security functions to the hardware, allowing to distinguish trusted and untrusted code and run it with different privileges (more details in [ARMTRUST]).

Interestingly the latest ARM architecture allows incorporating security functionality inside the hardware. But according to a press release [ARMTRUSTPR] this will be available to “licensing in ARM CPU cores in 2004”, so it will take some time until mobile devices will incorporate processors with this security functionality.

3.2 Available Pocket PC 2002 mobile devices

For this work, I only looked at the mobile devices available in January 2003. The mobile devices are then categorized by their processor architecture (see 3.2.2) and by their CPU clock and RAM size (see 3.2.3). For more details on each device, please see Appendix B.

3.2.1 Short list of mobile devices

A short list of most mobile devices available in the market by January 2003:

- ASUS MYPAL A600
- Casio Cassiopeia E-200
- Compaq iPAQ H1910 / H1915
- Compaq iPAQ H3765
- Compaq iPAQ H3830 / H3850 / H3870
- Compaq iPAQ H3950 / H3970
- Compaq iPAQ H5450
- Dell AXIM X5 Standard / Advanced
- Fujitsu Siemens Pocket LOOX 600
- HP Jornada 565 / 568
- HP Jornada 928 WDA
- NEC MobilePro P300
- O2 XDA / T-Mobile MDA
- Toshiba e310 / e570
- Toshiba e740 / e740 BT / e740 WLAN
- ViewSonic Pocket PC

⁶ Variant E is only allowed, if a Thumb (T) instruction set is also used.

3.2.2 Categorization by processor architecture

Processors based on the ARM architecture can be incorporated into many PDAs not only limited to the Pocket PC 2002 devices, which all build on processors with ARM architectures (see [ARMWEB]). For example since the release of Palm OS version 5, ARM architecture based chips can also be used for PalmOS based PDAs.

The core operating system MS Windows CE 3.0 can run on a lot of ARM based processors [MSCEPROCESSORS], as the processor only needs to support the instruction set version 4.

The mobile devices as listed in 3.2.1 are built on three different processors and use two different ARM instruction sets:

- Intel StrongARM SA-1110 (ARM v4)
- Intel XScale PXA250 (ARM v5TE)
- Texas Instruments OMAP 710 (ARM v4)

According to the instruction set the processor of a mobile device understands, the mobile devices will be grouped into two groups:

- **First generation of mobile devices**
ARM version 4:
 - Intel StrongARM SA-1110
 - Texas Instruments OMAP 710
- **Second generation of mobile devices**
ARM Version 5:
 - Intel XScale PXA250

The first Pocket PC 2002 devices that were available in the market incorporated the StrongARM SA-1110, these, together with other ARM version 4 device, will be referred to as the “first generation” of mobile devices. The newer devices build on Intel’s application processors named XScale. In particular, the Intel XScale PXA250 [XSCALE] and probably other ARM version 5 devices are used in the “second generation” of mobile devices.

3.2.2.1 Intel StrongARM SA-1110

Intel StrongARM SA-1110 is specially targeted for “portable wireless multimedia devices” [STRONGARM] and belongs to the ARM version 4 family [INTELSA1110WEB]. In summer 2000 the development board was made available by Intel [INTELSA1110DEV], at about the same time the first devices were in the market. Earlier versions of mobile devices, running MS Pocket PC 2000, were also based on ARM processors; some of them (for example the Compaq iPAQ 36xx) can be updated to run MS Pocket PC 2002.

3.2.2.2 Intel XScale PXA-250

The next chip on the list is the XScale PXA250 [XSCALE] also from Intel. Compared to the StrongARM it is already build on the next ARM architecture generation (ARM Version 5), with the instruction set ARM v5TE. Therefore, it allows shortened “Thumb” instructions (T) and has enhanced Digital Signal Processing functions (E). As it is backward compatible to older ARM architectures, Pocket PC 2002 runs also on this ARM architecture.

3.2.2.3 Texas Instruments OMAP 710

The last chip incorporated into mobile devices available in the market at the beginning of 2003 is the Texas Instrument OMAP 710 chip [OMAPWEB].

The HP Jornada 928 WDA is build on the TI OMAP710, which incorporates an ARM 7 MMU and an ARM 925 which Texas Instruments has enhanced with a DSP chip from Texas Instruments to provide the cellular phone functions among other enhancements [JORNADAPR].

Just as a side note: The first MS Smartphone 2002, the SPV from the UK operator Orange is also build on an OMAP ARM 710 processor [SPVSPEC].

The ARM 925 is part of the ARM 9 core family [ARMCOREWEB], which uses the ARM version 4 instruction set. Texas Instruments has further enhanced it with DSP capabilities.

3.2.3 Categorization by CPU clock and size of RAM

The mobile devices in the market can also be categorized by their CPU clock speed and their RAM size. First generation devices mostly have an Intel StrongARM processor running at a CPU clock speed of 206 MHz. Only the Jornada 928 WDA is an exception to this rule, integrating a Phone it uses the TI OMAP chip and runs only at 132 MHz.

The second-generation devices mostly have a higher CPU clock speed of 300 MHz or 400 MHz. The Compaq iPAQ 191x model is an exception here, it is especially targeted for the low cost market and operates an Intel XScale at only 200 MHz. Probably to not compete with Compaq's own upper range products iPAQ 5450 or 39xx.

	132 MHz	200 MHz	206 MHz	300 MHz	400 MHz
32 MB RAM			iPAQ 3830 Jornada 565 NEC P300 O2 XDA T-Mobile MDA Toshiba e310		
64 MB RAM	Jornada 928	iPAQ 191x	Casio E-200 iPAQ 3765 iPAQ 3850 / 3870 Jornada 568 Toshiba e570	Dell X5 Std. ViewSonic V35	ASUS A600 Dell X5 Adv. iPAQ 3950 / 3970 iPAQ 5450 Pocket LOOX Toshiba e740

Class:	Processors:
1 st generation:	Intel StrongARM SA-1110
	Texas Instruments OMAP 710
2 nd generation:	Intel XScale PCA250

Table 1: Grouping of mobile devices by processor speed and RAM size

It can be observed from the above table, that the second generation devices are all equipped with 64 MB RAM, and apart from one device especially targeted for the lower segment (iPAQ 191x), they run at higher CPU speed than first generation devices.

4 Pocket PC 2002 Operating System

After having looked at the Hardware used in Pocket PC 2002, this chapter will look at the operating system. The Pocket PC 2002 OS, as described in 2.2.2 is based on components from the MS Windows CE 3.0 OS, this work will concentrate on those MS Windows CE 3.0 components that are available in the Pocket PC 2002 device. Therefore, when I speak of MS Windows CE 3.0 then the complete set of components available to OS developers is referred to in general. Only where explicitly stated this functionality is not included in the Pocket PC 2002.

This chapter will take a short look at the predecessors and at the different versions of Windows CE, and then drill down into different components of the Pocket PC 2002 operating system: core functions (see 4.3), network functions (see 0), power management functions (see 4.5) and finally the security functions (see 4.6).

4.1 Windows CE Versions

Like most operating systems the version 3.0 of the Microsoft Windows operating system, which is the basis for Pocket PC 2002, has predecessors. I will only very shortly look at the different versions. Most interestingly, Microsoft has already released a successor to Windows CE 3.0, named CE.net. MS Windows CE.net 4.2 introduces among other changes the concept of the .net-framework to mobile devices [MSCEDOTNET].

This work will not further look at this new operating system, and only consider the mobile devices available in January 2003. These devices all come installed with Pocket PC 2002, although some of them might be upgradeable to the new "Pocket PC 2003" based on Windows CE.net.

4.1.1 Windows CE Version 1.0

Windows CE version 1 was released in November 1996 [MSCE1]. It was run on devices named handheld PC (HPC) and already offered the office application programs Pocket Word and Pocket Excel, additionally to providing a personal information manager (PIM).

4.1.2 Windows CE Version 2.0

The next version emerged about one year later in September 1997 [MSCE2]. It introduced real-time deterministic task scheduling and ran on different processor architectures: ARM, MIPS, PowerPC, StrongARM, SuperH and x86. It supported 32-bit color screens and increased connectivity support with SSL 2.0 and SSL 3.0. The devices were named palm-sized PC.

Version 2.0 also had some updated versions 2.1 and 2.12.

4.1.3 Windows CE Version 3.0

Windows CE 3.0 is the operating system base of the Pocket PC 2002 and Smartphone 2002 mobile devices that are in the market today. It was released in June 2000 [MSCE3] and was also the operating system of the older Pocket PC 2000 devices.

In version 3.0 some of the prior limitations of Windows CE were relaxed (see [MSCECOMPARISON] for a greater list of differences). For example the number of priority levels was increased from 8 to 256 (see 4.3.1.1 for priority level details), or the number of objects in the object store was increased from 65,536 to 4.19 million allowed objects.

It also introduced some new security features like restricting access to critical APIs or restricting write access to parts of the registry, by introducing what I call a "trust-model", which is not available in Pocket PC 2002 (see 4.6.6).

4.1.4 Windows CE.net

Windows CE.net or version 4 of Windows CE was released January 2002 [MSCE4]. The latest version is 4.2. A lot of features are newly introduced, among are new networking functionality like Bluetooth support, and increased security protocol support, like supporting TLS (SSL 3.1), IPsec L2TP VPN, or Kerberos. A detailed list of what features are new in Windows CE 4.x can be found in [MSCECOMPNET] and [MSPPC2003].

Windows CE.net 4.2 is the base of the new platform "Windows Mobile 2003 for Pocket PC", also unofficially known as "Pocket PC 2003", which was just recently announced [MSPPC2003]. No new Windows Mobile 2003 devices are available on the European or German market at the time of writing, but some of the hardware could be upgraded to the new operating system.

4.2 Versions build on Windows CE 3.0

To make it even more problematic to get an overview of what can be seen as the Pocket PC 2002 platform, as described in 2.2.2, there are different versions of Pocket PC 2002. Additionally Smartphone 2002 is also build on the Windows CE 3.0 foundation. Starting at the one end from a Pocket PC 2002, which is a PDA, over the Pocket PC 2002 Phone Edition to the Smartphone 2002 on the other end, the PDA converges more and more into a mobile phone.

4.2.1 Pocket PC 2002: Professional, Premium & Phone Edition

Pocket PC 2002 would be the standard version of Pocket PC 2002, including the basic operating system in the ROM. Additional components have been added to the larger ROM, also with the idea of a smaller RAM memory usage in mind. To roughly compare the three Pocket PC 2002 Editions I have selected certain components (see also [EVT3HELP]) trying to illustrate the main differences in Table 2.

Component / Feature	Professional Edition	Premium Edition	Phone Edition
Transcriber Input		In ROM	In ROM
Terminal Services Client		In ROM	In ROM
Windows Media Player		In ROM	In ROM
Microsoft Reader		In ROM	In ROM
MSN Messenger		In ROM	In ROM
WAP Support		In ROM	In ROM
Cell Core			In ROM
Phone Dialler			In ROM
SIM Manager			In ROM
SIM Inbox Transport			In ROM

Table 2: Difference between the three Pocket PC 2002 editions [EVT3HELP]

Pocket PC 2002 Professional Edition is optimised for size, as it is targeted as an update version for older Pocket PC 2000 devices like the H36xx or H31xx iPAQ series of Compaq (see [EVT3HELP]) or devices, which are marketed at the entry-level (see [HP1910FAQ]). A blank field in Table 2 does not always mean that it is not possible to run the component or application. Sometimes it is possible to install certain applications in RAM (see [HP1910FAQ]).

The Phone Edition is used in all the devices that incorporate a GSM mobile phone into the mobile device. To allow the operating system and its applications to take advantage of the phone capabilities additional operating system APIs are included (for example the Cell Core component in Table 2). The user gets the possibility to use the mobile phone as a modem for wireless connections. Included is also an application to make phone calls and for sending and receiving SMS through Pocket Outlook.

4.2.2 Smartphone 2002

With Smartphone 2002, Microsoft allows mobile phones to be based on a Windows CE 3.0 operating system as well. Smartphone 2002 offers mobile phone capabilities rather than being a PDA, but allows developers to build applications on the same operating system basis as for Pocket PC 2002. Looking at the device hardware, they do not offer touch screens and have a far smaller screen size.

The only Smartphone 2002 device in the market in January 2003 is the SPV build by HTC and sold by Orange (see the press release [SPVPRESS] and the specification [SPVDEV] for further details). Other mobile phone operators have announced that they will begin to offer mobile phones running Smartphone 2002 as well, but no other device than the SPV from Orange is in the shops at the beginning of 2003.

The Smartphone 2002 implements what I called the “trust-model” of Windows CE 3.0, to control the access rights of applications. More information can be found in chapter 4.6.6.

4.3 Pocket PC 2002 core functions

In this chapter the functionality of the Pocket PC 2002 operating system, which is essential to the operating system and to other applications, will be looked at.

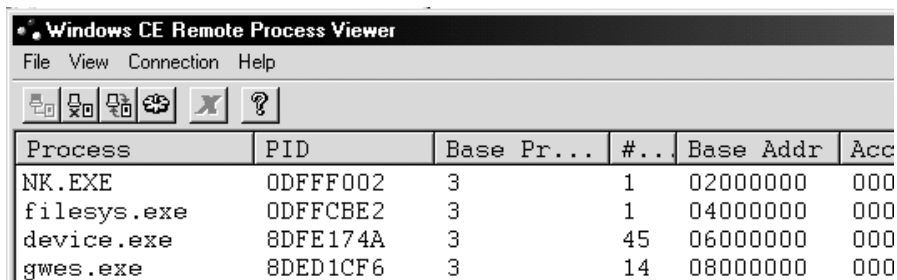
Most of the information for this chapter is taken from the MSDN website [MSDNWEB] or the help files [EVT3HELP].

The kernel, residing in the `core.dll` module, provides the following functions: process and thread management, memory management as well as limited file management (called memory-mapped files). Also essential to applications are the storage handling, the user interface and multimedia or interprocess communication functions.

4.3.1 Process and Thread Management

The Windows CE 3.0 operating system runs each program in a separate process. Windows CE can manage a maximum of 32 processes at one time. When the loader calls the `WinMain` function of the program the primary thread is created. Each process contains at least one such primary thread, but the program can create additional threads, which number is only limited by the amount of random access memory (RAM).

The operating system also runs in different processes, so, application programs cannot use all of the 32 process slots. With the start of Pocket PC 2002 the kernel (`NK.EXE`), the device drivers (`device.exe`), the file system (`filesys.exe`) and the Graphics, Windowing, and Events Subsystem (`gwes.exe`) are all started as different processes. Figure 7 shows these processes running on the test Pocket PC device (T-Mobile MDA) after a reset.



Process	PID	Base Pr...	#..	Base Addr	Acc
NK.EXE	0DFFF002	3	1	02000000	000
filesys.exe	0DFFCBE2	3	1	04000000	000
device.exe	8DFE174A	3	45	06000000	000
gwes.exe	8DED1CF6	3	14	08000000	000

Figure 7: View of the running processes on a Pocket PC 2002

Therefore, Windows CE applications are limited in the number of processes they can use, but each process can hold as many threads as memory permits.

Running each program in a separate process with the restrictions that apply to the access of memory (see chapter 4.3.4) prevents that a malicious or erroneous process can interfere with other processes.

4.3.1.1 Scheduling and thread priority levels

Windows CE 3.0 uses a priority-based time-slice algorithm to provide pre-emptive multitasking to the threads. Prior versions of Windows CE only allowed eight different priority levels for the threads, with version 3.0 of Windows CE there are 256 different thread levels, with zero being the highest and 255 the lowest priority (see Table 3). To set the priority a program can call `CeSetThreadPriority`, to query the actual priority level the function `CeGetThreadPriority` can be used.

The eight priority levels 255 through 248 are used for normal application program threads, and are equivalent to the eight priority levels from earlier Windows CE versions. The higher levels (247 through 0) are reserved for real-time applications, drivers and system processes, these priority levels can be restricted by the original equipment manufacturer (OEM), so that they cannot be used by normal applications.

Only the thread's priority is relevant for the scheduler, as the process that the thread belongs to does not influence the threads priority.

Number*	Priority	Description
0 - 247		Highest possible priority often reserved for OEM use
248 [0]*	THREAT_PRIORITY_TIME_CRITICAL	Indicates 3 points above normal priority
249 [1]	THREAT_PRIORITY_HIGHEST	Indicates 2 points above normal priority
250 [2]	THREAT_PRIORITY_ABOVE_NORMAL	Indicates 1 point above normal priority
251 [3]	THREAT_PRIORITY_NORMAL	Indicates normal priority for applications
252 [4]	THREAT_PRIORITY_BELOW_NORMAL	Indicates 1 point below normal priority
253 [5]	THREAT_PRIORITY_LOWEST	Indicates 2 points below normal priority
254 [6]	THREAT_PRIORITY_ABOVE_IDLE	Indicates 3 points below normal priority
255 [7]	THREAT_PRIORITY_IDLE	Lowest possible priority

*) Windows CE 2.0 priority levels in brackets

Table 3: Windows CE thread priority levels

Pre-emption of a thread is based on the priority: higher priority threads run first and threads of the same priority are allocated processor time in a round-robin fashion. The low priority threads do not run until all higher priority threads have either finished or are blocked. They are interrupted, if a blocked thread of higher priority becomes unblocked.

Blocking of a higher thread occurs when a lower thread uses an object, which can only be used by one thread at a time, and the high priority thread also needs access to this object. Then Windows CE 3.0 changes the priority of the low priority thread to the priority of the high priority thread. This is called priority inversion. Priority inversion in Windows CE is restricted to a depth of one level.

Threads of the same priority are scheduled round robin. Each thread will be allowed to use the processor for a time slice, which is called a thread quantum or just quantum. A quantum is normally 100 ms, but can be set to another value by the OEM in the OEM adaptation layer (OAL), which is then the default processor time for a time slice throughout the operating system. To enable an application to flexibly adapt this default scheduling quantum the quantum can be set to a different value for each thread by the application⁷. Another function that allows adjusting the timing of a thread is to call the sleep function.

The sleep function suspends the execution for a specified interval in milliseconds⁸, if called with zero as wait time the thread returns control to the scheduler and allows other threads of equal priority to be scheduled next. The timing functions are depending on an uninterrupted normal scheduling of the thread. I will look at how Windows CE 3.0 creates and handles interrupts soon (see 4.3.1.3). To have a thread waiting until something has happened in another object a wait function can be used.

4.3.1.2 Wait Function

The wait functions⁹ allow blocking a thread until the specified synchronization object's state is signaled, so that threads can be synchronized (i.e. one thread can wait for the termination of another thread).

Windows CE 3.0 allows waiting for other processes and threads and some special synchronization objects (event, mutex, semaphore), which I will describe a little bit later (see chapter 4.3.2). All these objects are addressed by a handle and the wait function will block the thread either until the synchronization object's state changes from non-signaled to signaled. To prevent threads from waiting forever and causing deadlocks, the wait functions allow to specify a timeout value in milliseconds, after which the wait function will unblock the thread even if the awaited object's state has not changed to signaled. That a time out has occurred can be seen in the return value. Infinite waits are possible with setting the timeout value to `INFINITE`.

The return value of a wait function has four different values, catering for invalid handles, successful wait, wait time-out and one for an abandoned mutex (see chapter 4.3.2.2).

Waiting for signals is one reason for a thread to become blocked and then pre-empted, but also external events can interrupt the processing of a thread, which is done by interrupt handling.

⁷The functions `CeGetThreadQuantum` and `CeSetThreadQuantum` are available in Windows CE 3.0, but there might be implementation differences by different OEMs up to no support of these functions [EVT3HELP]

⁸ A call of `Sleep(1)` delays for $1 > n < 2$ ms, under normal circumstances [MSDNWEB]

⁹ There are three different wait functions: `WaitForSingleObject`, `WaitForMultipleObjects`, `MsgWaitForMultipleObjects` [EVT3HELP]

4.3.1.3 Interrupt support

To respond to external events Windows CE 3.0 uses interrupts, to be able to respond in a performance saving way Windows CE 3.0 splits the interrupt handling in two parts, an interrupt service routine (ISR) and an interrupt service thread (IST).

The first part is the interrupt service routine (ISR), which is a subroutine in the OEM adaptation layer (OAL), running at kernel level. There is one ISR (logical interrupt) for each physical interrupt (IRQ), which will return the interrupt identifier (interrupt ID) to the kernel. The ISR should do only a minimum of processing and is associated with the IRQ by the device driver at system boot [MSDEVCON01]. At system boot the associated IST is also started, but is blocked as it only sits there waiting for the interrupt to occur and the event to be signalled.

While the ISR is executed, all ISRs with the same or lower priority are blocked. In other words, only interrupts with a higher priority can interfere the interrupt processing, this concept is called “nested interrupts”.

According to the returned interrupt ID an associated thread, the interrupt service thread (IST) is unblocked by signalling the associated event and then does the required processing. This thread, running at user mode with a priority of preferably above normal (priority# 250, see Table 3) [MSDEVCON01], does the processing required for this interrupt and handles the event. Until the IST releases the interrupt, this interrupt is masked.

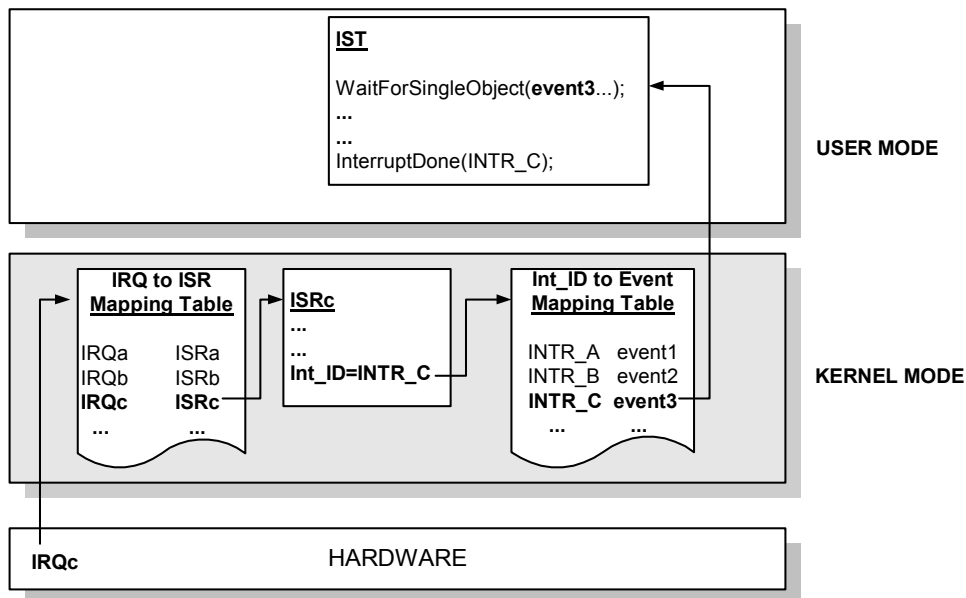


Figure 8: Interrupt handling in Windows CE 3.0

With this splitting Windows CE 3.0 tries to balance performance against ease of use. Another performance issue is the handling of higher priority interrupts. To not delay or lose interrupts of higher priority, Windows CE 3.0 allows so-called nested interrupts. This means that a higher priority interrupt disables all interrupts of the same and lower priority and it suspends the handling of any lower priority ISR, the kernel saves the current state of the lower ISR. Then the higher priority ISR is called and handles the high priority request. After that the processing of any suspended ISR is continued. All hardware interrupts can be nested in such a way.

4.3.2 Synchronizing Processes and Threads

The different threads of one process need to be able to synchronize their activities. Windows CE 3.0 has several synchronization objects that can be used for inter-process or inter-thread communication:

- Critical Section Objects
- Mutex Objects
- Event Objects
- Semaphore Objects
- Interlocked Functions

I will look at each of these, but some are especially interesting as they use memory or variables that can be shared between processes. Processes normally run in different memory slots (see 4.3.3 for more details on memory management), which means that they cannot access each other's variables.

4.3.2.1 Critical Section Object

Threads can register critical section objects (calling `InitializeCriticalSection`), which are sections of memory, which can only be accessed by one thread at a time. Critical sections cannot be shared amongst processes, but allow to protect the memory sections against the use by other threads of the same process.

A thread acquires the ownership over a critical section by calling `EnterCriticalSection` with a pointer to the critical section, which was registered before. The order in which the operating system will allow different threads to obtain ownership is fair [EVT3HELP]. If the critical section is in use by another thread, the thread calling `EnterCriticalSection` will be blocked, if not it can access the critical section and shall call `LeaveCriticalSection` when it has finished its work. To release any resources (like memory) that were allocated for the critical section a thread can call `DeleteCriticalSection` this shall be done with care, as any thread of the process can issue this command [EVT3HELP]. Deleting a critical section renders it unusable for synchronization, and leaves threads, which wait for the release of the critical section, in an undefined state.

4.3.2.2 Mutex Objects

Mutex objects provide mutually exclusive access to sections of memory between the threads of one or more processes. In a way, they are a multi-process version of critical sections.

With the initial creation of a mutex object (`CreateMutex`), it is given a new name, which can be used by other processes to acquire the ownership of the mutex object. They call `CreateMutex` with that same name to make sure the mutex object exists (`CreateMutex` will return an error indicating that a mutex with this name exists). As the `CreateMutex` function will return a handle to that mutex object, it can be used with the `Wait` function to wait for the mutex object to become available. The thread that owns the mutex object, can release the ownership of the mutex (`ReleaseMutex`), which will signal all waiting threads that the mutex object is free.

The signalling on mutex objects is done automatically, the state is set to signaled when a mutex object is not owned and non-signaled when not.

If a thread terminates without releasing the mutex ownership, the mutex will be called abandoned and waiting threads will get an error message (see 4.3.1.2) when they acquire ownership, it is the other threads' responsibility to react accordingly (i.e. not trust the information in the mutex object, as it comes from a terminated thread).

4.3.2.3 Event Objects

Events must be named to be used for inter-process synchronization. Then they work somehow like mutex objects, but event objects allow for more freedom in the application development. The states signaled and non-signaled can be set freely by the thread that owns the event object.

With the creation (`CreateEvent`), an initial state can be set. The signalling can be further adapted to the application needs, by allowing to build manual resetting events. In manual reset mode an event with a state set to signaled stays in this state until `ResetEvent` is called, even when a thread waiting for this event is released. In automatic reset mode the state of a signaled event automatically becomes non-signaled when a waiting thread is released.

To set the state of an event to signaled, meaning the thread has finished doing what other threads might have waited for, the thread can call `SetEvent`.

To close an event object, the `CloseHandle` function can be called, as an event object, like all synchronization objects, has a handle. With named handles, Windows CE 3.0 maintains a counter of how many open handles exist (count is increased by every call to `CreateEvent`); if the last handle is closed, the event object is removed from the system.

Events are the synchronization objects with the most freedom for signalling states to other threads and to other processes, if they are named.

4.3.2.4 Semaphore Objects

The semaphore concept is known from synchronization in Petri nets to model for example the concurrent access to a limited resource (see [DUDENINFORMATIK] for more on Petri nets).

Semaphore objects are new to Windows CE version 3.0; they maintain a counter between zero and a maximum value, which is defined on creation (`CreateSemaphore`). The counter being zero makes the state non-signaled, meaning all waiting threads are blocked.

Waiting threads are released, when the semaphore counter is bigger than zero; with the release of a waiting thread (the return of the wait function) the counter is automatically decremented by one. If a thread has finished accessing the resource that is limited by the semaphore, it calls `ReleaseSemaphore`, which can increment the semaphore counter even by numbers greater than one¹⁰, but the semaphore counter cannot be greater than the maximum value. Like event objects it can also be named, this allows using a semaphore object among different processes.

4.3.2.5 Interlocked Functions

If two or more threads share a common variable, the access to this variable needs to be synchronized as well. How the threads create or access a shared variable in memory will be examined in more detail later (see 4.3.4.4), for the interlocked functions the variable is referenced by a pointer to the memory address. The four Interlocked Functions¹¹ are executed atomically so the thread executing such a function cannot be pre-empted.

4.3.2.6 File and Device I/O

Under Windows CE 3.0, concurrent threads can make concurrent calls to functions that access files¹². If a section of a file is used by another thread's write access, concurrent access to that section is blocked and the `WriteFile` function will fail. Looking at the parameters for the call to `WriteFile` there is the parameter `lpOverlapped` that, in Windows NT/2000/XP, is used to allow for asynchronous overlapped I/O functions. Windows CE 3.0 does not support asynchronous or simultaneous synchronous calls to `ReadFile` and `WriteFile` from the same thread, allowing a thread to continue, even though the executed I/O operation has not yet finished. Using separate threads, which might overlap in time, asynchronous or simultaneous synchronous access modes can be supported.

4.3.3 Memory Types

The physically available data storage in a Pocket PC 2002 device can come from different memory types. First, there is the read only memory (ROM), where the complete Windows CE 3.0 operating system resides, and then the random access memory (RAM). Additional storage capacity can be provided by external storage cards either being Expansion RAM or Persistent Storage (i.e. flash memory) [INTELMEM].

4.3.3.1 ROM

As mentioned, all the files of the Windows CE 3.0 operating system are stored in the ROM. Also all the applications that come with the device are normally stored in ROM (i.e. MS Instant Message Client or vendor specific applications like iTask [ITASKWEB] installed on Compaq iPAQs). A larger ROM in the Pocket PC 2002 compliant devices (minimum of 32 MB), compared to the devices of earlier Pocket PC versions (only 16 MB), allows for a lot of applications to be stored permanently in the ROM, which means that they remain on the mobile device even if it is reset.

The memory in the ROM module can be compressed, depending on the OEM. If the ROM based application is compressed, it needs to be decompressed and is then executed in RAM. If it is not compressed however, it is directly ran in the ROM, using no additional RAM, this is called execute in place (XiP). To allow to be executed in place the applications and DLLs in the ROM must be page-aligned. The memory holes that are created by aligning the start of each ROM based DLL or application on the next ROM page can be filled with other maybe compressed data. Special tools are available [INTELMEM] for this task.

¹⁰ Even the help [EVT3HELP] gives only the example that with each released wait the semaphore counter is decremented, therefore the `ReleaseSemaphore` will typically be used to increase the semaphore's counter by one. No reason for also allowing greater values is given here.

¹¹ `InterlockedIncrement`, `InterlockedDecrement`, `InterlockedExchange`, `InterlockedTestExchange`

¹² `ReadFile`, `WriteFile` and also mentioned by [EVT3HELP]: `WaitCommEvent`

The objects in ROM are managed by the ROM file system. Windows CE 3.0 introduced the concept that a file located in the object store can overload a file in ROM, which will be looked at in detail in 4.3.5.2.

4.3.3.2 RAM

The RAM of the Windows CE device is logically divided into two areas:

- the object store (storage memory) and
- the program memory (sometimes called “system RAM” [INTELMEM])

The user can change the RAM allocation temporarily in the settings (see Figure 9).

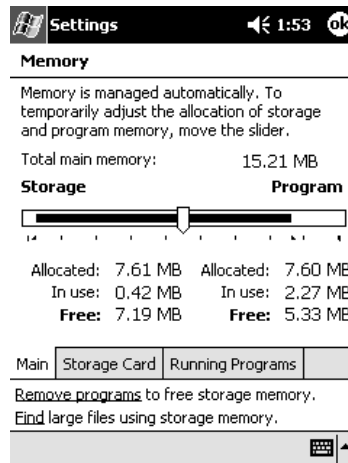


Figure 9: Screen shot from the memory settings dialog

4.3.3.2.1 Object Store

The object store could be seen as a permanent RAM disk, as this is the place where files and other objects are stored. The files are stored using the RAM file system; the object store also is the place where the Windows CE registry and database objects are stored.

So there are three types of objects in the object store:

- Files in the RAM file system
- Registry
- Database objects

As all the data in the RAM would be lost when the battery power is interrupted, many devices have a small backup battery, which powers only the RAM to overcome short power outages (i.e. change of main battery), so that the files saved in RAM are not lost.

More information about the object store is given later in chapter 4.3.5, as the object store is the primary storage for files in Windows CE 3.0.

4.3.3.2.2 Program Memory

The program memory works like the RAM of normal desktop computers; it is used to provide the memory for the running applications (including stacks and heaps). The program memory is allocated using the virtual memory functionality described in detail in the next chapter 4.3.4.

4.3.3.2.3 Expansion RAM

Expansion RAM can be treated like build-in program memory, after it has been detected and mapped into virtual memory during a cold boot [INTELMEM].

4.3.3.2.4 Persistent Storage

In Windows CE 3.0 persistent storage is supported using either the FAT files system (FAT 16 and FAT 32 are supported) or an installable file system. The latter is accessed through a custom file system driver (FSD) that provides the functions to access the special file system. The standard file system is FAT with the `fatfs.dll` as a file system driver.

Flash memory cards and other external storage devices can be used to provide additional storage, but programs cannot be run directly from the flash memory, they are copied into the RAM and then executed in RAM.

More about the access management to storage of Windows CE 3.0 can be read in chapter 4.3.5.

4.3.4 Virtual Memory

Also a kernel service is the virtual memory system of Windows CE 3.0. It allows the applications to allocate virtually contiguous blocks of program memory.

The kernel of Windows CE 3.0 can address up to 512 MB of physical memory. When the operating system is initialised a 4 GB virtual address space is created. The upper 2 GB are reserved for access by the operating system, the lower 2 GB are used to provide virtual memory for applications.

From this 2 GB virtual address space, a little more than 1 GB is divided into 33 so-called slots of 32 MB size. For each process that is started, Windows CE selects an open slot from the address space for that process. The slot number zero is reserved for the currently running process, the original process slot is copied to slot zero, when it is scheduled to run, so that the current process is always addressable through slot number zero and is always running at the same memory location.

The bottom 64 KB are left free as in other Microsoft operating systems, to cater for various incidents, such as not initialised pointers [BOLING1999-2].

When a process is first started the operating system allocates a slot and loads the following information into the allocated 32 MB slot:

- All the dynamic-link libraries (DLLs) used
- Stacks for the primary thread (initial size: one page = 1 - 4 KB depending on the hardware)
- Heaps for the process
- The executable file (EXE) of that process

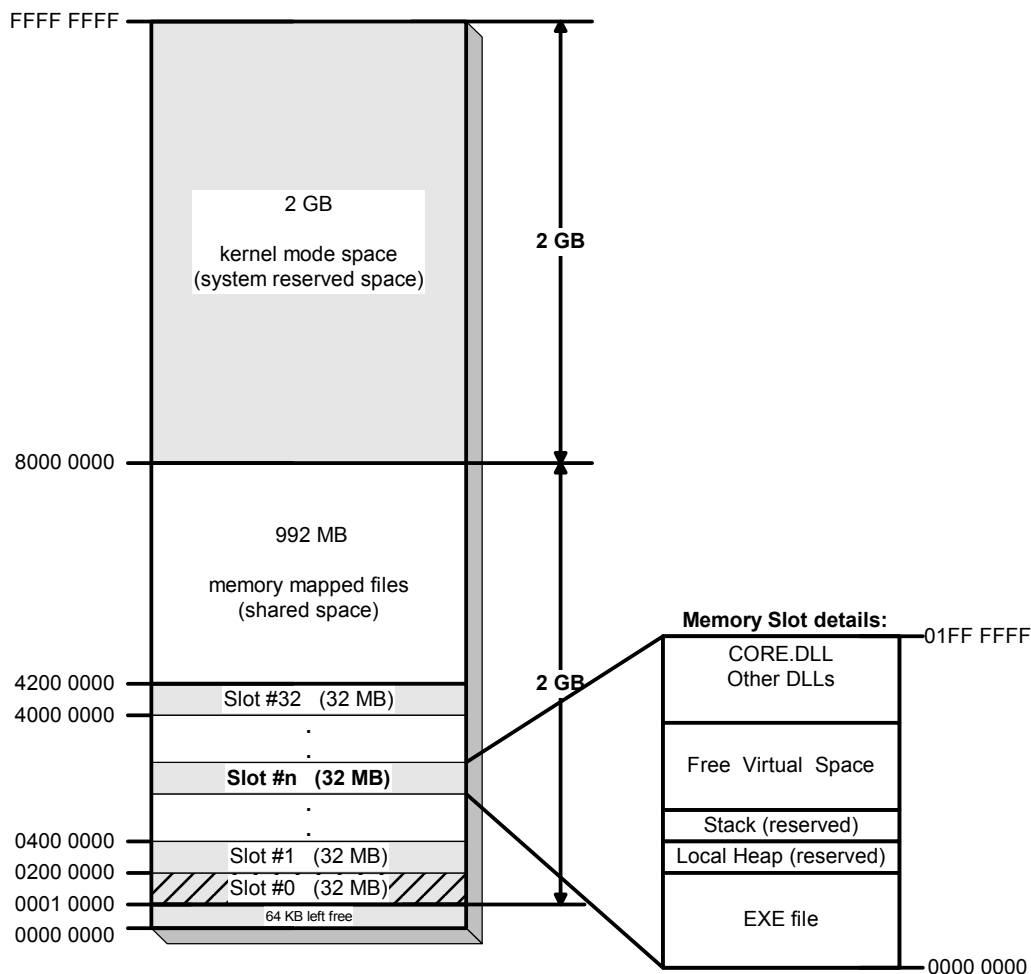


Figure 10: Windows CE memory allocation

The DLLs are loaded and controlled by the loader. The same required DLL is placed at the same virtual address for each process. Loading each required DLL (i.e. coredll.dll) to the same address means that if a single DLL is loaded and used by different processes, it must always be at the same virtual address in all the different processes [BOLING2002]. For example: *The base address of the coredll.dll loaded from filesys.exe and from gwes.exe is the same, as shown in Figure 11.* To guarantee that at the same address within the 32 MB slot no other DLL is loaded, the loader places it below the address of the latest loaded DLL from other processes. As DLLs coming from the ROM, using execute-in-place (XiP), also use up virtual address space, this can be up to 16 MB as [BOLING2002] also graphically illustrates. This problem has been addressed in CE.net using a separate 32 MB space for XiP DLLs.

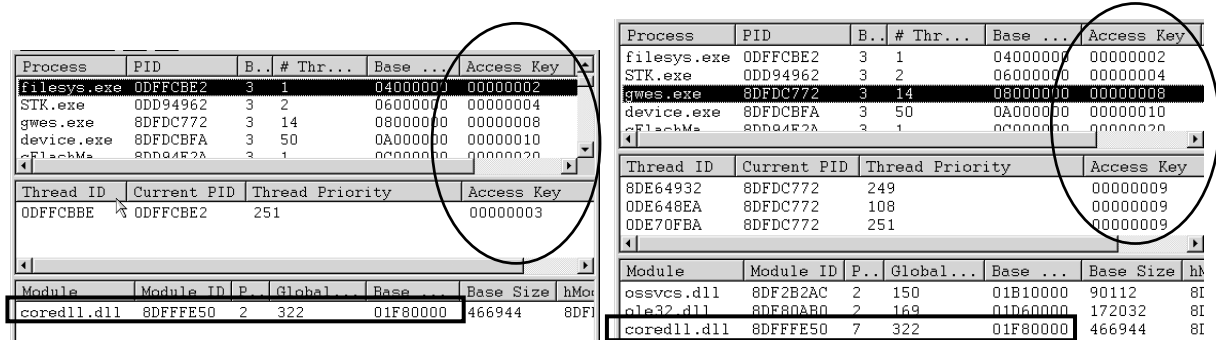


Figure 11: Remote Process View showing: Memory Access Key and DLL Memory Base Address

The rest of the lower 2 GB virtual memory is reserved by the operating system and used for all memory requirements outside the process slots, such as memory mapped objects.

Virtual memory is allocated in pages of either 1,024 bytes or 4,096 bytes size, depending on the processor. A virtual memory page can have one of the following three states:

- Free: not used by an application at any time
- Reserved: held for an application, but not yet mapped to physical memory
- Committed: allocated and mapped to physical memory and in use by an application

Pages allocated, using the `VirtualAlloc` function, are aligned along 64-kilobyte boundaries, also-called 64 KB regions. So, for efficient memory allocation, virtual memory shall be reserved in multiples of 64 kilobytes (i.e. 16 pages on a device with 4,096 bytes pages). Sometimes it is more efficient to use the heap, if the application does not use all of the 64 kilobytes, because as [EVT3HELP] states, an application typically wastes half of the virtual memory page per allocation.

The virtual memory pages all have an access protection flag. Access protection flags are `PAGE_EXECUTE_READWRITE`, `PAGE_READONLY` or `PAGE_NOACCESS` to name just a few of the available flags (see [EVT3HELP] for a complete list). These flags protect the committed virtual memory page against being accessed in another than the specified way.

Each process in Windows CE 3.0 can only use the 32 MB virtual memory from their process slot to satisfy the memory need for the heaps, the stack and the static data block. The only ways to overstep the 32 MB border are memory-mapped objects. Access to memory from other processes is prohibited by the operating system as each process has its own 32bit memory access key (see [GESYTECH] or [MSSEARCHITECTURE] and Figure 11). This is important for security, as two processes can only share data if they agree on sharing defined data (for example using a memory mapped file, see 4.3.4.4).

As memory is a valuable resource, the virtual memory manager also manages how applications use virtual memory in the case the mobile device runs out of free memory.

Pocket PC 2002 defines three low memory conditions: 224 KB, 160 KB and finally only 24 KB of free memory [EVT3HELP]. In the final case of only 24 KB or less memory is available no new application can be opened. When the available memory falls below 224 KB, the Pocket PC 2002 shell will signal running applications that they shall release memory. If they do not (because they cannot or do not react to these messages) and the available memory falls below the next threshold (160 KB) the Pocket PC 2002 OS will send a `WM_CLOSE` message to the application, which will automatically shutdown the application. This comes without a warning, so the application must be able to handle this [PPCLOGO].

Under normal conditions, the user shall not close applications, just open the next application, and whenever memory is a limiting factor the OS will close unneeded idle applications automatically. The design guide for Pocket PC applications [PPCLOGO] also recommends not to implement a close or exit button into the graphical user interface (GUI) at all.



Figure 12: Closing running programs

To completely close an application the user can facilitate the RAM Settings dialog¹³ as shown in Figure 12. Appendix E.3 shows how an application can be written in eMbedded VisualBasic (eVB) that can be only closed using this dialog from the memory settings, as it does not close on “ok” button clicks.

The user can also find out which “normal” applications are running in the background with this dialog, but only applications that have a window with a window name are listed (see [SHARGIN2003] for details). So, this can be seen as an equivalent to the Windows desktop versions task bar at the bottom of the screen rather than a real task manager, which would show all processes.

4.3.4.1 Heaps

In Windows CE 3.0 an application gets a default so-called local heap of 384 pages (on a 1,024-byte page size hardware) reserved. This local heap is initially only reserved, as pages are only committed when they are allocated (see the three states in chapter 4.3.4).

An application can reserve memory on the heap in 4-byte or 8-byte units, again depending on the hardware, this allows for a more efficient memory usage for small allocation than using `VirtualAlloc` to allocate virtual memory directly. Another benefit of using the heap is, avoiding dealing with the different sizes of memory pages on different hardware.

Whenever an allocation of more than 192 KB is requested this attempt is fulfilled internally using the `VirtualAlloc` function. Also the heap might grow over the initial size set by Windows CE, during the runtime of the application, then Windows CE 3.0 will try to find unreserved, free memory, which might not be physically continuous with the original heap memory, which will then lead to a fragmentation of the heap.

Sometimes it is wiser to use several heaps, as an example [EVT3HELP] states a word processor using a separate heap for each opened document file. Several heaps are possible by using the `HeapCreate` and `HeapAlloc` functions, as long as there is enough virtual and physical memory available. You can perform the same functions on the additional heaps as on the local heap.

To allow Windows CE to use unused allocated memory again, it must be deallocated¹⁴ correctly. It should be noted that also for the heaps the virtual memory is used on a per page basis, so that even by deallocating units only completely free pages (of 1,024-byte or 4,096-byte size) can be used for new purposes by the virtual memory system.

¹³ Reachable through: Start → Settings → System → Memory → Running Programs

¹⁴ `HeapFree`, `HeapDestroy` or `LocalFree` are used to free allocated memory from heaps

4.3.4.2 The Stack

The stack is the storage area. The typical stack size available for each thread in Windows CE 3.0 is 1 MB with 2 KB reserved for stack overflow detection. The size of the stack can only be changed at compile time. If the stack limit is exceeded by the application, or the thread, it is aborted by the operating system, normally generating an error message.

4.3.4.3 Static Data Blocks

The static data block contains the strings, buffers and other static values referred to by the application. Two sections, one for read/write data and one for read only data, are allocated. Constants are placed in the read only section of the static data block. For both sections, Windows CE 3.0 allocates memory on a per page basis, so again an application shall carefully watch their memory requirements¹⁵, as they might waste valuable amounts of memory here.

4.3.4.4 Memory Mapped files

Memory mapped objects can be used to share data between processes. It also is the only way for an application to allocate memory outside of the 32 MB process slot.

To enable the operating system to prevent concurrent processes to delete memory mapped objects, which are still in use, the application can name these object in Windows CE 3.0. The memory mapped object is created (with the `CreateFileMapping` function) and given a new name.

This name is then passed to the other process, which can then access the object by calling the same function with the already existing name, given to the object by the first process. The name of the memory mapped object is global so the memory mapped object can be referenced by name also in other processes.

These objects are based on files, an existing file can be opened for mapping, or a complete new file can be created. Additional to the access parameter with which the file was opened for mapping (read or write access) the mapping function allows specifying the access to the memory mapped object again. This access to the memory mapped file must be allowed by the way the original file was opened. The virtual memory pages can be set to read only or to read and write, the latter allowing other processes to change the memory mapped file.

Changes are made in virtual memory, and the operating system then reflects any change of the memory mapped object back to the file opened for this mapping.

4.3.5 Files, Databases and Persistent Storage

The default storage device on a windows CE 3.0 device is the object store. Additionally Windows CE 3.0 allows for up to 256 different installable file systems (IFS). With their file system drivers (FSD), some of the already build-in file system drivers are for FAT12, FAT16 and FAT32. The last is for example used to access the file systems on external storage media like PC-CARDS or CF-cards.

4.3.5.1 Object Store in general

The object store can hold up to 256 MB of compressible, non-volatile memory. It integrates both the read only memory (ROM) and the write- and readable random access memory (RAM). It shall provide the same functionality as a hard drive provides to a normal desktop PC.

The access to the object store is transaction-based, so the operating system ensures that the object store is not corrupted, by interrupted write operations.

For example: If the power is interrupted during a write, the transaction system of the object store can recognize the interruption when power is restored and either undo the operation or complete it if possible.

The undo operation, of course, needs to revert the object store to the last known good state. [EVT3HELP] states that in the case of a corrupted registry this will result in loading the defaults from the initial settings saved in ROM.

¹⁵ [EVT3HELP] advises to use tools like DumpBin.exe or the Remote Memory Viewer

The object store stores the following types of objects:

- Registry keys
- Registry values
- Files (consist of one or more chunks)
- 4-kilobyte so-called chunk of file data
- database records (4 kilobyte)
- extensions of database record (additional 4 kilobyte)
- databases
- database volumes

Each object gets a unique identifier, which can be used to reference the object. For files and directories, for example the object identifier (OID) is returned by the `FindFirstFile` and `FindNextFile` functions used to browse through the file system.

Mounted database volumes additionally get a Windows CE globally unique identifier (CEGUID); it is used in conjunction with the object identifier (OID) to uniquely reference a database object in a database or in the object store.

An interesting remark on the way the operating system assigns object identifiers found in [EVT3HELP]: “If an object in the object store or a record in a database volume is saved to persistent storage and is then restored, the restored object or record is not guaranteed to have the same OID as the original.” Moreover “an object identifier will not be reused for at least 16 object allocations”.

4.3.5.2 File Systems

The maximum size for the object store RAM file system is 256 MB, and a single file in the object store can be up to 32 MB in size. The maximum number of objects in the object store is $2^{22} = 4194304$.

Windows CE offers three types of file systems by default:

- ROM-based file system
- RAM-based file system
- FAT file system

Additional file systems can be installed (installable file system = IFS) through files system drivers (FSD). An example of a file system driver is intel’s Persistent Storage Manager (PSM) [ZEHLER2002] a driver for intel’s flash memory technology¹⁶.

All the storage, regardless of the file system, is accessed via the same application programming interface (API). For Windows CE 3.0 the API calls are equal to the normal Win32 file-system API. Files are opened or created via the `CreateFile` function, which will return a so-called handle for the file. This handle will be used by other file related functions to reference the file.

An application accesses an object through the API; Windows CE then decides according where the object is located, which file system driver should be used to handle the access.

The Windows CE 3.0 file system API allows applications, apart from some limitations, to use the same function calls as in Win32 systems. There are some differences to known desktop file systems:

Firstly Windows CE 3.0 does not use the concept of current directories, all file objects must be referenced by their full path.

Secondly, Windows CE automatically compresses all files in the object store.

Looking at the table of attributes for files [MSDNWEB] in Windows CE 3.0 you can also spot some new attributes like `FILE_ATTRIBUTE_INROM`, which indicates that this file comes from the device’s ROM. This means that you can access a file with this attribute like all the other files from the object store, but they are opened as read only of course. Another file attribute caters for modules, which can be loaded directly from the ROM (XiP), these files need to be opened using special functions¹⁷.

Additional storage beside the object store, as PC Cards or linear flash cards, are supported by the operating system. They can contain one or more volumes, which are mounted separately. Mounted volumes are accessed through installable file systems and their drivers, so mostly FAT is used.

¹⁶ intel StrataFlash [ZEHLER2002]

¹⁷ `FILE_ATTRIBUTE_ROMMODULE`, must be opened with `LoadLibrary` and `CreateProcess`

Windows CE 3.0 does not assign any drive letters, so a folder in the root directory represents a mounted folder from a storage card. The name of the folder is either retrieved from the default name the file system driver (for example the PC Card driver) assigns to it, or it is given the default name “Storage Card”¹⁸.

Additional to having the default directory name “Storage Card *”, directories of mounted volumes can be identified as they have the file attribute `FILE_ATTRIBUTE_TEMPORARY` assigned by the file system driver.

Easiest way to find the mountable storage devices that are accessible in a Pocket PC 2002 system are the functions `FindFirstFlashCard` and `FindNextFlashCard`.

If a removable storage medium is inserted the Pocket PC OS can automatically execute an auto-run application. An `autorun.exe` application needs to be in a directory named 2577 (identifier for ARM processors) and will be copied automatically to the `\Windows` directory. Then the OS executes it with the parameter `install` when the storage medium is inserted and with the parameter `uninstall` when the storage medium is removed from the mobile device. After the uninstall execution the `autorun.exe` is deleted from the `\Windows` directory. Some examples for the use of this auto-run feature are given in [EVT3HELP], the problems and risks are discussed later in chapter 6.5.4 on page 83.

The file systems that come with Windows CE 3.0 do not offer any security, such as restricting rights to certain files. It might be possible to install a file system, that offers additional security functions, but as there is no operating system support to distinguish between different users, this needs to be all implemented in the file system.

The only protection offered by the mobile device by default is that files in ROM cannot be overwritten, apart from flashing the ROM image. But as “a DLL in the object store will be loaded before the like-named DLL in the same directory in ROM” [BOLING1999] a ROM file could be “overloaded” with a file in RAM allowing to effectively exchange the original ROM based file. The file properties of such an overloaded file reveal this fact, as the file system does correctly not set the “ROM” attribute. This allows easier updating of DLL files by loading the newer version into RAM, because building a complete new ROM image and flashing it into the ROM would require more time for both users and vendors.

Trying to reproduce this shows some differences in different applications:

Using the File Explorer on the Pocket PC 2002, the standard application to browse and manipulate files, I could not copy a file with the same name to the directory, trying to replace the original alarm sound file from the ROM (left screenshot in Figure 13 shows the Pocket PC 2002 File Explorer).

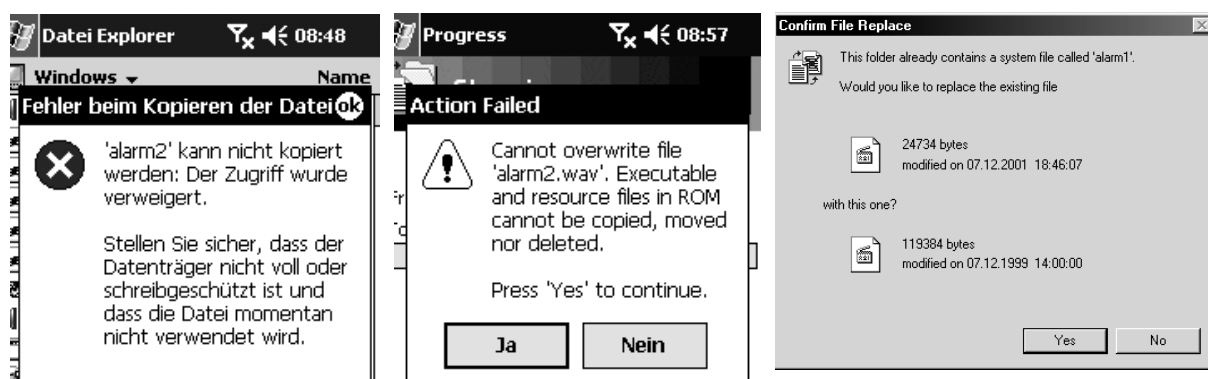


Figure 13: Trying to overload a ROM file in: File Explorer (l.), Resco Explorer (mid.), ActiveSync (r.)

More complex third party programs (like Resco Explorer 2003 [RESCOWEB]) also give an error message (see middle screenshot in Figure 13). However, it is very easily achieved using the desktop ActiveSync connection, which allows you to browse the device as if it would be a hard disk.

¹⁸ or “Storage Card 2”, “Storage Card 3” if more than one volume without a driver assigned name is mounted at the same time

For security this means, that also files stored in ROM are not protected and can be “modified” by malicious software. Trojan horses or malicious codes spreading by mail often exchange or modify the system DLLs. The `winsock.dll` is for example modified by a worm called W32/SK@M (see [NAIVILSKA] for more information). Such an attack would be possible by overloading the ROM file.

4.3.5.3 Windows CE Registry

The Windows CE registry is not different to other Win32 registries; it is a hierarchical system of keys and values. Values consume less memory than keys, so [EVT3HELP] advises to use as less keys as possible.

It is implemented as a RAM-based heap file. As it is RAM-based, the initial values of the registry are placed in the ROM, if no backup procedure exists¹⁹ or no backup exists, the initial values are loaded from ROM. The registry is always present in RAM, since it is also used for the system initialisation.

Windows CE 3.0 offers a protection mechanism that could prevent write access to certain registry hives, but it is not implemented in Pocket PC 2002 and therefore cannot be used in today’s mobile devices. I will look at it in more detail in chapter 4.6.6.1.

4.3.5.4 Windows CE Database

This is a special database system, which is optimised for small efficient storage, and only exists in the Windows CE environment. The Windows CE database provides storage, access and, most interesting, sorting and searching capabilities.

The databases are stored either as files in the object store or in a database volume. The latter enables the database to reside on any file system a Pocket PC device provides, so also in persistent storage, for example on a mounted volume of a flash-card.

Each database contains one or more records, which contain properties. For example the contact database contains a record for each contact, and each record has properties like first name, last name, street and so on. A record can only contain properties, not other records.

The whole object store itself is considered to be an always-mounted database volume²⁰ [MSDNWEB]. Like the access to the object store, the access to database objects is also transaction-based, allowing to rollback to a last known good state after system errors. To fully enable transaction-based write operations also on databases stored on files systems the file system driver needs to support recovery functions, as [INTELMEM] notes this is not supported on FAT file systems. The database is updated after each individual transaction, not only when the database opens or closes.

Each database volume can be up to 256 MB in size containing a maximum of 65,535 records.

As more than one application (or more than one thread) might access the same database, Windows CE allows the application to listen to notifications for her opened database. This notification is sent to the thread that requested database notifications during opening of a database, and it is sent whenever a record is changed, created or deleted.

Databases can store the data in a compressed format, the compression flag can be changed at any time, but only newly added records are affected by a change in compression. Using compression adds an additional processing overhead, which might additionally slow down the access to a database residing on a slower storage card [EVT3HELP].

4.3.6 Interprocess Communications

To allow processes to communicate with other processes on the same device or anywhere else, Windows CE also supports the component object model (COM), which is already used to build distributed applications in the desktop and server environment. The COM of Windows CE is a smaller version than the standard, with only a minimum number of interfaces, to preserve memory.

¹⁹ The OEM must implement this registry backup procedure. The registry can be loaded and written to another persistent storage device. [EVT3HELP]

²⁰ `CREATE_SYSTEMGUID` can be used to get the CEGUID for the object store database volume.

Nevertheless, Windows CE 3.0 in general also supports a nearly full-featured COM implementation, with only the operating system dependent function stripped off. This richer implementation is only available on “some OEM-provided platforms that are built on Windows CE 3.0” [MSDNWEB]. Pocket PC 2002 is no such platform [MSDNFAQ], therefore the richer implementation of COM, known as DCOM, which would have also allowed for some security functions, namely the Windows NT LAN Manager system security package (NTLMSSP), is not available in Pocket PC 2002 devices.

Apart from COM, Windows CE 3.0 applications can facilitate the Microsoft Message Queuing Service (MSMQ) to exchange messages. This allows applications to exchange messages even when not both of them are running.

4.3.6.1 COM

As laid out before, there is the minimal implementation and a richer implementation of the Microsoft component object model. Only the richer one does allow distributed COM (DCOM), therefore the Microsoft documentation uses the term “DCOM” to refer the extended, richer implementation, I will do the same. DCOM is not implemented in Pocket PC 2002 [MSDNFAQ], but the Pocket PC 2002 COM implementation supports one of the most often used COM controls, the ActiveX controls.

4.3.6.2 Message Queuing (MSMQ)

The message queuing, as the name suggests, makes it possible for distributed applications to exchange messages, even if a direct connection cannot be established, the message queuing service stores the message for later delivery, until the recipient becomes available online again. The receiving application reads the messages from their message queuing service, once the network is connected again or when the application is started again.

The receiving application can be another CE-based device or a Windows 2000 or NT machine. Message queuing can also be used to facilitate the communication of different applications running on the same device. Message queuing is especially important to mobile devices, as it enables applications to work independently from an existing network connection, which in a wireless world might not always be available or too expensive to be used as an always-on connection.

Some functions have been left out compared to the full MSMQ implementations found in the desktop operating systems to allow the Windows CE implementation to use up only 100-150 Kb of memory. From a security standpoint, the heaviest weighting limitations are that the Windows CE implementation of MSMQ does not support encryption [MSDNWEB], and it offers no access control [MSDISCONN].

To use MSMQ on the Pocket PC it must be installed from the MSMQ add-on pack, which can be downloaded from the web.

4.3.7 User Interface Services

The modules that are important for the user interface in Microsoft Win32 applications are bundled into one Windows CE module. The three Win32 modules: Application User Interface, User Interface (UI), and Graphics Device Interface (GDI) are combined into the Windows CE Graphics, Windowing, and Events Subsystems (GWES) module²¹.

GWES handles all the windows, dialogs and other resources that can be found in the Pocket PC 2002 user interface (UI), it also provides the basic messaging capabilities and the power management functions (see 4.5 for details on the power management).

Every client application interacts with the GWES [KRELL2002]; it is the “interface between the user, your application, and the OS” [MSDNWEB].

²¹ \Windows\Gwes.exe is the module's filename.

The GWES contains one message queue that receives messages from device drivers (such as mouse messages from the touch screen) and transforms them into another message queue, which is read by the application thread, by which the application can then react to that input [KRELL]. To react to these messages the application also needs to use functions from the GWES, if a window or dialog box is modified, the application uses functions from the user interface (UI) module, if graphical output is needed functions from the Graphics Device Interface (GDI) are used.

This chapter will continue to describe some details on the user's input messages; Pocket PC 2002 devices allow different forms of user input:

- Stylus input on the touch screen
- Input from an Input Panel (i.e. the virtual keyboard on the touch screen)
- Handwriting recognition

Stylus input is mapped to a subset of mouse events, and the other two forms of user input are converted by the input panel subsystem to keyboard events as they would have been created by an attached keyboard. The input panel subsystem is used by the GWES.

4.3.7.1 Stylus Input: Tap

A single touch with the tip of the stylus on the touch screen is called a "tap" and is translated to a click on the left mouse button. If the stylus were kept down, this would translate to a Left-Mouse-Button-Down message.

For example:

If the user touches the screen with the tip of the stylus and then moves the stylus, while still touching the screen and then stops touching the screen, this would normally be converted to first a Left-Mouse-Button-Down message and then a couple of Mouse-Move messages with finally a Left-Mouse-Button-Up message.

Under special circumstances a left mouse button double click message can be signalled, by two consecutive taps.

4.3.7.2 Stylus Input: Tap-and-Hold

However, with the stylus no right mouse button can be emulated. To overcome the lack of a right mouse button with stylus input Microsoft introduced a mouse gesture called "Tap-and-Hold".

This means that the user touches the touch screen with the tip of the stylus (for example over a file object in the File Explorer) and continues to hold the stylus on that position still touching the screen. If the application is able to recognize this gesture it shall give a visible confirmation by showing a clockwise appearing ring of red dots (see Figure 14). The user has to hold the stylus down until the ring is completed to perform a tap-and-hold.

If the user lifts the stylus before the ring is completed, it will be interpreted as a single "tap".

The "tap-and-hold" gesture is mapped to a right mouse click, and mostly brings up pop-up context menus, as the right mouse click in normal desktop applications would.

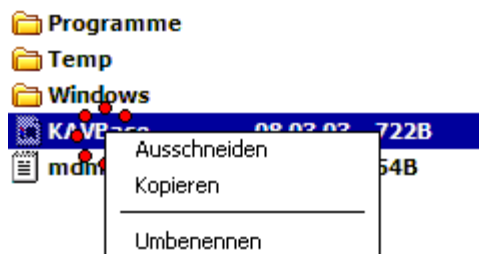


Figure 14: Tap-and-Hold gesture brings up a pop-up menu

Some additional programming is needed to support tap-and-hold gestures in the application, and to signal it to the user by showing the red dots [MSDNTAPNHOLD].

4.3.7.3 Input Messages in General

Messages of keystrokes or stylus actions are passed to the window that has the focus, which is similar to the behaviour in Windows 32 systems. This should make it generally possible to log user input as it is done with key-loggers on desktop computers used by Trojan horses, allow spying on the user.

4.3.8 Multimedia

I will, for completeness only, very briefly describe the multimedia functions of Windows CE 3.0.

4.3.8.1 Graphic

The Windows CE 3.0 graphics device interface (GDI) is used to control the graphics and text displayed on a Windows CE 3.0 device. It is designed with the restrictions of mobile devices in mind, and supports three drawing objects: pens, brushes and fonts. This allows an application to draw lines, curves, closed figures, text and bitmapped images using GDI functions. Additionally to writing it to the display, the GDI also allows to write to memory regions or to printers.

Additionally to regular graphics access through the GDI applications, especially for faster graphics like in games, Pocket PC 2002 provides the Pocket PC 2002 Game API (GAPI). [EVT3HELP] describes the GAPI also as a way “to use the device in a way that is not supported by the operating system”.

Apart from gaining control over the screen, GAPI also allows to query for different hardware buttons and manage key presses from the hardware keys of the Pocket PC 2002 device.

4.3.8.2 Sound

Windows CE 3.0 applications can play waveform audio files to enhance the application with sound. They can play sounds from three sources: audio files, system sounds (standard or user set), or waveform stored as resource within the application.

An API exists allowing to directly access the waveform audio input / output (I/O) device (called WaveAPI). This way an application can specify audio details like the volume or the playback rate.

MIDI is considered a special WAV format, an extra MIDI API has been added to Pocket PC 2002 allowing to play MIDI sound or generate DTMF tones.

To record and playback wave even more easily applications can use the Voice Recorder API, which is specific to Pocket PC 2002 devices.

4.4 Pocket PC 2002 network functions

Windows CE 3.0 supports a multitude of options to transmit and receive data. Different Pocket PC 2002 devices even contain special hardware extensions, for example Bluetooth, to facilitate data exchange on different mediums. The different network functions can be grouped by their support for client or server functionality, which I will indicate in brackets at each more detailed description. Some of the components can be used as both.

To provide a quick overview of the network components of Windows CE, I have combined the information from two partially contradicting graphics from Microsoft’s documentation ([MSCEOSI] and [EVT3HELP]) into Figure 15.

4.4.1 Overview of Network Components

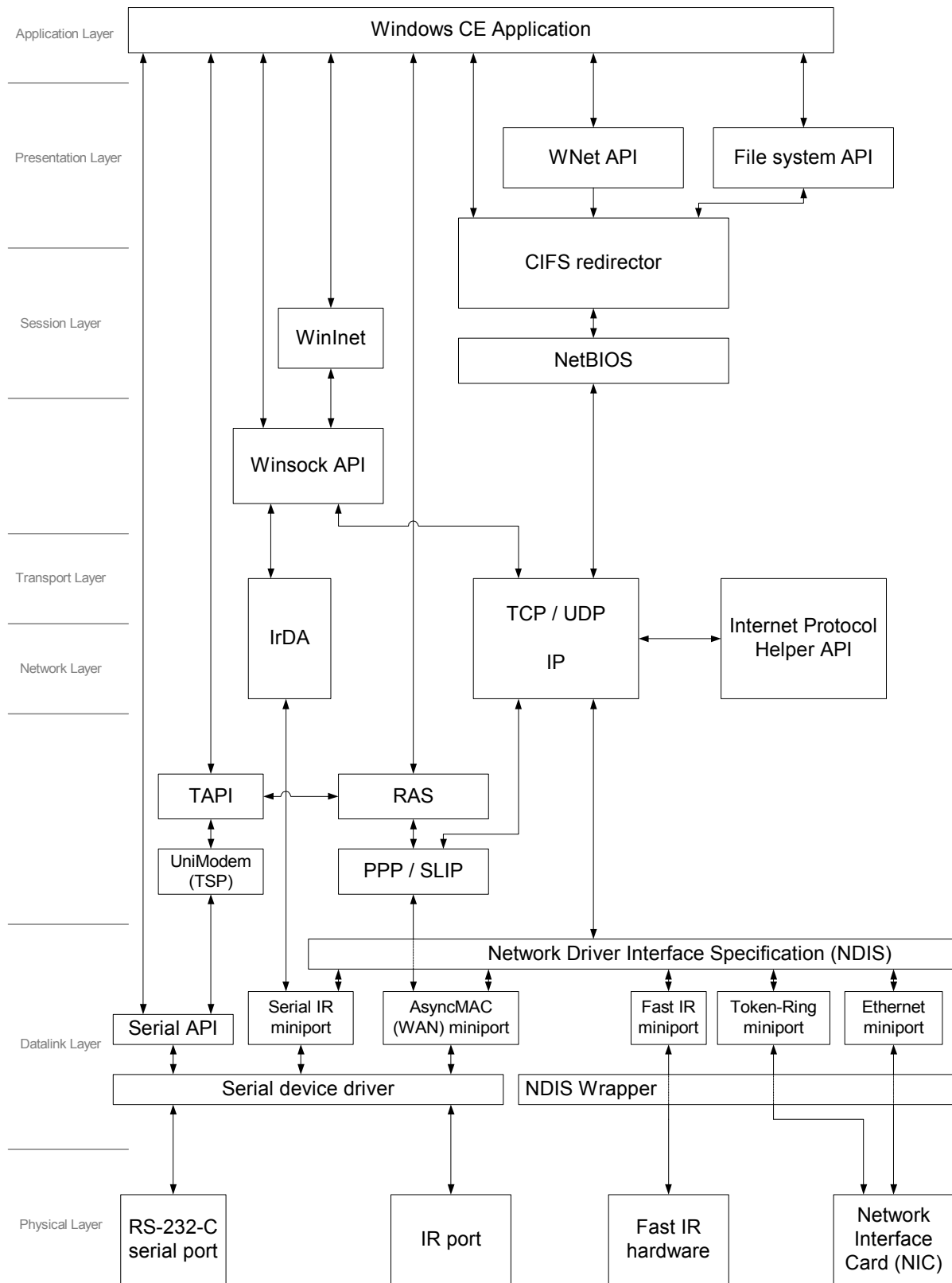


Figure 15: Overview of different Windows CE networking components and their interactions

The indicated OSI layering shall not be given too much emphasis, as the Windows networking was not built along this reference model. The OSI layers can only be used as a rough guide.

4.4.2 Serial Connections [client/server]

Applications can use the Serial API to send and receive data through standard serial connections. An application opens, reads and writes to a serial port as if it is a file, using the file I/O operations, additional functions are used to fully facilitate the serial connection of an RS-232-C connection.

All the standard MS Windows desktop functions of serial communication are supported in MS Windows CE 3.0 [MSDNWEB].

Serial connections cannot only be established over a serial cable, but also the infrared port can be used. Windows CE allows three different ways to access the IR port:

- Raw infrared (raw IR),
- IrCOMM and
- Infrared Sockets (IrSocks).

With raw IR, the application has the most freedom in access to the IR hardware, but also the most duties. It is opened as if it is a serial port with an infrared hardware attached. This access is non-IrDA compatible, and the application has to take care of all the eventualities that can occur (like signal interruption or signal collision) during a transmission.

IrCOMM is a protocol specified by the Infrared Data Association (IrDA), which allows emulating a serial or parallel connection over an infrared port, in Pocket PC only serial communications are used (see [IRCOMM] for details).

The serial IR connection can be facilitated to communicate with all kinds of devices reachable via infrared, for example modems in GSM mobile phones. An application that facilitates IrCOMM is relieved of the hassle to detect and handle interruptions and other occurrences of the infrared wireless transmission. Apart from a difference in transmission time, an application can use IrCOMM connections with the same ease as serial cable connections.

Infrared Sockets are additional transport service providers for WinSock [JOHANNSEN1998], allowing using WinSock connections over IrDA protocols. Used over an IrDA connection some WinSock functions work differently from the general WinSock functionality as described in more detail in chapter 4.4.5.

4.4.3 Telephony API (TAPI) [client]

The TAPI is situated above the OSI Data Link layer; it provides an application with easy access to modem functions. So-called telephone service providers (TSPs) are needed to communicate. Windows CE 3.0 comes with a standard modem driver (called Unimodem); this allows the communication with an AT-command-based modem. TSPs can also be supplied by third companies to integrate other hardware or services into the TAPI.

Except some minor differences, the TAPI in Windows CE 3.0 mirrors the functionality found in the MS Windows NT 4 TAPI Version 2.0.

To sum it up, the Windows CE TAPI allows managing outgoing and incoming telephone connections of any kind (voice or data).

4.4.4 Remote Access Service (RAS) [client]

The remote access service, also known as RAS, makes it possible for an application to connect and communicate with a remote host computer. The remote access service is located above the data link layer, and it uses the point-to-point protocol (see 4.4.10) to send the data to another computer.

RAS connections are either dial-up connections, where the RAS uses TAPI, or direct serial and infrared connections.

Which number to call, and other settings, such as authentication data, for each remote connection are managed in so-called phone book entries. The phone book entries are stored in the registry²², and they store a subset of the information available in desktop phone book entries. The functions for managing phone book entries are also contained in the remote access service.

4.4.5 Windows CE Sockets (WinSocks) [client/server]

“Sockets is a general-purpose networking API” [EVT3HELP]. The MS Windows implementation of Sockets is called Windows Sockets or WinSock, Windows CE 3.0 supports WinSock version 1.1.

²² HKEY_CURRENT_USER\Comm\RasBook

Many applications facilitate the Windows Sockets located just above the OSI transport layer. All applications that want to access the TCP/IP or IrDA protocols in Windows CE 3.0 must do so through WinSock. For example: *Pocket Internet Explorer uses the Windows Internet API (WININET), which builds on WinSocks.*

Being located above the OSI transport layer, WinSock can make connections either using the TCP/IP protocols or using the Infrared Data Association (IrDA) protocols, which are referred to as IrSocks.

4.4.5.1 Secure Sockets

From Windows CE 3.0 WinSock two security protocols are offered to the layers above. The first protocol supported is the Private Communication Technology (PCT) protocol 1.0 and the second one is Secure Socket Layer (SSL) in version 2.0 and 3.0. After an application has established a secure socket, the security of the communication is transparent to the application.

[MSDNCOM] compares SSL, TLS and PCT, and states that PCT “should not be used for new development”. Therefore, this work will not go into any details on PCT, which is only kept for compatibility reasons.

To stay with the list of security protocols from the above-mentioned comparison, TLS in its latest version 1.1 [RFC2246] could be considered SSL version 3.2.

Windows CE 3.0 does not support TLS at all; support for TLS 1.0 is implemented in Windows CE .net.

The second protocol supported by Windows CE 3.0, SSL [SSL3], is built on certificates allowing authentication of the communication partner. These certificates are issued by certification authorities (CAs), which themselves have certificates. Some certificates of the trusted CAs, which are on top of the hierarchy, must be accessible in order to verify a chain of the hierarchically layered certificates (certificate chain). These certificates are called root certificates.

MS Windows CE stores a database of trusted root certificates from different big CAs (for example VeriSign or Thawte) in the registry²³ (more about certificate management in chapter 4.6.4). During the establishment of a secure socket connection the certificate of the communication partner is received, the root certificate is extracted from the validated certificate chain and then compared against the database of trusted root CAs. Only if this check returns the value `SSL_ERR_OKAY` to identify the SSL certificate as verified, the secure socket connection can be established through SSL.

4.4.6 Windows Networking (WNet) [client]

This API allows a CE-based application to use a windows network. WNet facilitates the Common Internet File System (CIFS) redirector to access resources over the network.

WNet is also known as Server Message Block (SMB) Protocol and the functionality is located in the `netbios.dll`, but only supports the NetBIOS functionality provided by the CIFS redirector located in the DLL `redirector.dll` [MSDNWEB].

The functionality of the WNet API is basically the same compared with the desktop environment, but there are some small changes. The differences are:

- Windows CE does not support drive letters, but network mappings can be made between a remote Universal Naming Convention (UNC) name and a local name instead of the drive letter.
- The connections are not restored when the CE device is rebooted and the concept of a network context (i.e. Workgroup) is not known.
- The only network provider supported by Windows CE is Microsoft Windows Network. Only connections to such networks can be established.
- Also some error messages are different to the desktop implementation, but the API and the network access shall be identical.

Most standard Windows CE file functions can deal with UNC names [MSDNWEB], so that directly accessing WNet API functions is not needed in most cases.

Windows CE can connect to Windows based desktop platforms, or servers compliant with the Windows NT LM 0.12 dialect of the Common Internet File System (CIFS) specification via the redirector.

²³ `HKEY_LOCAL_MACHINE\Comm\SecurityProviders\SCHANNEL\CAs` contains the trusted root certificates

To access a remote file, the Windows CE device needs a name that is unique on the network, which it uses to register on the remote server.

If no unique or valid device ID is found, or the default device ID ("Pocket_PC") is not changed, the Pocket PC device reports an error when connecting to the network and requires the user to enter a valid unique device ID. Not only the actual ID²⁴ is stored in the registry, but also the default device ID²⁵.

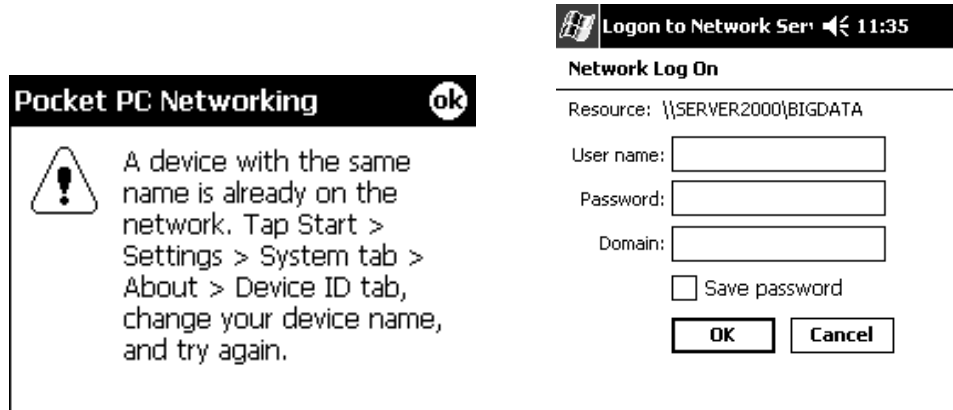


Figure 16: No valid device ID found (left), Network Logon dialog (right)

If required, the user is then prompted to enter the account details (username, password and domain) to gain access to the remote resource (see Figure 16). This account information, including the password in encrypted state, can be saved to the registry²⁶ [MSDNWEB], to avoid any security problems from the start, storing should not be considered.

Interesting for daily work is that even though the files and folders on remote network locations can be explored with the Pocket PC File Explorer application, the files cannot be opened directly with the associated Pocket PC application directly from the Pocket PC File Explorer.

This limitation is not only true for Word and Excel documents [MCPHERSON2002] but also for other file types, like plain text (*.txt) or HTML (*.html) files. To open the document from the Pocket PC the file either needs to be copied to the Windows CE device or a shortcut needs to be created to the remote location (as also suggested by [MCPHERSON200]), then the associated Pocket PC application opens the file with a single tap.

4.4.7 Windows CE Internet API (WinInet) [client]

The Windows CE 3.0 Internet API can be used to easier develop Internet client applications; the Pocket PC Internet Explorer for example makes use of WinInet. The Windows CE Internet API does not, in contrast to the desktop version, support Microsoft ActiveX controls and the Gopher protocol.

WinInet builds on the WinSock and allows using handles, known from working with files, to be established for Internet connections. The handles are called `HINTERNET`, and they are created by calling the function `InternetOpen`. With a `HINTERNET` handle you can then use commands like `InternetOpenURL` to open an URL and if the URL is a file it can be accessed through `InternetReadFile` as if it were a local file, which would be accessed through the `ReadFile` command.

If an application wants to use FTP or HTTP functions, it must first open a connection with the function `InternetConnect`. This function opens either a FTP or HTTP connection to a specified site.

4.4.7.1 FTP Functions

A set of FTP functions is available to enable the client functions necessary to access, rename, and delete files and folders on remote FTP servers. For a FTP connection the authentication data for the FTP sessions (username and password) is specified when calling the `InternetConnect` function.

²⁴ The registry key `HKEY_LOCAL_MACHINE\Ident\Name` holds the unique device name.

²⁵ The registry key `HKEY_LOCAL_MACHINE\Ident\OrigName` holds the default device name.

²⁶ `HKEY_CURRENT_USER\Comm\Ident\Username` (plain text) and `HKEY_CURRENT_USER\Comm\Ident\LMPW` (hashed) hold the values, if the user selects save.

4.4.7.2 HTTP Functions

For a HTTP connection the application can also use a set of functions to send HTTP requests to a server.

For initiating authenticated HTTP connections the `HttpOpenRequest` function, which opens a handle to a HTTP request, is directed at the default HTTPS port and the flag `INTERNET_FLAG_SECURE` must be set. WinInet will then use a secure Windows Socket connection (PCT or SSL) to connect to the server (see 4.4.5.1). Additionally two other flags specify if security errors can be ignored. To establish the session even if the host name does not match the certified name the flag `INTERNET_IGNORE_CERT_CN_INVALID` can be set, even an outdated certificate can be ignored by setting the `INTERNET_IGNORE_CERT_DATE_INVALID` flag. From a security point of view, I see no reason for including the last flag.

Other functions, described in more detail in [EVT3HELP] can then be used to send HTTP requests.

4.4.8 TCP/IP stack in Windows CE 3.0 [client/server]

The TCP/IP stack implements the TCP and the IP protocol on Windows CE 3.0. This work will not go into the details, as these are the standard Internet protocols.

Among the protocols supported are TCP, UDP, IP, ICMP, and DHCP. The configuration information for the TCP/IP stack is stored in the registry. More details can be found in [EVT3HELP].

The Windows CE 3.0 TCP/IP stack's robustness was tested in [MEUNIER2002] and was found to lack robustness.

Some functions to manage network adapters are located in an additional API called Internet Protocol Helper.

4.4.9 Network Driver Interface Specification (NDIS) [client/server]

Windows CE 3.0 implements the version 4.0 of the Network Driver Interface Specification (NDIS). I will only shortly introduce the driver concept of NDIS. NDIS provides two defined interfaces, one to the protocol drivers and the other to the network hardware. The NDIS implementation in Windows CE supports the following communication media: Ethernet (IEEE 802.3), Token Ring (IEEE 802.5), IrDA and WAN. Network hardware that is supported out of the box with Windows CE is the NE2000 compatible Ethernet network interface card (NIC). A so-called miniport driver exists for NE2000 NICs, this miniport driver is encapsulated by the NDIS layers as shown Figure 17, it communicates through NDIS with both sides, the NE2000 hardware and the protocol driver TCP/IP.

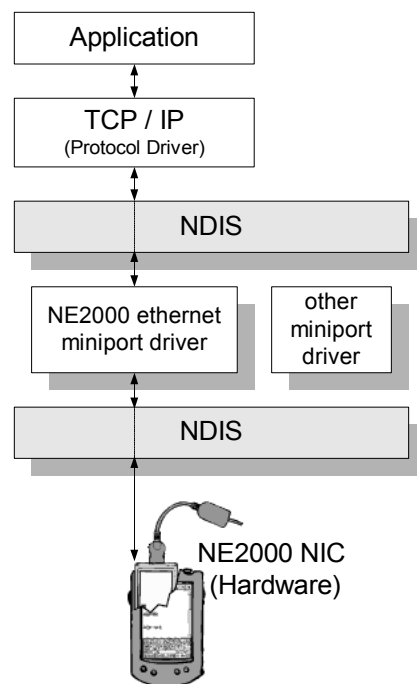


Figure 17: NDIS: NE2000 miniport driver

The support of the NDIS concept, which was first implemented in Windows NT, makes NDIS miniport drivers from Windows desktop systems compatible with Windows CE, so they are portable. They are not however directly exchangeable, as they need to be compiled as DLLs for Windows CE.

Through NDIS, the drivers are also informed about inserted or removed network adapters; so that the drivers can be Plug and Play aware.

For example: If a PC CARD network adaptor is inserted, NDIS will notify the miniport driver and a running protocol driver will be notified. Then the driver can bind to that new network adapter hardware.

It shall be noted that if a PC CARD network adapter is inserted the mobile device will no longer be automatically suspended to save battery power. If the power save function shall suspend an idle mobile device even if a PC CARD is inserted a registry key²⁷ needs to be set accordingly. More about the power save function in chapter 4.5.

4.4.10 Point-to-Point Protocol (PPP) [client/server]

The remote access service (RAS) is located above the data link layer, and it uses the point-to-point protocol (PPP) [RFC1661] to send the data to another computer. PPP encapsulates higher-level protocol packets and sends them over to the other computer, for example using a modem a Windows CE device can use IP over PPP to access an Internet host computer.

RAS connections are either dial-up connections, where the RAS uses TAPI, or direct connections through serial or infrared.

Which number to call, and other setting for each remote connection are managed in so-called phone book entries. The phone book entries are stored in the registry²⁸, and they store a subset of the information available in desktop phone book entries. The functions for managing phone book entries are also contained in the remote access service.

As said earlier the connection is made using the point-to-point protocol (PPP), a PPP dial-up connection is established in three steps:

1. PPP link establishment
2. PPP User authentication
3. Invoking network layer protocol

During step one the two communication parties establish the physical connection and negotiate different connection parameters such as compression and encryption. They negotiate also the authentication protocol used in step two.

Step 2 is interesting from a security standpoint, Windows CE offers three authentication methods for PPP User authentication:

- Password Authentication Protocol (PAP):
clear text exchange of username and password, not secure
- Challenge-Handshake Authentication Protocol (CHAP):
username in clear, password used as key for the MD5 hashing of the challenge and other information
- Microsoft Challenge-Handshake Authentication Protocol (MS-CHAP):
username in clear, password is hashed and then used as key for the hashing of the challenge and other information, version 1 and version 2

The three methods are also available and used by Windows desktop systems. Whilst the first method (PAP) is very insecure, the Challenge-Handshake Protocol is relatively secure. The main difference between CHAP and Microsoft's extension is that MS-CHAP allows storing only hashed password on the server side. Version 2 of MS-CHAP introduced several changes, among them mutual authentication, details on the different versions of MS-CHAP can be found in [MSCHAPV2].

In step three the network layer protocols are invoked; among these is the Microsoft Point-to-Point compression (MPPC) protocol. MPPC does not only allow setting the encryption, but also negotiating the encryption of the data exchanged using the Microsoft Point-to-Point data encryption (MPPE) protocol.

The encryption is done using RC4 with a key with 40bit or 128bit length, more details on MPPE can be found in [RFC3078]. Encryption can only be used if MS-CHAP was used for the authentication in step 2 (see [MSVNPPTP]).

²⁷ HKEY_LOCAL_MACHINE\Comm\Cxport\NoIdleTimeReset=0

²⁸ HKEY_CURRENT_USER\Comm\RasBook

4.4.11 Virtual Private Network (VPN) [client]

Virtual Private Networks (VPNs) allow establishing secure connections between two VPN endpoints even over insecure networks. To give an example, the mobile device, as one VPN endpoint, encrypts and secures the data before sending it over an insecure public wireless LAN to the company server, as the other VPN endpoint.

Pocket PC 2002 offers only the client functionality and only allows using the Point-to-Point-Tunnelling Protocol (PPTP) [RFC2637] for the establishment of a VPN connection.

The Point-to-Point-Tunnelling protocol is used to encapsulate packets, the authentication is done using PPP and the PPP encryption is used. More details on how PPP and PPTP work can be found in [MSVPNPTP].

Pocket PC 2002 does not allow the use of more secure VPN protocols such as L2TP tunnels secured by IPsec, also the Enhanced Authentication Protocol (EAP) is not yet supported. Microsoft sees that PPTP has security limitations and advises users that need stronger authentication and encryption to use third party products [MSVPNFAQ]. Additionally as PPTP only uses username and passwords as authentication Microsoft advises to use strong passwords like "f*3L~qO2>xR3w#4o" [MSVPNFAQ], as one of the attacks is simple password guessing.

4.4.12 Terminal Services [client]

With the Pocket PC 2002 Terminal Services client the mobile device can connect to a remote desktop. So the user can remotely do work as if they sit in front of their Windows desktop computer. The functionality was first introduced to Windows NT where a special server extension is required and with Windows XP this functionality is known as Remote Desktop and can be installed with the operating system. With the Pocket PC 2002 terminal service client, that understands both the Remote Desktop Protocol version 4.0 (Windows NT 4.0 Terminal Server Edition) and version 5.0 (Windows 2000) the user can remotely connect to and work on machines running a terminal service server. Therefore, from the security point of view using the Pocket PC terminal service client has the same impact as using the desktop based Windows Terminal Clients, because it is the same protocol.

4.4.13 ActiveSync

ActiveSync is the desktop application that will be installed on the home system (as defined in 2.3.4) to establish a connection between the desktop and the mobile device.

ActiveSync version 3.7 is the latest, but version 3.6 was current at the time conducting the first tests, as version 3.7 brings no major changes. Only version 3.6 was used throughout this work. ActiveSync allows the exchange of data or applications with a cradled mobile device, and can act as an Internet proxy allowing the cradled mobile device to use the Internet for e-mail or browsing.

4.4.13.1 ActiveSync Pairing Process

To initially establish a connection between a mobile device and a desktop system the following steps are necessary:

1. Install and run the ActiveSync software on the desktop system
2. Connect the mobile device with the desktop system, normally done through a cable and a docking-station
3. Create a partnership (either a Standard- or Guest-Partnership) from the desktop computer

Once installed on the desktop system ActiveSync automatically starts each time Windows is started (step one) and by default, ActiveSync will monitor if a mobile device is connected. Once a mobile device is connected (step two), ActiveSync will begin the synchronisation process. If it is an unknown device, it will present a dialog asking whether to establish a guest- or a standard-partnership (see Figure 18).

If the Guest-Partnership is selected ActiveSync will forget that a connection has been established between the desktop system and the mobile device, after it has disconnected. Only if the Standard-Partnership is chosen, the mobile device and the desktop system will save that Partnership and this is the pairing process (as defined in 2.3.4) resulting in the desktop system becoming the home system of the mobile device.

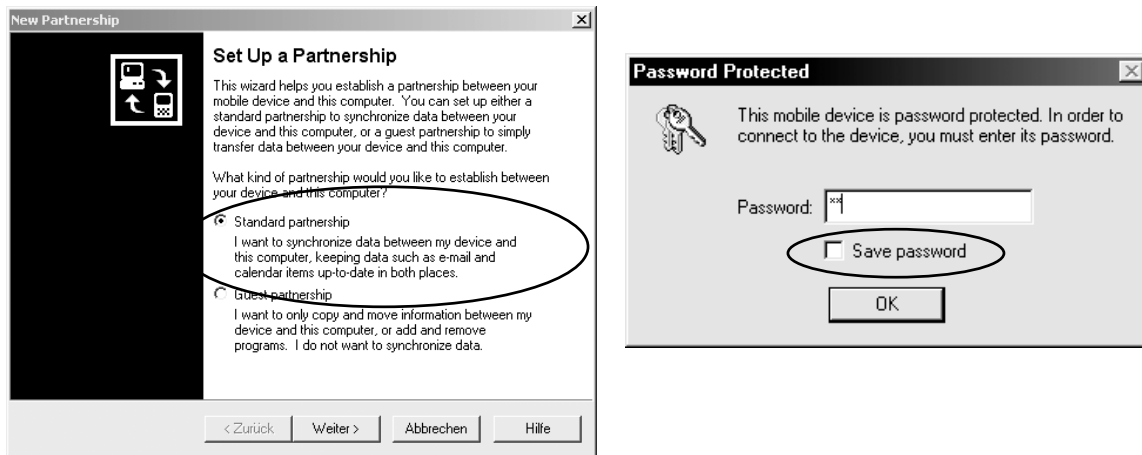


Figure 18: ActiveSync: Partnership-Dialog, Password-Dialog on the desktop computer

A Pocket PC 2002 device can store two entries for standard partnerships in two registry paths HKLM\Software\Microsoft\Windows CE Services\Partners\P1 and P2, if the user tries to pair with more than two systems he is asked to delete one of the two existing partnerships.

In general an ActiveSync connection to a password protected Pocket PC can only be established, if the password is entered correctly in the ActiveSync dialog box on the home system (see Figure 18), the mobile device can stay locked during ActiveSync. More details on the power-on protection can be found in chapter 4.6.1.

Saving the password on the desktop side is never recommended, even though accessing the saved password in ActiveSync version 3.6 is no longer as easy, because it is no longer just saved in the registry (like in ActiveSync 3.0 see [OCHOA1999]). Looking at the registry keys touched during an ActiveSync process (using a registry monitoring tool) it can be observed that ActiveSync 3.6 actually requests to delete the registry key²⁹ that was used to save the password in the earlier version 3.0 as described in [OCHOA1999].

While the power-on protection on the mobile device makes password guessing hard by delaying the time before passwords are checked with each wrong attempt, the desktop side of the password protection does not. It allows the user to enter three wrong passwords, before it disconnects the mobile device and requires the user to connect again. In [MEUNIER2002] a brute-force attack using this password entry dialog has been demonstrated, that does not require physically removing the mobile device.

Once installed on the desktop system ActiveSync automatically starts with Windows. By default, ActiveSync will monitor if a mobile device is connected. If a mobile device is detected, the synchronization process is started. With this default configuration an ActiveSync synchronization will also take place if the desktop system is locked, as described on Bugtraq [SAMPLES2001].

Microsoft points out in a reply to [SAMPLES2001] that an ActiveSync synchronization with a locked desktop is only possible, if a standard partnership exists and no password needs to be provided on the desktop (dialog from Figure 18), either because the device is not locked, or the password was saved. In addition, the ActiveSync application must be configured to continuously synchronize to synchronize with a mobile device even on a locked desktop system. If the ActiveSync application would be configured like shown in Figure 19 the mobile device would not be able to synchronize with a locked desktop, even without power-on protection (see the original reply [UY2001]).

²⁹ In ActiveSync 3.0: HKCU\Software\Microsoft\Windows CE Services\Partners\XXXX>Password, where XXXX stands for the partnership ID

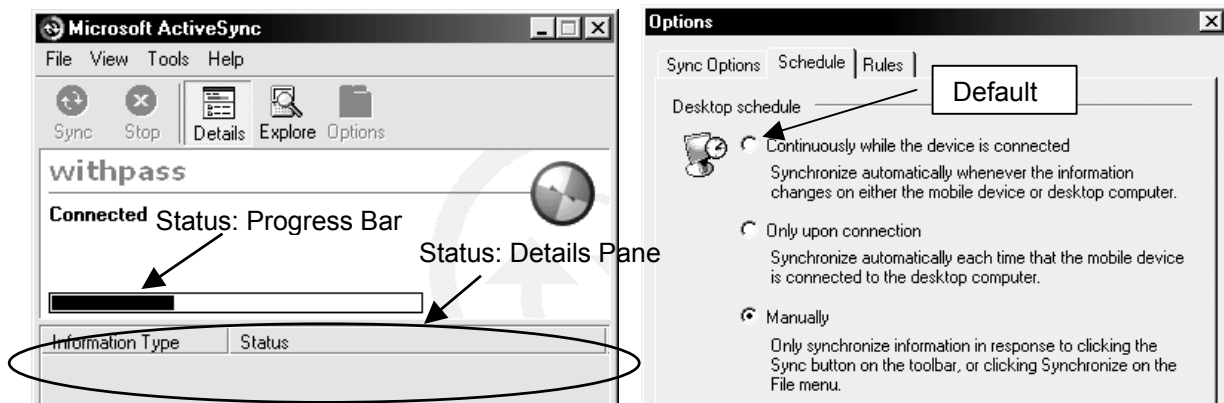


Figure 19: ActiveSync Application: main screen (left), options dialog (right)

The cradled state can also be achieved over a network connection; so wireless ActiveSync connections are possible [HPFAQWSYNC]. Only with an existing partnership, ActiveSync is possible over network connections, so a cable or infrared connection is initially required.

In a document, even though it is about Windows CE .net, Microsoft discourages users from using ActiveSync connection over networks, as “Network-based synchronization, especially in wireless networks, poses security risks, such as sniffing” [MSACTVIESYNC].

The messages exchanged during the ActiveSync are looked at in more detail next.

4.4.13.2 ActiveSync Details

The ActiveSync messages exchanged are not documented, nor are the exact details of the ActiveSync protocol.

This work will not try reverse engineering the protocol, but try to develop a basic understanding.

The ActiveSync application consists of two modules `wcescomm.exe` and `wcesmgr.exe`. The program `wcescomm.exe` is started using the RUN entry in the registry every time Windows is started and it displays a little icon in the taskbar. If a connection is detected and the device is power-on protected it will prompt for the password and if correct will then launch the second program `wcesmgr.exe`, which displays the main screen as shown in Figure 19. The two programs communicate using TCP messages, therefore `wcescomm.exe` uses ports 5679 and 999 and `wcesmgr.exe` uses 5678 and 990.

Especially the port 5679 is opened all the time whether a mobile device is connected or not. For ActiveSync version 3.5, an older version³⁰, this port could be used for denial of service attacks against ActiveSync as described in [DAVIS2003]. These ports are not only used for the communication on the same device (localhost), but also need to be opened if the synchronization takes place over the network, so firewalls need to allow traffic on these ports in order to sync over a network [MSQ259369].

To actually monitor what is going on when a local ActiveSync takes place is difficult as the Windows OS handles localhost loopback connections different than Ethernet connections, meaning that usual sniffing tools (for example ethereal or commview) cannot capture packets. To at least see some of the packets I managed to split the two parts of the ActiveSync application on two different machines and forward the missing part's TCP-ports to the other machine using a tool called TCP Tunnel [TCPTUNNEL]. The following Figure 20 illustrates how the TCP ports are forwarded.

³⁰ ActiveSync 3.6 was used for all tests carried out in this work, the latest version 3.7 was just released in the middle of 2003.

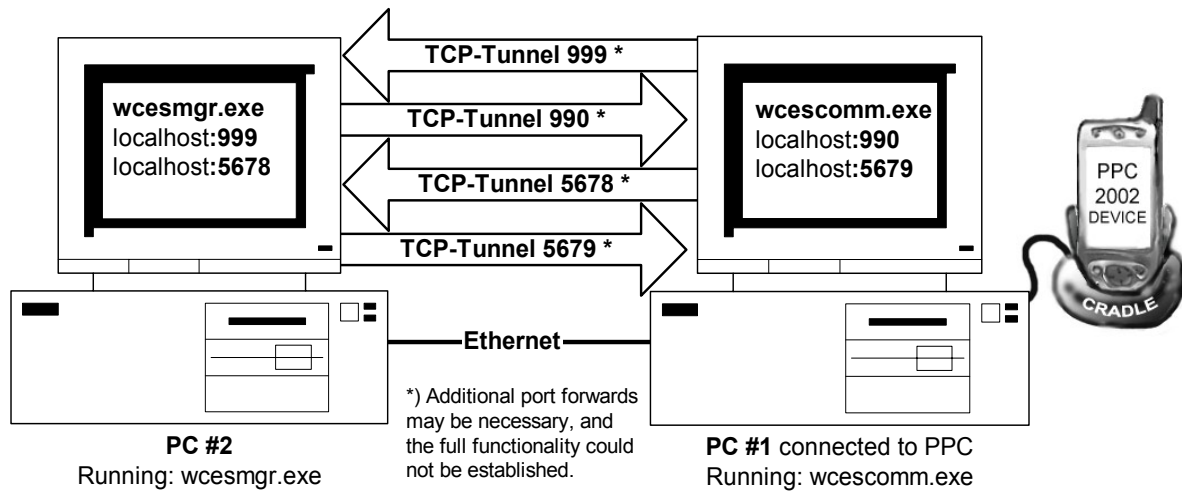


Figure 20: Illustrating the splitting of the ActiveSync application over two PCs

Also with additional port forwards the ActiveSync connection could not be established completely, but the setting allowed to sniff among others the messages exchanged when the password is entered on the desktop machine. As far as my analysis has shown the password is not encrypted when exchanged between `wcesmgr.exe` and `wcescomm.exe`. Further details on the sniffed password and details of the packets can be found in appendix D.

Other people have also analysed parts of the ActiveSync protocol, especially in [MEUNIER2002] there are interesting findings on some other inner workings of ActiveSync. However [MEUNIER2002] does not state which exact version of ActiveSync they have analysed, and it might be an earlier version than 3.6. Also some details on ActiveSync, although not under aspects of security, can be found at [LANGE2001] and [NORDENHOLZ2002].

This work will not further analyse the ActiveSync protocol, but continue stating the possible use of ActiveSync connections.

4.4.13.3 Use of ActiveSync

The ActiveSync application allows to act like a proxy, to provide internet connectivity to the mobile device, if the home system also has internet connectivity. The ActiveSync application needs to be configured to do so and allows to either act as a proxy for the “Internet Connection” or the “Work Connection” as defined in the connection manager (see next chapter 4.4.14 for details on connection manager). ActiveSync calls this option “Pass Through” and the dialog (shown in) allows to select the connection for which the proxy service will work. On the mobile device’s side the same connection must be chosen to use the proxy.

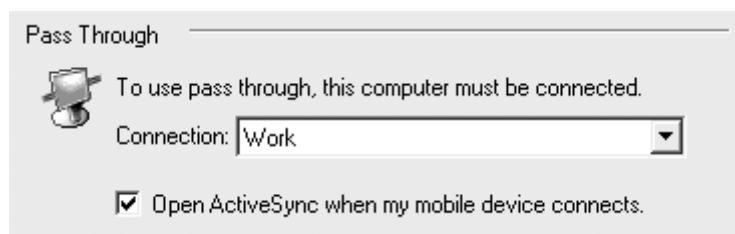


Figure 21: ActiveSync: With Pass Through the home system acts as a proxy server

There is no real way to disable this feature, other than to choose “Work Connection” on the desktop and “Internet Connection” on the mobile device’s side and vice versa.

In ActiveSync so called Synchronization Service Providers (SSPs) running on both sides allow replicating data between applications on the mobile device and the desktop. ActiveSync version 3.6 already comes equipped with SSPs for:

- Outlook (E-mail, Calendar, Contacts, Tasks, Notes)
- Internet Explorer Favourites (including Offline contents)
- Pocket Access
- My Documents folder (File synchronization)

All SSPs consist of two modules, one module resides on the desktop side, one on the mobile device side. All SPPs shall report their status details to the Details Pane as depicted in Figure 19 on the left. The SSPs are invoked once a connection is detected.

But not only the ActiveSync Synchronization Service Providers are notified that a connection is established, all applications listed in two registry keys³¹ are launched whenever a connection is established or ended, additionally applications can register via the COM interface to receive notifications. A connection in the above sense is also established if no partnership exists, but for power-on protected devices the correct password must be supplied.

An example for the use of registry based notification:

The pattern update process of Kaspersky Antivirus (see chapter 7.3.1.6). The update program running on the desktop is started whenever a connection to a mobile device is made. It will check the Internet for the actual pattern and transfer the actual pattern to the mobile device.

Also interesting to observe here, is that port 990 of `wcescomm.exe` is used to transfer the file data of the updated pattern, once the connection is established. This is the port used to remotely talk to the mobile device using the Remote API (RAPI).

4.4.13.4 Remote API (RAPI)

The Remote API Allows to execute functions on the mobile device from the desktop (remotely).

The RAPI functions that can be executed remotely from the desktop side are mirroring standard Windows CE functions and come from the following areas:

- System information queries
- Database manipulation
- File and Directory manipulation
- Registry manipulation
- Shell and Window management

Additionally the Remote API can be used to create or call processes on the mobile device.

This all is already possible without establishing a partnership (Guest- or Standard-Partnership), but for power-on protected mobile devices the correct password needs to be supplied on the desktop side. This allows desktop applications to gain nearly full control of the mobile device, for example they can transfer files, which includes overloading ROM files or modify any registry entry.

Further analysis has shown that during RAPI calls the password is checked by `wcescomm.exe` when the underlying transport connection is established, as the following can be observed:

1. An ActiveSync connection to a power-on protected mobile device is established and the password is correctly entered on the desktop side
2. A RAPI call is successfully executed
3. The power-on password is changed on the mobile device
4. Another RAPI call is executed, but with no effect

This again shows that `wcescomm.exe` does establish the full connection, including providing the password for power-on protected devices. So for RAPI to work the underlying ActiveSync connection must be established by `wcescomm.exe`.

But once an ActiveSync connection is established to a Pocket PC 2002 mobile device, the RAPI calls are allowed to process without any restrictions, only in the next version of Pocket PC based on Windows CE .net version 4 it is possible to restrict what RAPI calls can do on the mobile device [MSRAPI].

³¹ HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows CE Services\AutoStartOnConnect & AutoStartOnDisconnect list the names and the command lines to the applications automatically executed each time a mobile device is connected or disconnected.

4.4.13.5 File Conversion Filter

Even though the Pocket PC applications can mostly handle the standard document file formats, as described in further detail in chapter 5, they mostly have their own file formats. When files are exchanged through ActiveSync synchronisation operations, so-called filters, can automatically convert files. The file filters can be configured individually through the options dialog of ActiveSync (see Figure 22).

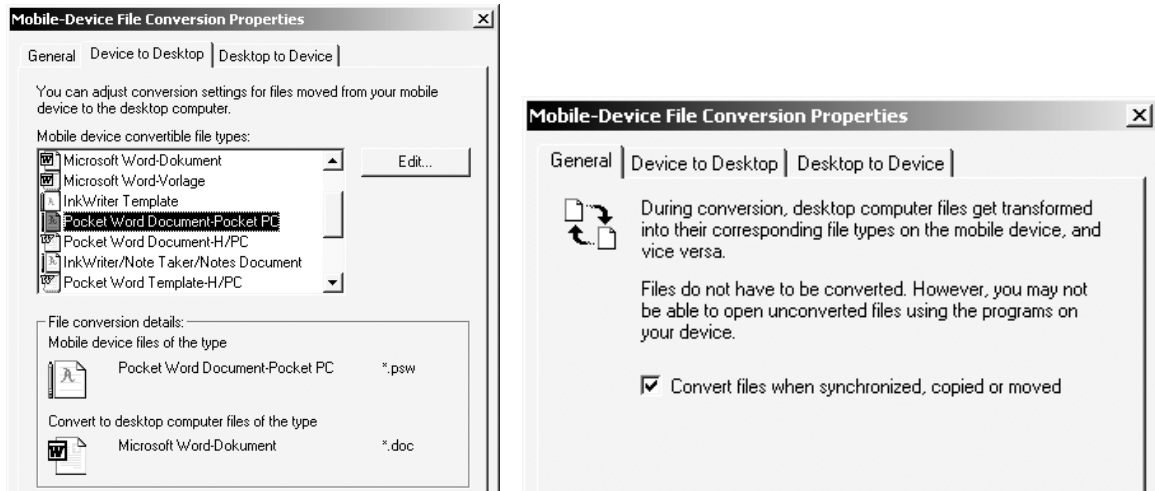


Figure 22: ActiveSync: File Filter Configuration Dialogs

By default this conversion is enabled (see Figure 22 right screenshot) and ActiveSync comes equipped with the following filters [MSFILEFILTER]:

- Pocket Word (.pwd) to Microsoft Word (.doc)
- Word (.doc) to Pocket Word (.pwd)
- Microsoft Pocket Excel (.pxl) to Microsoft Excel 5.0 (.xls)
- Excel (.xls) to Pocket Excel (.pxl)
- Windows bitmap (.bmp) to Windows CE 4-color bitmap (.2bp)

Additional Filters from third parties can be installed and are also configurable through this dialog.

4.4.14 Connection Manager

Central management point for all connections, including VPN connections is the Connection Manager. The Pocket PC design guidelines [PPCLOGO], require Logo certified Pocket PC applications to establish outside connections using the Connection Manager. Either a so called “Internet” or “Work” connection can be configured by the user and can then be used by applications through the Connection Manager API. Some of the many dialogs to setup and configure connection and modem settings is shown in the screenshots of Figure 23.

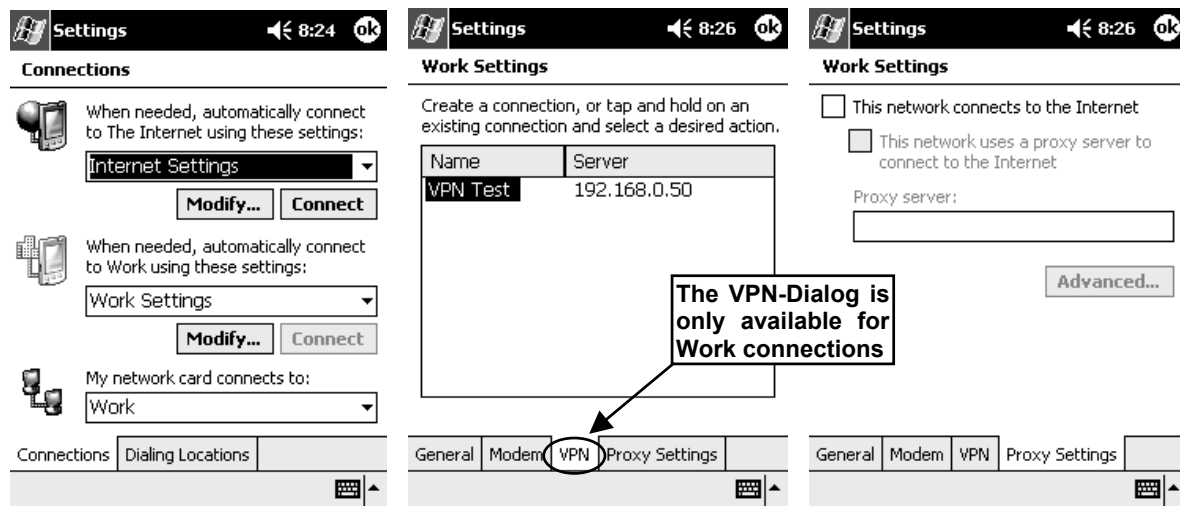


Figure 23: Connection Manager: Selection dialog (l.) and Work configuration dialogs (m. + r.)

The two connection types that Microsoft thought to be the most common connections made by mobile devices are the direct connection to the Internet by calling an Internet Service Provider (ISP) or dialling into a company's network. Additionally the user can submit information about proxy servers, under the "Proxy Settings" tab as seen in Figure 23 on the right, this allows to access the Internet through the company's proxy when dialled into the company. Using the "VPN" tab allows using the VPN client (as described in 4.4.11) to connect to VPN-endpoints specified by hostname or IP address.

The Connection Manager attempts to simplify connection establishment by detecting which connection to use for each request: The internet connection is initiated for requests containing a dot, like entering "http://www.server.com" in Pocket Internet Explorer (PIE) the mobile device starts establishing the internet connection. If no dots are found in the network path the Connection Manager thinks its a local resource on the company network and initiates the work connection, like in "http://server".

The Connection Manager helper API contains a number of calls that make it easier for applications to establish or use an established connection. To detect an already established connection applications can use the `IsAvailable` function. Or they can establish a new connection, which by the way drops other already established connections, by calling the `AttemptConnect` function and providing the network path to connect to, letting the Connection Manager decide which connection to establish. More information about the Connection Manager Helper API can be found in the help [EVT3HELP].

The user is advised not to select the "Save password" option when providing the login information (Figure 24 left shows that dialog), because saving the password means that the user is only informed when a connection is initiated and established (see Figure 24 on the right for an example on the notification received when a modem connection is established).

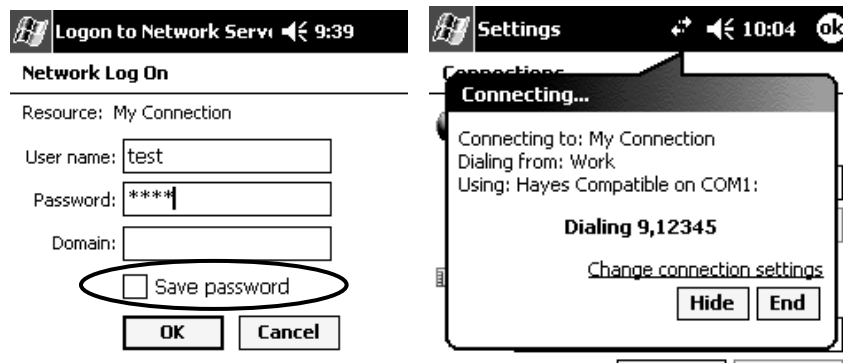


Figure 24: Connection Manager: Logon credentials dialog (left), Connecting (right)

If the user does not save the password, he is presented with the Network Log On dialog (Figure 24 left) every time an applications tries to establish an outside connection.

4.5 Pocket PC 2002 power management

Under the settings the user can specify the time after which the device goes to sleep. Three values define when a Pocket PC 2002 mobile device goes to sleep:

- On Battery Power go to sleep after being idle for xx seconds (`BattPowerOff`)
- On External Power go to sleep after being idle for xx seconds (`ExtPowerOff`)
- When waking up due to events (for example an appointment) go to sleep after being idle for xx seconds (`WakeUpPowerOff`)

The first two of these values can be set using the Power settings dialog as depicted in Figure 25. All the values are also stored for retrieval in the registry³², to actually change the settings the application can use `SystemParametersInfo` (SPI) functions³³.

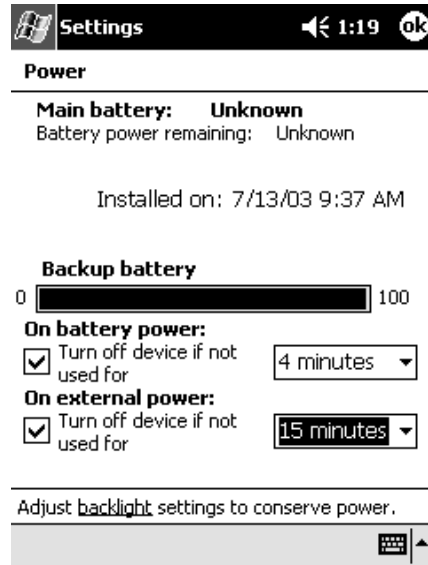


Figure 25: Power settings dialog

The automatic suspend can be initiated by an application. Different ways are possible, one is calling a dedicated function in the GWES API (as described in [TACKE2001]) another way is simulating the hardware power button being pressed (as described in [EVT3HELP]).

Also it is possible to prevent an automatic sleep, by resetting the idle timer counter before it reaches the idle time. So in the above depicted example the idle timer is set to four minutes so the application must call the function `SystemIdleTimerReset` at least every 360 seconds, to keep the device from powering off. Another way to prevent an automatic sleep is to set each of the three idle values to zero.

More information about the actual power state of the mobile device can be inquired by calling `GetSystemPowerEx`, which will provide information about the main and backup battery, or if the device is connected to external power.

Additionally an application can get notified whenever the power state is changed, read [MSPOWERNOTE] for more details on this topic.

³² `HKLM\SYSTEM\CurrentControlSet\Control\Power` is the registry key for storing the idle timeouts, before the mobile device goes to sleep.

³³ `SPI_SETWAKEUPIDLETIMEOUT` for example sets the `WakeUpPowerOff` value.

4.6 Pocket PC 2002 security functions

After having looked at core, network and power-management functions of the Pocket PC 2002 operating system, I will look at Pocket PC 2002 functions that add to the security of Pocket PC 2002 mobile devices in more detail.

4.6.1 Power-On Protection

If configured³⁴, the mobile device will ask for a password or other user authentication to unlock the mobile device if the mobile device is switched on or rebooted. To get exactly this behaviour the idle time in the password dialog must be set to “0 Minutes”, so that once the device is switched off, the password is required after it is switched on again. Using the power management options seen in 4.5, the device can be turned off after a given idle time, so that a password is required to turn it back on again. This configuration is preferable, and will be called “Power-On Protection”.

If the power-on-protection is enabled, establishing an ActiveSync connection and setting the mobile device into the cradled state also requires to enter the password on the desktop machine, ActiveSync allows to save this password on the desktop side.

Setting no password is not suggested, as this means everyone can access the mobile device. Pocket PC 2002 devices are not made for multi user environments, so no login name is required, and once logged on, the mobile device is completely open. Only if applications or network connections require additional authentication the mobile device might ask for a username or password again.

The power-on-protection is normally a password with two different strengths:

- Weak: 4 digit PIN number
- Strong: 7 characters long alphanumeric (capital letters, digits, and special characters)

Apart from the poor security offered by a 4 digit PIN, it shall be noted that if used on a mobile phone edition the use of PINs that begin with the same numbers as international emergency numbers will reveal parts of the password (911x, 112x, 08xx) as the password dialog will remove the stars, that normally hide the entry (as shown on the right in Figure 26).

But this only happens in the phone edition. The numbers, which will produce this behaviour are stored in the registry under HKLM\Security\ECall and unnecessary emergency numbers can be removed manually.

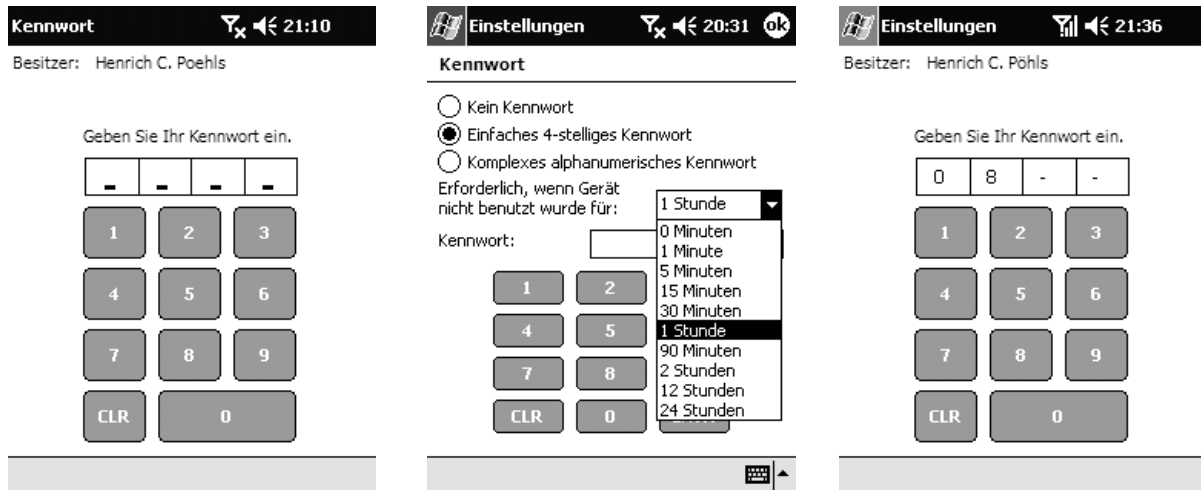


Figure 26: Standard PIN screen (l.), password settings (m.) and revealed emergency number (r.)

If the user enters a wrong PIN or password, the mobile device will increasingly delay the time it takes until a password is verified as correct or wrong. During the first five wrong log-in attempts nearly no change is noticed, but after the tenth log-in attempt the waiting time has increased to around 10 seconds.

With 15 wrong attempts the waiting time has already increased to roughly a minute and will increase further.

³⁴ Power-On-Protection is configured under: settings → system → password.

This makes brute-force password guessing a too lengthy attack, but even if the legitimate user after some wrong attempts finally enters the correct PIN or password the wait is enforced. So if an attacker has tried to gain access to an unattended mobile device and has tried more than 10 times the user will notice that attempt, because of the increased login time. Entering a couple of wrong passwords can of course be used to temporarily deny access to that mobile device, but requires physical access.

The standard power-on-protection can be enhanced, it can even be a biometric based authentication as in the Compaq iPAQ 5450 model, which has a built in fingerprint reader, which has extra settings as depicted in Figure 27. So the power-on-protection can be extended by third parties (for example PointSec [POINTSEC]) or by OEMs (for example Compaq). This allows increasing the security and maybe using the credentials supplied during power on even beyond the power-on-protection. [MSQ314989] describes how to replace the standard password dialog, the example replaces it with a graphical grid were the user has to draw lines as a password.

One of the enhancements shown on the right in Figure 27 is that the user can specify a maximum number of wrong logon attempts after which the mobile device will erase the memory. This enhancement is also available from Compaq for other Pocket PC devices using Compaq's Security Enhancement for Microsoft Pocket PC [COMPAQSECENH], but is it not available or included in the standard Pocket PC 2002 power-on-protection.

Tests have shown that if the maximum number of logout attempts is reached the mobile device requests the user's assistance in order to reset the device. This means that this setting will not automatically wipe the mobile device's memory once the number of logon attempts has been reached, which decreases the effectiveness of this option dramatically.

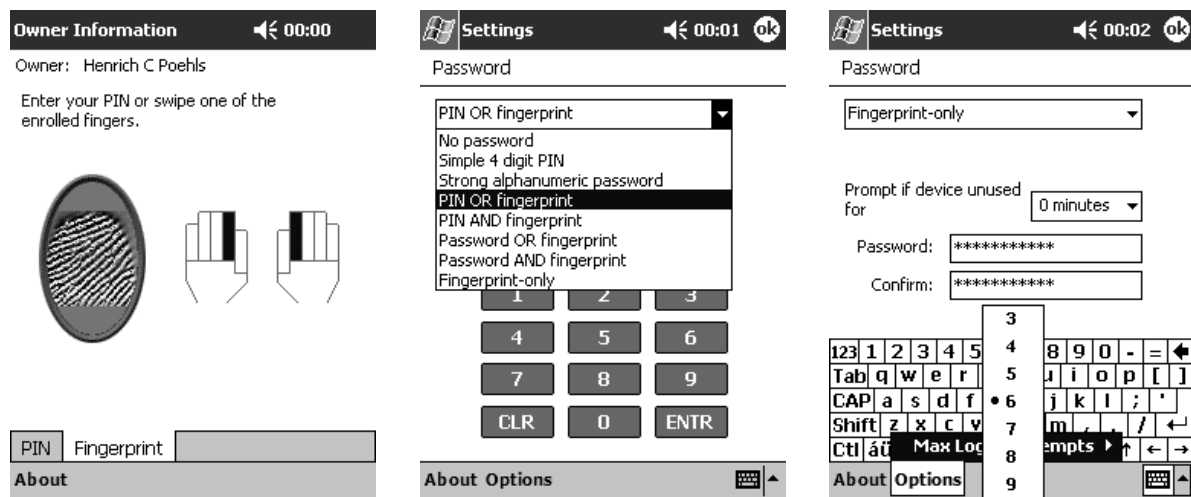


Figure 27: iPAQ5450: Fingerprint-Login (l.), PIN/Fingerprint combinations (m.), max logon attempts (r.)

Further applications can make use of the system's password, as the API function `CheckPassword` allows applications to check whether a password supplied as argument is correct or not. So any application can ask the user to provide the power-on password and act accordingly after checking it through `CheckPassword`. But using `CheckPassword` an application can send hundreds of wrong queries to an unlocked device, as this function does not trigger the previously described lockout mechanism, which was tested using a small eVB program (see Appendix E.1). The function also exists in a RAPI version³⁵, but RAPI can only be used over established connections, meaning the password must be provided to the ActiveSync application on the desktop side (as described in chapter 4.4.13.4).

During a chat with Microsoft tech people [MSTECHNET2002] the following problem got mentioned: If an alarm event is triggered the mobile device wakes up from sleep mode and shall display the power-on-protection dialog. Instead it displays the alarm notification on top of the screen, revealing the information associated with the alarm.

³⁵ `CeCheckPassword` is the RAPI version of `CheckPassword`



Figure 28: Alarm notification displayed on top of the power-on-protection dialog on wake up

According to the Microsoft tech people in the chat “[...] there is no setting that will force the pin security password to the top” [MSTECHNET2002]. How the effect looks like can be seen in Figure 28.

With this bug a malicious program could smuggle information out of a locked device, if the malicious program is started before the mobile device is locked. Then when still running the malicious program could trigger alarms, displaying information.

But even without a malicious program, this causes information to leak out of a locked mobile device.

4.6.2 Security Support Provider Interface (SSPI)

The security support provider interface (SSPI), shall supply networking applications with a defined interface to different security functions for authentication and encryption.

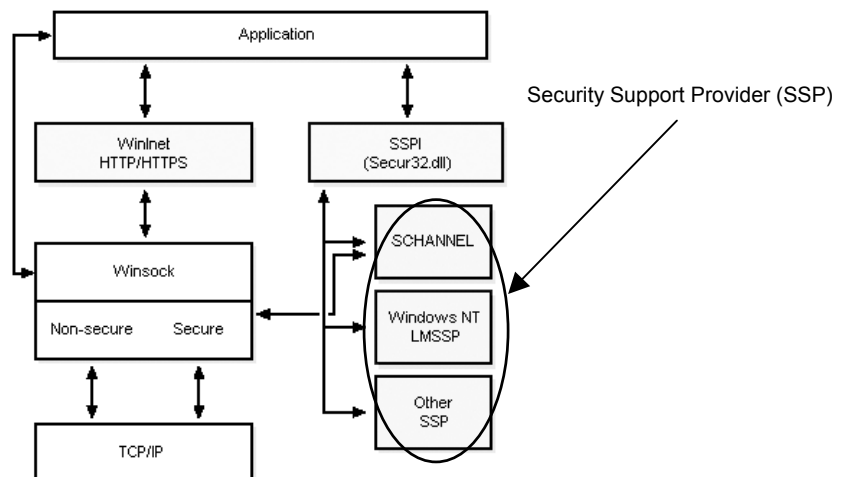


Figure 29: Overview of the SSPI [EVT3HELP]

Different Security Support Provider (SSP) can be installed and used by the application through the SSPI. One such application is the Internet Explorer, it uses the SCHANNEL SSP for SSL connections, also the Windows NTLM SSP is used to allow network applications to use NT LAN Manager authentication.

4.6.3 Cryptography API (CAPI)

Additionally to using the SSPI to provide an abstract layer to communication security, the cryptography API (CAPI) defines an application program interface (API) to cryptographic algorithms for encryption and decryption. Again similar to the SSPI below the CAPI different so-called cryptographic service providers (CSPs) are located to implement different algorithms.

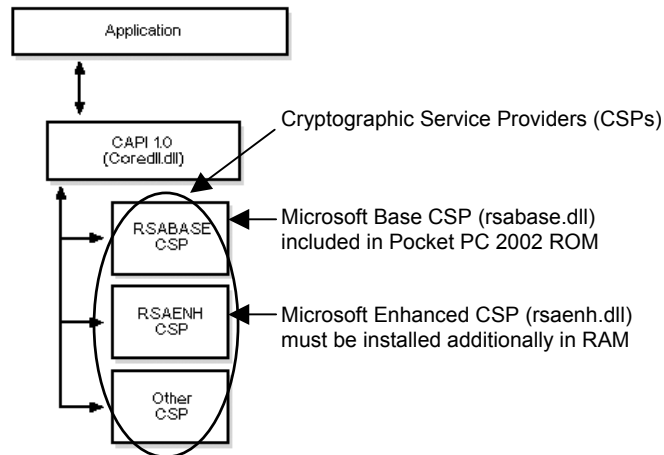


Figure 30: CAPI overview [EVT3HELP]

Each CSP has its own key database to store persistent cryptographic keys. These key databases can contain different keys all used by one CSP, for different purposes, such as session keys (which will be deleted after the session) and public or private keys. The keys normally do not leave the key database, as the CSP is responsible for the handling of keys, but sometimes applications need access to those keys, then the CSP can be instructed to export such keys using the `CryptKeyExport` function. As keys are sensitive the CSP exports the keys in an encrypted state, only under the following special circumstances the keys can be exported in a non encrypted state:

- The key is a public key and it is exported as a public key
- The key is a private key and the CSP explicitly allows to export it without encryption

Only under the two above-mentioned circumstances an application shall be able to get an unencrypted key exported to their memory, for most security protocols the application has to specify an encryption key under which the exported key is encrypted. Nearly all keys can be exported, but only secured by being encrypted under a key supplied by the application that requests the export.

Out of the box, Pocket PC 2002 devices have the basic MS Base CSP installed³⁶, which supports the following algorithms with the indicated key length [MSCSP]:

Algorithms supported by MS Base CSP:	Key Length:
RSA public-key signature algorithm	512 bits
RSA public-key exchange algorithm	512 bits
RC2 block encryption algorithm	40 bits
RC4 stream encryption algorithm	40 bits
DES	56 bits

Table 4: Algorithms and key length of the default MS Base CSP

Additionally the user can install other CSPs. From a security standpoint, at least the Microsoft Enhanced CSP [MSCSP], which supports the above algorithms with longer keys and additionally TripleDES, shall be installed.

4.6.4 Digital Certificate Handling

Windows CE allows using digital certificates according to the X.509 standard. Digital certificates are used for authentication purposes for example during SSL sessions in Pocket Internet Explorer (PIE) (see chapter 5.4.4 for more details on PIE and SSL).

³⁶ `rsabase.dll` is located in the `\Windows` directory

The mobile device has no software that allows viewing or editing the digital certificates by default, but [MSQ322956] presents an additional software (named Addrootcert PowerToy), which allows installing additional certificates into the root store. This is done using Cryptography API functions to access the certificate store (see previous chapter 4.6.3). This software only allows to install self signed root certificates, but it comes with source code, so it can be modified to also remove certificates if needed. Another way to add a certificate to the mobile device is putting the ASN.1 encoded X.509 certificate into the registry under `HKLM\Comm\SecurityProviders\SCHANNEL\CA`, which is described in more detail in [MSSCHANNEL].

Pocket PC 2002 comes with 12 root certificates, located in the `schannel.dll` [MSSCHANNEL], from the following certification authorities:

- Verisign (Class 1, Class 2, Class 3, Class 3)
- RSA Data Security
- Cybertrust
- Thawte (Server CA, Premium Server CA)
- MS SGC Authority (1024bit, 2048bit)
- Entrust (1024bit, 2048bit)

4.6.5 Smartcard Support

None of the Pocket PC 2002 devices in the market by January 2003 have a built-in smartcard reader for the use with the Pocket PC OS. Only those mobile devices with built-in GSM phones have one for the GSM subscriber identity module (SIM) used by the GSM phone. But the Windows CE operating systems generally supports smart cards. If an application supports smartcards it can use a smartcard service provider (SCSP) and a smartcard reader driver to access the hardware.

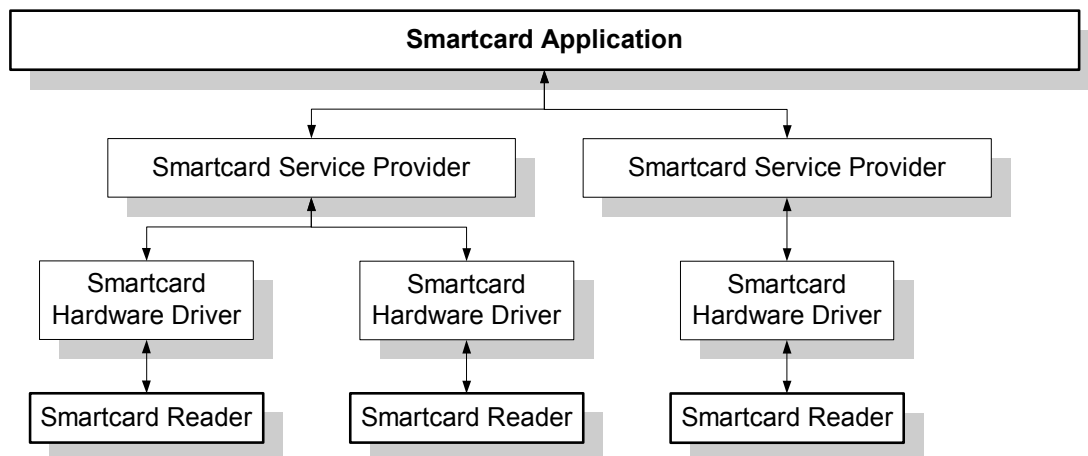


Figure 31: Smartcard support through SCSPs and Smartcard Hardware Drivers

No smartcard service providers or drivers are supplied with the Pocket PC 2002 operating system [MSSSMARTCARD], but some vendors of smartcard hardware provide drivers and service providers for Pocket PC 2002.

So basically none of the components shown in Figure 31 are provided by Windows CE operating system, but need to be implemented by third parties. But the smartcard subsystem of Windows CE 3.0 conforms to the Interoperability Specification for ICCs and Personal Computer Systems [EVT3HELP], which shall allow to port existing applications.

4.6.6 Trust-Model [not implemented in Pocket PC 2002]

The functionality described in this chapter is part of the Windows CE 3.0 operating system, on which both Pocket PC 2002 and Smartphone 2002 are building. It is not implemented in Pocket PC 2002, but found its way into the Smartphone 2002.

It offers a way to restrict applications and what applications can do³⁷ on the Smartphone 2002 and is referred to as enhanced Authenticode [SMARTPHONE2002SDK].

An undocumented function that can at least keep unwanted executables away from the mobile device is described in chapter 4.6.7, but that function does not offer the security offered by this trust-model.

³⁷ Also called: trusted application model [EVT3HELP]

As this security feature is not available on Pocket PC 2002 devices I will only shortly describe some of the security functions offered.

One of three levels of trust is assigned to all Smartphone 2002 applications:

- Privileged trust
- Unprivileged trust
- Untrusted

The level of trust that an application gets is determined by the certificate that was used to sign the application and the settings of the Smartphone device.

Applications are signed using basically two signatures a privileged or an unprivileged, or they are not signed. The Smartphone checks this signature during the load process.

The Smartphone device itself can run the application in two different modes, or it can decide not to load the application at all:

- Trusted Mode:
Allow full access (normally not granted to custom applications by the operators)
- Untrusted Mode:
Restricted application execution and restricted access to the registry, APIs, and System files.

The process is described and depicted also in [SP2002SDKHELP] in more detail; Figure 32 shall provide a quick overview of the process.

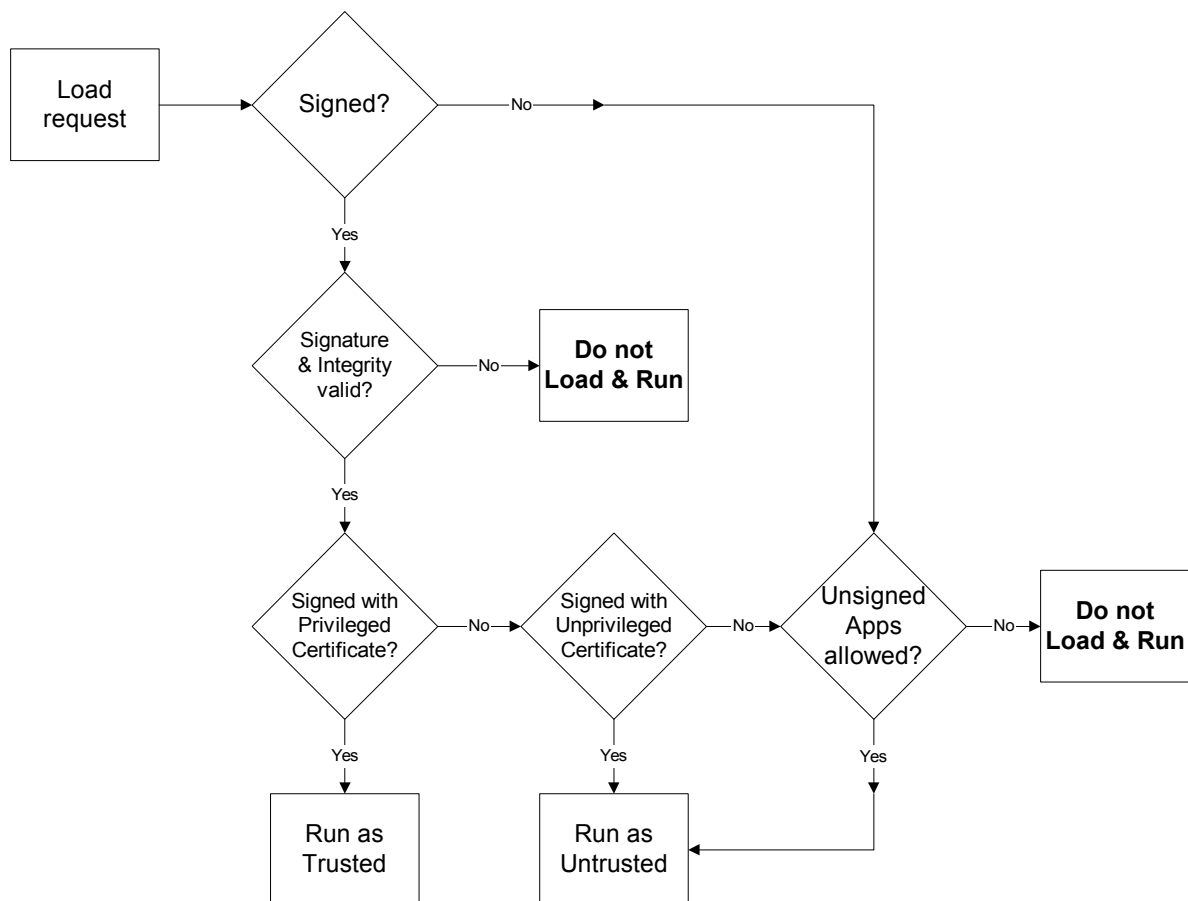


Figure 32: Application execution under the Smartphone 2002 trust-model

The decision, if an application is executed or not, is made upon the application's signature according to the settings of the Smartphone device.

The Smartphone device can be configured to only allow signed applications, this was how the first of the Smartphone 2002 devices, the SPV from the operator Orange was delivered to the customers.

This allowed only applications certified by Orange to run on the SPV, which a lot of users and developers did not like. Especially the use of freeware and cheap shareware was not possible on the SPV, because the certification of an application by Orange costs money, and so freeware developers did not get their applications signed by Orange.

At first users tried unsupported ways to enable the execution of unsigned applications [SPVUNLOCK], but Orange then reacted and is now offering a procedure to enable the execution of unsigned applications for development purposes [SPVUNLOCKDEV].

4.6.6.1 Registry Protection of Trust-Model

Applications running as “untrusted” have restricted access to parts of the registry. The following registry hives can only be changed (write access) by applications, which run as “trusted”. But they can be read by all applications:

- HKEY_LOCAL_MACHINE\Comm
- HKEY_LOCAL_MACHINE\Drivers
- HKEY_LOCAL_MACHINE\HARDWARE
- HKEY_LOCAL_MACHINE\SYSTEM
- HKEY_LOCAL_MACHINE\init
- HKEY_LOCAL_MACHINE\WDMDrivers

4.6.6.2 API Protection of Trust-Model

Certain APIs can also only be accessed by trusted applications. For the Smartphone 2002 mobile devices the Radio Interface Layer APIs are protected to give an example.

For more details which APIs are restricted for untrusted applications see [ALFORQUE2000], [SP2002SDK] and [EVT3HELP]. Some flags especially those for debugging are also protected.

4.6.6.3 File Protection of Trust-Model

All objects in the file system that have the attribute “System” are read and write protected for untrusted applications. Only trusted applications can get access to these system files of the Smartphone 2002. Please see [SP2002SDK] for more details.

4.6.7 Policy Restriction [undocumented Pocket PC 2002 function]

One of the many registry keys in the Pocket PC 2002 registry allows to enable a special function to restrict new programs. This functionality is not documented by Microsoft, but the registry changes needed to enable this restriction policy can be found on the internet, for example in [POLICYTWEAK].

As this function is not found documented on Microsoft’s websites, nor in any of the SDK’s documentation. So the functionality described in this chapter is not assured by Microsoft, further the authors that found or described the registry settings (like in [POLICYTWEAK]) cannot give assurance on the functionality either, as they are not allowed to reverse engineer operating system functions without copyright infringement. But the functionality offered by this policy restriction could help to stop the entry of unknown and potentially malicious executable code, so its functionality is reviewed at a higher level in the following.

The mentioned website [POLICYTWEAK] suggests to make the following change to the registry to enable the setting for policies (or “Richtlinien” in German language):

Rename `HKEY_LOCAL_MACHINE\ControlPanel\AdminPassword\Redirect`
to the following: `HKEY_LOCAL_MACHINE\ControlPanel\AdminPassword\xRedirect`

After a reset the new icon can then be found in the settings dialog (see left screenshot in Figure 33).

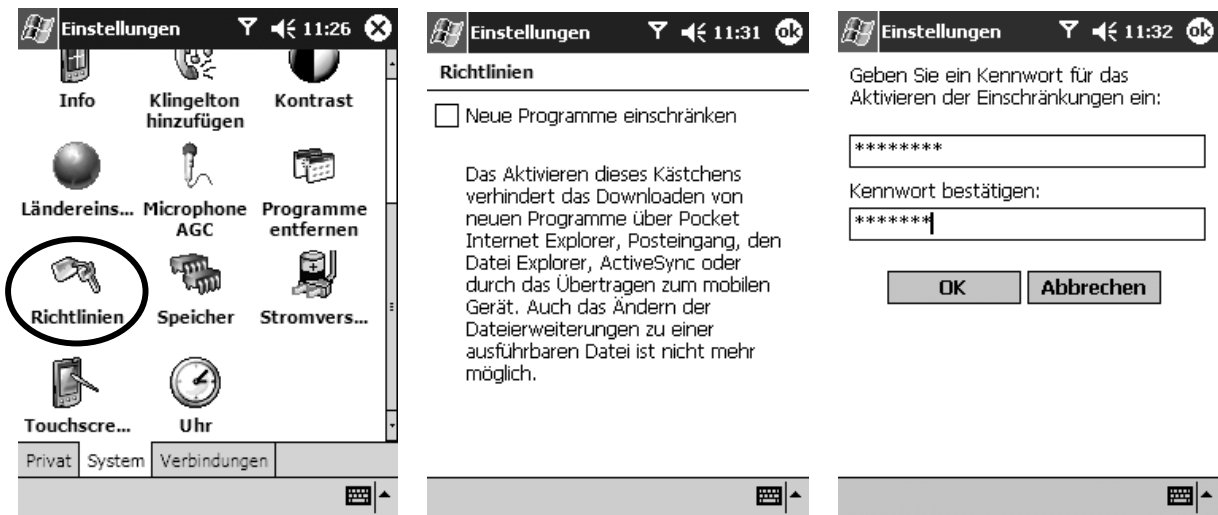


Figure 33: Undocumented Policy Restrictions: Icon (l.), dialog (mid.), administrator password (r.)

After enabling the policy the dialog requires the entry of a password to protect against further changes of this setting. This administrative password needs to be a strong password (see 4.6.1 for details).

The dialog states that enabling the policy will “prevent the download of new programs through Pocket Internet Explorer, Inbox, the File Explorer, ActiveSync or through the transfer to the mobile device. Also the change of file extensions to an executable file is no longer possible.” (see Figure 33 middle).

The first objective is observed to work well in restricting the reception of files with the extensions `exe` and `cab`, other extensions seems no to be affected (especially not `*.vbs`). But this is an undocumented function so this might be further configurable or apply also to other extensions, which was not checked by this work.

When downloading a file with the extension `exe` or `cab` with Pocket Internet Explorer or when copying it to the device using ActiveSync, errors are shown as depicted in Figure 34 and Figure 35.

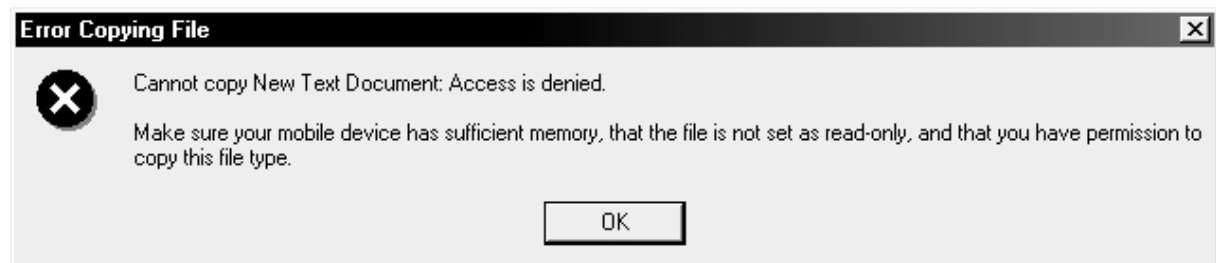


Figure 34: Error message when copying an `exe` or `cab` file to the mobile device with policy restriction

When the user receives a file with the extension `exe` or `cab` as an attachment via e-mail, it is simply not executed, and trying to save it from Pocket Outlook is not possible as no save dialog can be opened.

But the user can download files containing executable code to the mobile device even if the restriction is enabled, if it is compressed (i.e. in a zip-file) or if the file’s extension has been removed or renamed. The executable file is just not allowed to have the extension `exe` or `cab`.

This can be observed, as it is possible in Pocket Internet Explorer to download the file from the second link from the test website [WEBTESTPOLICY] can be downloaded to the mobile device without restriction. PIE is not offering a save dialog to files with no extension, instead PIE just displays the contents.

It is possible to save executables received with Pocket Outlook that have a renamed or no extension.

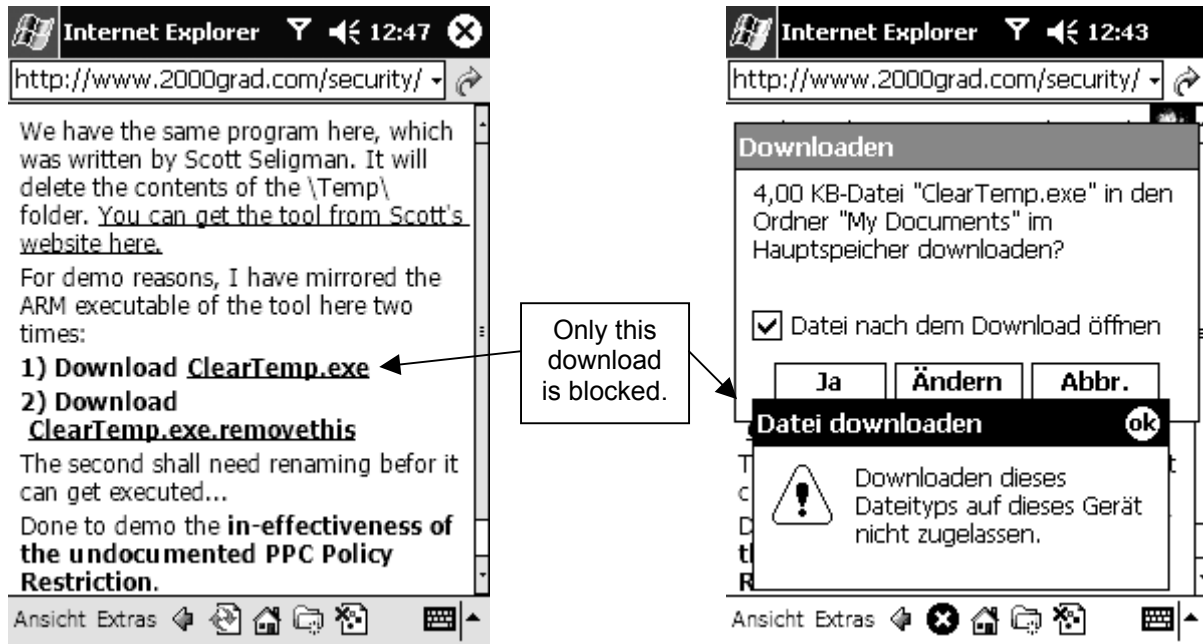


Figure 35: Downloading exe files through PIE is blocked if the policy restrictions is enabled.

This shows that renamed executable files can be transferred onto the mobile device, therefore the policy must also restrict that those can be renamed back into *.exe or *.cab files in order to restrict that these renamed executable files can get executed again. This is the second objective of the policy restriction.

It can be observed that with an enabled policy the renaming to for example *.exe is only restricted in the default file management application File Explorer. If the user has access to another file management application for example the Reso Explorer 2003 (from [RESCOWEB]), the user can rename the executable file and will be able to execute the program after that with no further restrictions.

Also the user cannot influence the extension when using the save dialog of PIE, but the user can freely chose the extension when saving an attachment without an extension in Pocket Outlook. The following screenshot shows the dialog reachable by a tap-and-hold on the attachment:

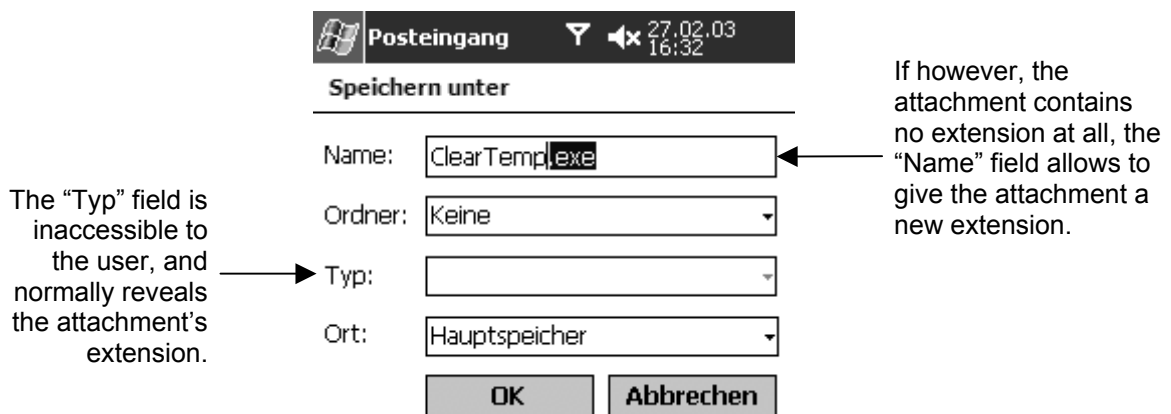


Figure 36: Save Dialog of Pocket Outlook allows to assign an extension

As shown, this undocumented function does not implement the restriction throughout all applications of the Pocket PC, and as the exception in the save dialog of Pocket Outlook shows, not even in all the standard build-in applications.

A further investigation of the registry has yielded that at least the following registry keys are all modified or created, once the user enables the policy restriction:

HKCU³⁸\Software\Microsoft\Windows\CurrentVersion\Policies\Shell\

- `AdminActive`: This is set to 0x1, if the policy restriction is enabled, setting this key back to 0x0 disables the policy restrictions without the requirement to provide the administrator password.
- `AdminInfo`: This contains a representation of the password, this work has not investigated further if it is stored encrypted, as it is not needed to enable or disable the feature directly from the registry (see above).
- `NoAutoRun`: This is set to 0x1. This is the option to disable the automatic run of the auto-run application on insertion of removable storage media (see 4.3.5.2), and can be used even without enabling the policy protection.
- `NoExternalExes`: This is set to 0x1. This causes the mobile device to reject the local storage of executables like described above, can be set to 0x1 even without enabling the policy protection, to cause the rejection.
- `NoRapiRegMod`: This is set to 0x1, but was not further investigated.
- `NoRunDlg`: This is set to 0x1, but was not further investigated.
- `Restriction`: This is set to 0x1, but was not further investigated.

As mentioned earlier when discussing the not implemented trust-model, this shows that the restrictions are limited in their security. They can be disabled even without knowledge of the administrator password provided in the dialog, or even more covertly, the imposed restrictions can be turned off, while the configuration dialog (see Figure 33 middle) would still show them as enabled and ask for the administrator's password.

With the limitations not restricting interpretable files (like *.vbs) and only blocking files by their extension it can offer a false sense of security, especially to users of additional third-party software. It will also not stop a sophisticated, treacherous user from transferring executables, as he or she might just use eVB programs that can disable the restriction by making the required registry changes. Or a third-party registry editor can be transferred using unnamed e-mail attachments and renaming them back to executables.

However, if enabled it can reduce the risk that a user accidentally transfers executable (*.exe) or software installer (*.cab) files to the mobile device.

³⁸ HKCU stands for HKEY_CURRENT_USER

5 Pocket PC 2002 Applications

5.1 File Explorer

The File Explorer is used to navigate through the file systems. It lists filenames and some additional attributes (date and file size). Like the desktop version it enables the user to execute programs and open documents with the associated application by tapping on them. The decision which application is used to open which document is based on the file extensions. The association of a file extensions with an application is stored in the registry.

5.1.1 File Extension not Displayed

Malware on desktop windows systems (for example VBS/VBSWG.gen@MM [NAIVILVBSWG]) have shown that the omission of the last's file extension can be abused by malware to get invoked and executed. This problem occurs in File Explorer, as it cannot be configured to display the complete file extension. The last extension is always stripped off. The Pocket PC application design guide also states that "it is strongly recommended that the .XXX extensions are hidden in the application itself" [PPCLOGO] as a result nearly all application shall behave like the File Explorer, when listing filenames. Then Frog.jpg.vb is displayed as Frog.jpg not only in the File Explorer (see Figure 37).

Not even the rename function will display the true extension to the user.

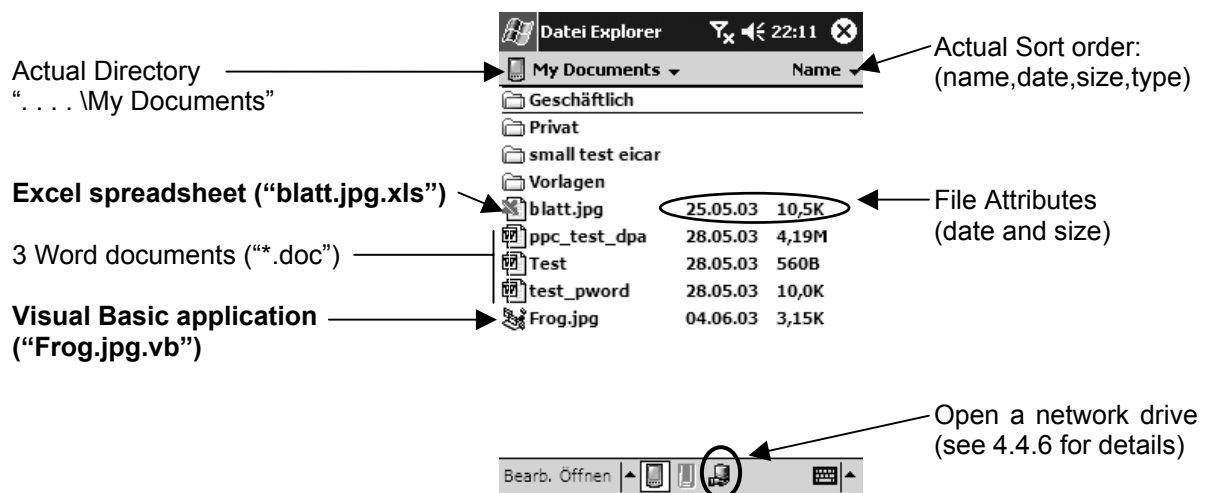


Figure 37: File Explorer does not reveal the last file extension

There are non conforming third-party applications (like the earlier mentioned RESCO Explorer [RESCOWEB]), which will allow the user to see and edit the true extension and access the file attributes.

5.1.2 Display the file extension using "Beam File ..."

An option to view the full extension if no additional third-party applications are installed on the mobile device is to select "Send via E-mail..." or even better "Beam File..." from the context menu accessible by tap-and-hold (see 4.3.7.2 for a picture and a description of tap-and-hold).

Selecting "Send via E-mail..." will open Pocket Outlook and compose an empty e-mail with the file attached, which normally reveals the extension if the filename is not too long (see 5.5.2 for details). However another complete application (Pocket Outlook) needs to be opened.

Selecting "Beam File..." will open just a dialog, as depicted on the next page in Figure 38.

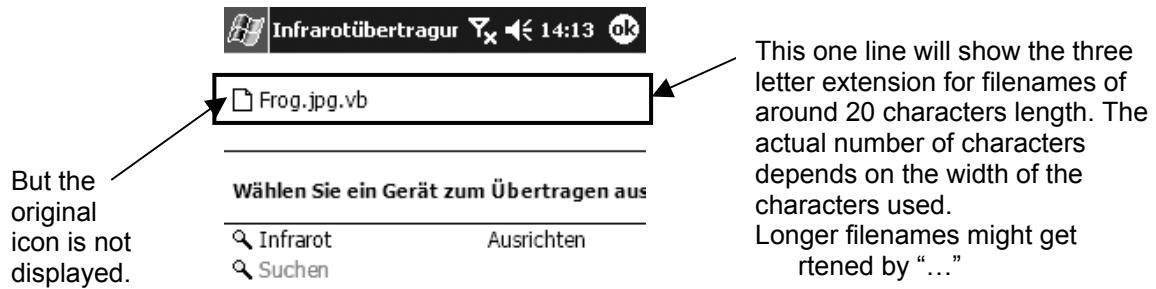


Figure 38: “Beam File...” dialog might reveal the full filename and the extension

The Infrared beam dialog displays the filename and the extension until the width of the one line is reached, then it will get shortened with “...”.

But already the display of an extension, like .jpg, in the File Explorer shall make an aware user suspicious in the first place. “Normal” files do not contain multiple three-letter extensions (like “Frog.jpg.vb”). Additionally the icon might not correspond to the extension shown. But if the user is not aware enough the extension hiding of File Explorer can be facilitated for malware contamination or distribution.

5.1.3 Hidden Files Not Listed by Default

Additionally the File Explorer does by default not list files with the attribute `Hidden`. The user can change that by invoking the context menu from the white space of the File Explorer Window using tap-and-hold. Then the option “View All Files” should be checked, File Explorer keeps that setting.

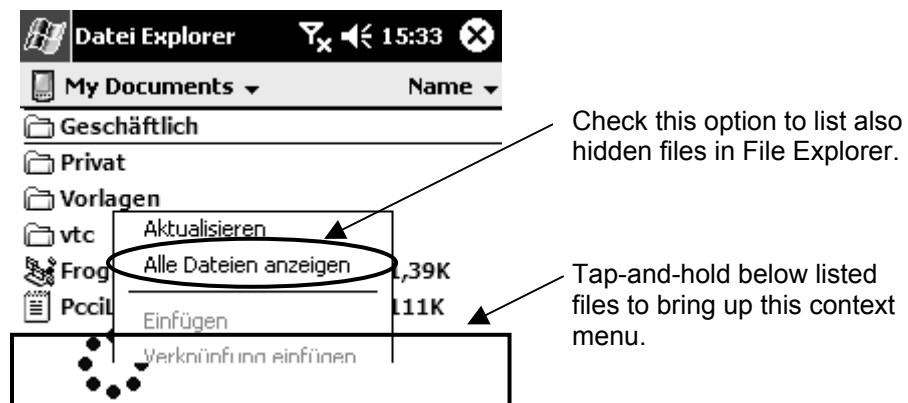


Figure 39: File Explorer: Activate “Show All Files” to list also hidden files.

5.2 Pocket Word

Pocket Word allows viewing and editing documents written with Microsoft Word. It only has limited functionality compared to the desktop versions of Word. Normally it uses a special format called Pocket Word document (*.psw), but it can also open and save in the standard desktop Word formats (*.doc, *.dot) as can be seen on the left in Figure 40.

Limitations when viewing standard word documents include limitations in display capabilities, like:

- Graphic objects are not displayed
- OLE-objects are not displayed
- Frames and their shades are not displayed
- Shades of tables are not displayed

Some of these limitations can be observed in Figure 40 comparing the original (middle) Word document with the version displayed in Pocket Word (right).

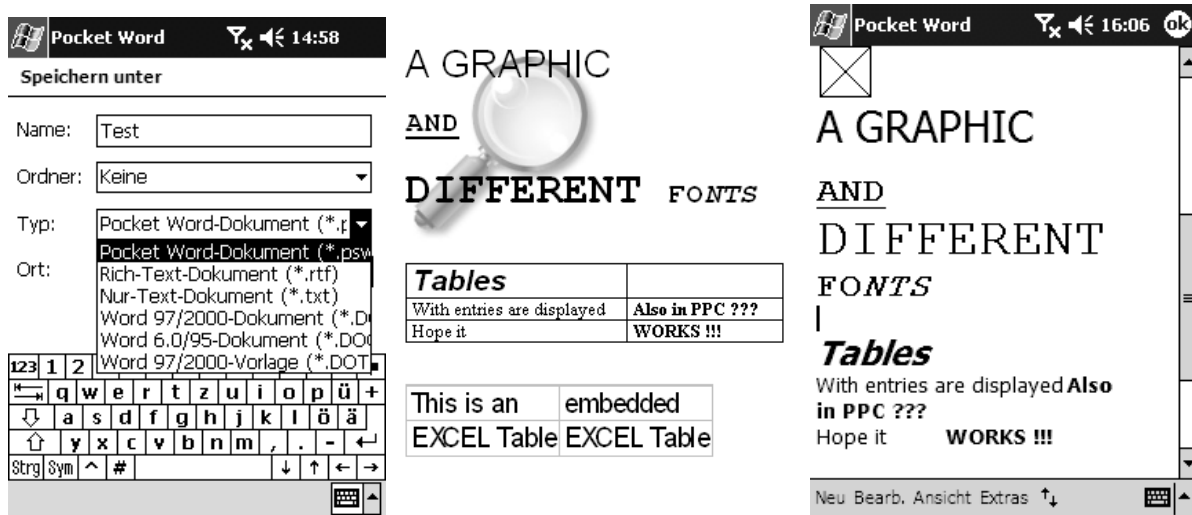


Figure 40: Pocket Word: Saving in different formats (l.), original DOC-File (mid.) in Pocket Word (r.)

Pocket Word also has limitations when it comes to active content, this makes it a secure viewer for macro contaminated documents. As Pocket Word does not know macros, it also does not execute any malicious macros. Instead Pocket Word totally ignores any embedded macros, but this means that a user gets no information that a Word document (*.doc) might contain macros. This facilitates malware distribution.

Also Pocket Word has limited support for security functions: It does not support office document protection.

For example: Let us assume that a document is protected and editing is forbidden, this is not honoured by Pocket Word. Moreover, if the document is edited and then saved, parts of the original formatting but most important also the document protection is lost.

5.3 Pocket Excel

Like Pocket Word, Pocket Excel offers the user a subset of the Excel functions known to users of the desktop version of the spreadsheet program. And like Pocket Word, Pocket Excel does also not execute macros. The problem that macros are unnoticeable for users is also relevant to users of Pocket Excel.

5.4 Pocket Internet Explorer

Pocket Internet Explorer (PIE) is the Pocket PC 2002 version of the Microsoft Internet browser. The following browser information is sent to the web server during a HTTP request by the test mobile device:

```
Version: HTTP/1.1
Method: GET
URI: /
Accept: */*
UA-OS: Windows CE (POCKET PC) - Version 3.0
UA-color: color16 (other possible values: mono2, mono4, color8, color24, color32)
UA-pixels: 240x320
UA-CPU: ARM SA1110
UA-Voice: TRUE (or FALSE if the mobile device is not equipped for voice telephony)
UA-Language: JavaScript
User-Agent: Mozilla/2.0 (compatible; MSIE 3.02; Windows CE; PPC; 240x320)
Host: www.2000grad.com (or another server address, depending on the request)
Connection: Keep-Alive
```

This shows that Pocket IE identifies itself to the web server as MSIE 3.02, but that can be changed³⁹ as it might prevent the user from viewing certain website that check for more up-to-date versions. It also gives away information about the operating system and the hardware.

I will look at features that are interesting from a security standpoint, as:

- supported active content (i.e. scripting languages)
- secured HTTP connections via SSL (HTTPS)
- authentication
- settings (i.e. cookies, dis-/allow scripting)

5.4.1 ActiveX

Pocket PC does not allow downloading and installing ActiveX controls from the web on the fly. Only ActiveX controls that are already installed on the Pocket PC can be referenced by tags in web pages [EVT3HELP]. To install a new ActiveX control a cab-file could be used, which must be downloaded to the mobile device and then installed. To run the ActiveX control that made its way onto the mobile device, it needs to be registered, which is normally done during the install process. Registration can also be performed by running `regsvrce.exe` with the appropriate parameters for the ActiveX control on the mobile device.

Pocket PC does not allow checking for signed ActiveX controls, as it does not support Authenticode security [EVT3HELP]. But Pocket Internet Explorer does check if an object is marked as "Safe-for-Scripting", before creating it [MSACTIVESECURITY]. "Pocket PC is the first Windows Powered device to enable scripting of ActiveX controls in Pocket Internet Explorer. To this end the browser validates that controls are safe for scripting." [MSPIESESECURITY]

This verification is done by calling a special interface inside the ActiveX object during the initialisation, named `IObjectSafety`. If the ActiveX object does not have this interface the user is asked, but "ActiveX Controls are required to implement the `IObjectSafety` interface to further enhance security" [PPCLOGO], if they want to get the "Designed for Microsoft Windows Pocket PC" logo.

If the interface exists the decision whether to properly initialise and instantiate or not to instantiate the ActiveX control by a JScript inside a web page is based on the return values of the `IObjectSafety` interface.

Objects that are not marked as Safe-for-Scripting are the Pocket Outlook Object Model (POOM) and the File Control. This makes access to POOM contact data through a scripted web page, like demonstrated in [CODY2001], no longer possible.

5.4.2 Java Virtual Machine

Java is not supported by Microsoft on Pocket PC 2002 [EVT3HELP]. If a user wants to run Java applets or Java programs a Java Virtual Machine (VM) from a third party vendor (like [JAVAVM]) needs to be installed on the mobile device.

5.4.3 JScript Version 3.0

Pocket Internet Explorer can execute embedded scripts. It understands the Microsoft version of JavaScript named JScript in version 3.0 (see Figure 41), the test script can be found at [WEBTESTJS].

Most of the JScript 3.0 functionality (see [MSJSCRIPTVERSION] for a detailed functionality list) that is offered by the desktop environment is also provided by PIE, except for arrays or regular expressions [CLINICK2000].

³⁹ HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\User Agent\Version



Figure 41: Pocket Internet Explorer shows its JScript version [WEBTESTJS]

Apart from the already mentioned scripting of ActiveX objects (see chapter 5.4.1) the JScript can be used to dynamically write and change the page content (using `document.write`) or to pop-up alert windows (using `alert`). More complicated scripts can be seen at a sample page from Microsoft [MSPIETEST].

Like in almost all software, errors have found their way into the program and have been discovered. One of the Bugs reported [ROTNES2003] causes the Pocket Internet Explorer to stop, using a malformed JScript. I have found this bug to completely exit the Pocket IE on the Pocket PC 2002 test device (a MDA) and also on a Smartphone 2002 device (an Orange SPV), a test page with an implementation of code described by [ROTNES2003] can be found at [WEBTESTBUG].

5.4.4 Tel-URLs

Even though this is not exactly active content it is an interesting feature of Pocket Internet Explorer: Instead of embedding normal HTTP-URLs (`Linked Text`) a so-called Telephone-URL can be used to initiate the dialling of a telephone number on Pocket PC Phone Edition and Smartphone.

A Telephone-URL looks like this: `Linked Text`, clicking on such a link in Pocket Internet Explorer will bring up a dialog on phone enabled mobile devices, asking to call the number 08003306667, as embedded in the Telephone-URL. The resulting popup window is shown in Figure 42.

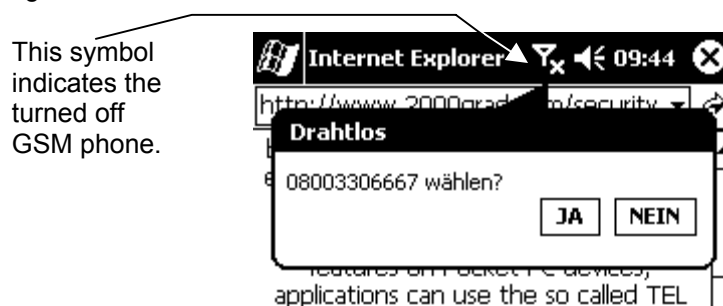


Figure 42: Asking to call, after clicking on a TEL-URL

5.4.5 SSL Connections

There is no possibility to install certificates directly from PIE, so when you are connecting via SSL to a website which does not have a certificate issued and signed by one of the pre-accepted authorities, then the PIE will display a message saying that the certificate is either expired or that the server name is different (see left screenshot in Figure 43).

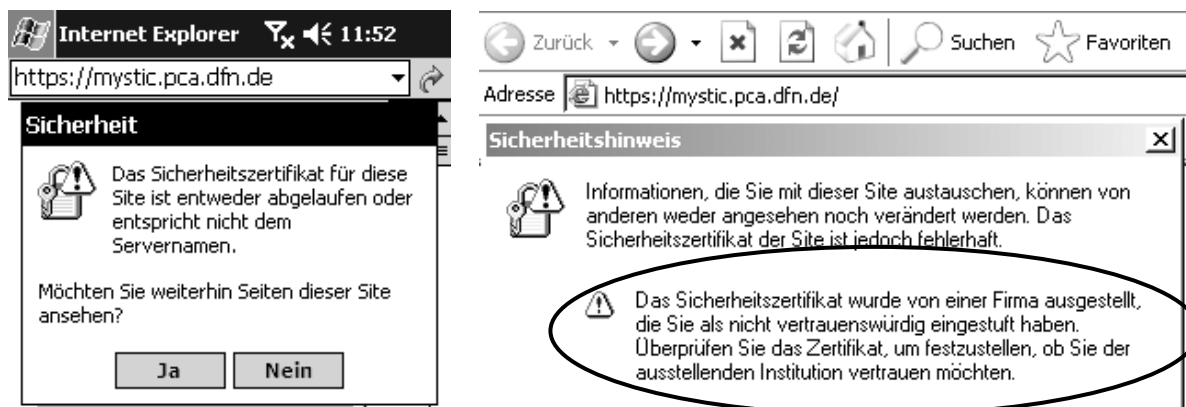


Figure 43: Wrong SSL certificate message in PIE (left) and the correct message in IE 6.0 (right)

What PIE should display is a message stating that the certificate could not be validated against a known root certificate (a list of root certificates can be found in chapter 4.6.4). This correct message is displayed by the desktop version Internet Explorer 6.0 (see right screenshot in Figure 43). Users can, by clicking “Ja” (Yes) in the PIE dialog, trust the certificate for one session and so still connect to the server using SSL. But a user only has very limited information over the certificate, as PIE does not allow to retrieve any additional information apart from presenting this message window, with sometimes wrong information.

I have pointed PIE to different websites that make use of SSL to secure the browser to server connection, with different results:

Fortify for Netscape	SSL-Link: https:// www.fortify.net/sslcheck.html
Fortify for Netscape used to offer a security update for weak export versions of older browsers, they have a web page that shows which cipher and which parameters have been selected during the SSL handshake. Their SSL certificate expired in March 2001. PIE displays the above-mentioned message, but the message is correct, as the server’s certificate has expired.	
Deutsches Forschungs Netz (DFN)	SSL-Link: https://mystic.pca.dfn.de
The DFN runs their own certification authority, which makes their test server’s root certificate, as it is not co-signed, not known to the PIE. PIE displays the above-mentioned message, which does not state the correct reason, which would be “unknown root certificate”.	
Deutsche Bank	SSL-Link: https://meine.deutsche-bank.de
The Deutsche Bank online banking web login is SSL secured and their web server has a VeriSign certificate. PIE displays the website correctly and allows to use the web online banking.	
Amazon	Link: http://www.amazon.de
To change customer details like the credit card details, Amazon uses an SSL secured connections with a VeriSign certificate. PIE displays the website correctly and allows to use online orders and other SSL secured parts of the website.	

Pocket Internet Explorer does not display any hints that a secured link has been established like the desktop Internet explorer versions do. The user has to take a complicated look at the page properties to find out if the connection is secured or not. Selecting View → Properties from the PIE menu brings up the page properties screen as depicted in Figure 44.



Figure 44: PIE page properties reveals SSL secured pages

Only through checking the page properties screen the user can verify that the connection with the web server has been secured by SSL.

As I said earlier, Pocket Internet Explorer does not allow the installation of new certificates, but the Adrootcert PowerToy (see chapter 4.6.4) allows installing additional certificates into the root store.

For example: After installing the DFN certificate and restarting the browser (stop the running PIE through the memory settings dialog and start it again), the connection to the DFN SSL test server is made secured by SSL and the HTML-page is displayed without any further warning messages.

5.4.6 Settings

The Pocket Internet Explorer allows setting different settings through an options dialog. It does not allow for more complex security settings as known from the desktop environment. One security relevant setting in the standard settings dialog (see Figure 45 middle) controls whether a warning is displayed when the user leaves a secured (HTTPS) web page and opens a non-secured page (HTTP) or if this warning is suppressed⁴⁰.

Additional settings can be controlled through a GUI by installing the Microsoft PowerToys for Pocket Internet Explorer [MSPOWERTOYS]. Here the user should be able to disable the execution of JScript (see Figure 45 right), I tested this setting and found it to be not effective.

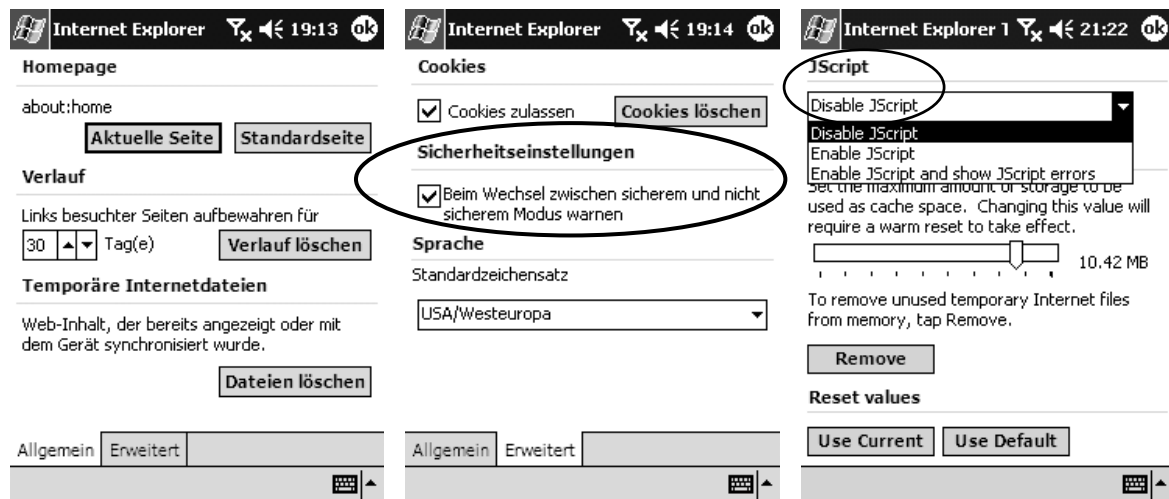


Figure 45: Standard settings in PIE (left and middle) and additional settings in MS PowerToys (right)

To effectively disable the execution of scripts within a web page different registry keys can be used. In the registry hive HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings two keys can be found:

- Security_RunActiveXControls: Setting this to zero effectively disables ActiveX
- Security_RunScripts: Setting this to zero effectively disables JScript

⁴⁰ Corresponds to registry key HKEY_CURRENT_USERS\Software\Microsoft\Internet Explorer\Main\EnableWarning0 being 0 or 1.

Three more settings can be security relevant, mostly from a privacy point of view:

- settings for cookies,
- browsing history and
- cached web pages.

This can all be adjusted through the standard settings dialog, for all three options a button exists, which deletes the relevant data from the mobile device's memory. The cookies are saved in `\Windows\Cookies`, the cache resides in `\Windows\Temporary Internet Files`, and the history in `\Windows\History`, and so the user can also directly manipulate these entries.

5.5 Pocket Outlook

Pocket Outlook is the mobile device version of MS Outlook. It allows managing e-mails (and SMS for the phone edition), appointments and the address book.

The e-mail program on which I will concentrate in this chapter is found under the name "Inbox"⁴¹, appointments in "Calendar" and the address book functionality is named "Contacts".

5.5.1 Connections

Pocket Outlook can use the following protocols to send and receive e-mail messages:

- ActiveSync (Microsoft's synchronization "protocol")
- POP3
- IMAP
- SMTP

Looking at the security of e-mails exchanged between different systems, e-mails can be secured on two different levels:

- application level security and
- transport level security.

5.5.1.1 Application Level E-mail Security

Application level e-mail security would mean that the e-mail is confidentiality and integrity protected from one e-mail application to the other, which can be achieved by encryption and applying digital signatures. This application level e-mail security enables an end-to-end protection between the sender and the receiver, as the e-mail is also protected in the transit systems.

This can be achieved by using defined message security protocols such as:

- Privacy Enhanced Mail (PEM) [RFC1421-1424]
- S/MIME version 3 [RFC2633]
- MIME Security with PGP [RFC2015]
- MIME Object Security Service (MOSS) [RFC1848]
- Security Multiparts for MIME [RFC1847]

Pocket Outlook supports none of the above-mentioned protocols. The often-used S/MIME format, which is supported by the MS desktop e-mail clients MS Outlook and MS Outlook Express [RSASMIME], shall be supported in future versions [MSDNCHAT2002]. So third party programs like PGP Mobile [PGPMOBILE], for sending and receiving PGP secured e-mail messages, or MovianMail [CERTICOM] are needed to achieve the application level e-mail security.

5.5.1.2 Transport Level E-mail Security

To secure e-mails while they are transferred or to allow for secure authentication with the last e-mail server, only the connection to the server is secured. Transport level security will depend on the protocols used for the connection.

For the standard Internet protocols (SMTP, POP3 and IMAP) this is usually achieved by using an SSL/TLS secured connection to the mail server or a VPN connection to the network on which the mail server resides.

The secured TLS connection is usually established using extended SMTP, POP3 or IMAP protocols as defined in [RFC2487] and [RFC2595].

⁴¹ Or "Posteingang" in the German version

A secured connection to a mail server is not only established to protect the e-mail messages that are exchanged with the server, as they need higher level protection on their way through other mail servers, but to ensure that the authentication data is secured.

In the standard POP3 protocol the user's name and password are in clear text, this does not allow secure use.

Pocket Outlook does not use the `STARTTLS` command, which is sent by the client to initiate an SSL/TLS secured connection with the mail server. No SSL/TLS secured connection can be used to protect user name and passwords of POP or authenticated SMTP.

Another problem is that standard SMTP does not require a password when sending e-mails, this opens up the mail server to relay e-mails for everyone, which is often abused by spammers to send their mass mailings. Hence authenticated SMTP was introduced, requiring the user to authenticate before using SMTP commands. Normally the same username and passwords as for POP3 are used, but even though they are not transmitted in clear, but authenticated SMTP in general does not require them to be encrypted.

Pocket Outlook allows using authenticated SMTP; it needs to be enabled in the advanced setting dialog (see Figure 46).

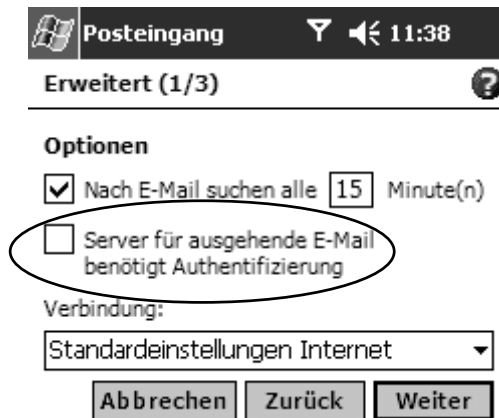


Figure 46: Advanced E-mail settings allow to enable authenticated SMTP

If enabled Pocket Outlook sends the `AUTH` command (see also [RFC2554]) and then uses SASL packets to authenticate (see [RFC2222]). In the following is a short extract from a sniffed protocol facilitating authenticated SMTP to send an e-mail:

```

Server:220 - mrelayng.kundenserver.de ESMTP Mon, 05 May 2003 11:55:08 +0200
Client:EHLO Inbox
S: 250 - mrelayng.kundenserver.de Hello Inbox [217.87.81.121]
  250 - SIZE 20971520
  250 - PIPELINING
  250 - AUTH=PLAIN LOGIN ← This server only accepts plain text logins.
  250 - AUTH PLAIN LOGIN   No security !
  250 - STARTTLS
  250 - HELP               Authentication begins:
C: AUTH                  ← The data is just Base64 encoded, not encrypted:
C: LOGIN
S: 334 - VXN1cm5hbWU6    334 - Username:
C: rZtijDFdgoI95DjjFGjwE= plain_text_name
S: 334 - UGFzc3dvcmQ6    334 - Password:
C: f4HgOP0N1tZr         plain_text_password
S: 235 - Authentication succeeded
C: MAIL FROM:
C: HENRICH@pr-poehls.com
  . . .
    
```

Trying to send mail over another mail server offering more security, by allowing for more than just a Base64 plain text LOGIN showed that Pocket Outlook still chooses to use plain text logins:

```
Server:220 - loft.local ESMTP MDAemon 6.8.0b;Tue, 06 May 2003 11:40:03 +0200
Client:EHLO Inbox
S: 250
S: 250 - ETRN
  250 - AUTH=LOGIN
  250 - AUTH LOGIN CRAM-MD5
  250 - 8BITMIME
  250 - STARTTLS
  250 SIZE 0
C: AUTH
C: LOGIN
S: 334 - VXN1cm5hbWU6
C: pQWsbkmiqW==
S: 334 - UGFzc3dvcmQ6
C: c3dgTVMIt5
S: 235 - Authentication successful
C: MAIL FROM:
C: <henrich@loft.local>
. . .
```

This server allows for more secure ways to authenticate, using the CRAM-MD5 method, but Pocket Outlook still chooses to use the insecure plain text LOGIN.

Just Base64 encoded, not encrypted:
 334 – Username:
 plain_text_name
 334 – Password:
 plain_text_password

Apart from the standard Internet protocols SMTP, POP3 and IMAP Pocket Outlook can also get new e-mails facilitating the ActiveSync connection when the mobile device is cradled. Through ActiveSync the Pocket Outlook databases are synchronized with the databases of a desktop Outlook.

[LUTTERBECK2003] gives a more detailed look at the security offered by different, also third-party, solutions to synchronize e-mail and contact data. Although they focus on client and server components (MS Exchange server), use VPN tunnels for all connections, the work shows which level of security can be achieved for synchronization. [LUTTERBECK2003] does not analyse the Pocket PC environment under the aspects of malware contamination or malware distribution.

5.5.2 Attachments

Today's e-mails are far more than just plain text messages, they often contain attachments. Pocket Outlook of course, is able to handle those attachments. The attachments of received e-mails are stored in `\Windows\messaging\attachments` as `*.att` files.

E-mails with attachments can be identified in the e-mail overview panel by a different symbol. When the email is opened the icon and file name of the attachment will be displayed at the end of the e-mail, the full name including all extensions is shown if the space of one line permits it.

As a result the trick known from desktop malware of naming an attachment "Frog.jpg.vb", hoping that the last extension is not shown to the user will not work.

But as the space is very limited, a very long name, like "Frog.jpgvb"⁴² will get shortened by displaying "Frog.jpg . . .", so there is the possibility to hide the full name. The effects are best seen in the screen shots in Figure 47.

⁴² There are 12 spaces between Frog.jpg and the .vb extension.

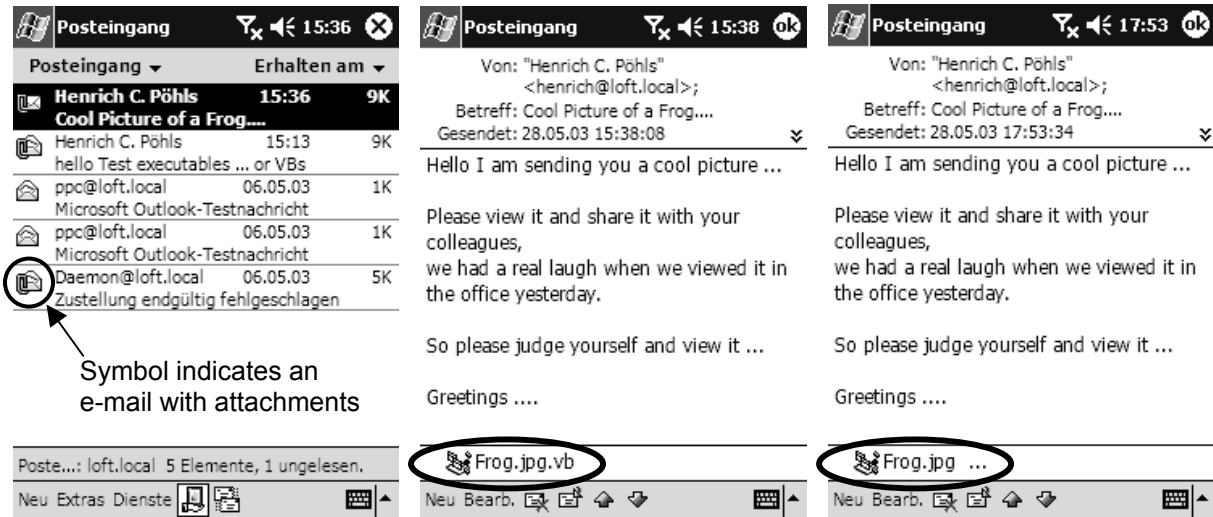


Figure 47: Pocket Outlook: Contents in Inbox (l.), E-mail with differently named attachment (m. + r.)

The space for listing the attachment's filename gets even smaller, if two or more attachments are present in an e-mail message.

If the attachment's filename is shortened like this, there luckily is a possibility for the user to determine the full name and see the file extension: Opening the context menu for the attachment by using the tap-and-hold gesture (see 4.3.7.2) on the filename. Then select "Save as ..." and in the save dialog the user can see the full name and the type of the object, including the extension.

5.5.3 HTML-Content in E-mails

Even if e-mail messages will only contain text, some are written in HTML to allow richer formatting, like font size or font color. Pocket Outlook ignores the HTML styles, as tests conducted have shown, so no embedded HTML can trigger any functions.

But Pocket Outlook helps the user with embedded hyperlinks. It recognizes HTTP:, FTP: and TEL: as prefixes and then underlines the following text, that will be interpreted as a hyperlink.

Figure 48 shows the automatically underlined embedded links.

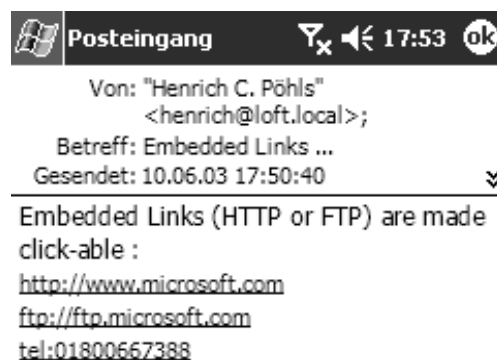


Figure 48: Embedded Links are made click-able in Pocket Outlook

5.6 eMbedded Visual Basic, eMbedded Visual C++

eMbedded Visual Basic (eVB) and eMbedded Visual C++ (eVC) are programming languages, not exactly applications. The Pocket PC 2002 SDK including the Emulator and the development tools [EVT3] including both eVB and eVC are available from Microsoft for free.

5.6.1 eMbedded VisualBasic (eVB)

An easy way to build applications for the Pocket PC is Microsoft eMbedded Visual Basic 3.0, as it is a subset of the standard Visual Basic known to programmers from the Windows desktop environment. eMbedded Visual Basic applications are not compiled, they are interpreted by an interpreter, so eMbedded Visual Basic has also some characteristics of VBScript.

The application developer uses eMbedded Visual Basic 3.0 to create an intermediate file (*.vb) from the source code, this file is then interpreted by the interpreter. Files with the ending *.vb are associated with pvbload.exe, which loads and interprets the vb-File. But the eMbedded Visual Basic interpreter needs more than the file pvbload.exe, additional files⁴³ are necessary [MSQ185223], Pocket PC 2002 includes all of these files in the ROM.

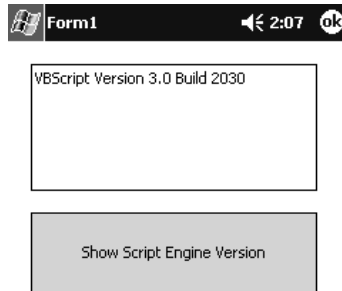


Figure 49: VBScript version shown by little eVB application (see Appendix E)

There are differences of course, between eVB and a full Visual Basic, more details on these differences can be found in [MSQ260081] and [MSQ184650]. There are also similarities between VBScript and eVB, as eVB is also interpreted and VBScript is the base of eVB [CLINICK2002].

This will make it easy to reuse or adapt code from older VBScript applications and rebuild them using eVB for the use on mobile devices. To see that they handle similar code, compare the following code for the creation of an empty file named `result.txt`, first in VBScript and then in eVB.

VisualBasic Script (works in the desktop environment):

```
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFile = objFSO.OpenTextFile("result.txt", 2, True)
objFile.Close
```

eMbedded VisualBasic source code (needs to be compiled to work on mobile devices):

```
Option Explicit
Sub Main()
    Dim objFile As File
    Set objFile = CreateObject("FileCtl.File")
    objFile.Open "result.txt", fsModeOutput
    objFile.Close
End Sub
```

More eMbedded Visual Basic source code can be found in Appendix E.

No further research has been conducted for this work on how these differences influence the security of the developed application or how they impact on malware contamination or distribution.

5.6.2 eMbedded Visual C++ (eVC)

With eMbedded Visual C++ the application developer can compile the C++ source code into Windows CE executables. As with eVB there are also differences between eVC++ and the desktop C++ language, the most significant one is the missing C++ error handling. The statements `try` and `catch`, normally used to program error handling cannot be used in eMbedded Visual C++. But as [CEGADGETS] states, error handling can be done on a lower level using the Win32 exceptions (`__try` and `__except` keywords).

⁴³ The eVB interpreter consists out of the four files: `pvbform2.dll`, `pvbhost2.dll`, `pvbload.exe`, `vbscript.dll`.

6 Risks of Malware Contamination and Distribution

Some general security issues of mobile devices have been identified during the analysis of the hardware and operating system functions of Pocket PC 2002 in chapters 3 and 4. Additionally security relevant points for the main pre-installed applications have been raised in chapter 5. This chapter will evaluate the risk of malware contamination and the risk of malware distribution for mobile devices.

First, I will make some general comments on malware (see 6.1) and establish the different requirements for different malware types (see 6.2 and 6.3). Then I will look how malware can enter mobile devices (see 6.4). When malware is on the mobile device, vulnerable functions and properties and could be exploited by malware. A number of vulnerabilities is then described in more detail (see 6.5). Finally this will all come together to an evaluation of the risks of malware contamination and malware distribution for mobile devices (see 0).

6.1 General Comments on Malware

6.1.1 No Pocket PC 2002 specific Malware known today

In August 2003, at the time of writing, no malware is known that will execute on the Pocket PC 2002 platform nor there is malware that makes explicit use of the Pocket PC 2002 environment. A search on internet sites which distribute malicious code was conducted by the malware crawler, a specialised search engine from another group at the AGN labs. The search was especially looking for malicious sites with the keywords "PALM", "Pocket PC", "PPC", and "Windows CE". The search only found some information on the already known JScript problem (see 5.4.3). The search found no hints or information on Pocket PC malware. More information on the malware crawler can be found in [FREITAG2000].

But there are reasons to assume that malware will be developed to target the Pocket PC 2002 platform in the future.

6.1.2 Reasons for Pocket PC 2002 specific Malware in the future

Malware writers might be tempted to retrieve valuable personal information, usually stored on such Personal Digital Assistants (PDAs). They might also be challenged to become the first to write malware for a new platform and to gain recognition in the virus writer community.

One point that might make Pocket PC 2002 an easy target is the consistent instruction set (see 3.2.2). The same instruction set in all the mobile devices makes it easier for malware to infect mobile devices from different vendors with the same malicious code. It is far more complicated to write a single program that runs on different platforms, and the number of mobile devices that are vulnerable to the same attack increases with their consistence. Thus the number of vulnerable targets for malware contamination increases through compatibility of different processors with a consistent instruction set. This leads to the "monoculture" that we have in most of today's desktop operating systems, that all run, up to a certain degree compatible, Microsoft operating systems with standard applications. If a general vulnerability exists in one of the standard components, this leads to a vast number of vulnerable targets. A large number of mobile devices is used today and the number will rise during the next years (see 2.2.3.1), this means that there is a growing number of vulnerable targets.

Another point that might facilitate the creation of malware on Pocket PC is that common programming languages (VisualBasic or VisualC++) and known Windows APIs can be also used on the Pocket PC 2002 platform, so code can be reused (see 5.6). Therefore it might be possible to port existing malware, having to apply only small adaptations.

Even though more and more mobile devices are available the malware writer does not even need physical access to a mobile device. The malware writer might just use the emulator [PPC2002SDK] to develop and test malware. Additionally a lot of documentation, code samples and aid is available for the Pocket PC platform, as a lot of enthusiastic users share their knowledge (see for example the complete website from Chris de Herrera [HERRERA2001]) and because Microsoft offers all the SDKs [PPC2002SDK] and development tools [EVT3] for free to promote their operating system. This enables malware writer to acquire a lot of information about the target system.

Malware has been developed for a lot of today's platforms. For example back in August 1991 Linus Torvalds created LINUX, then later distributions of LINUX came out (like Slackware in 1993), today Linux has evolved. But it took virus writers till late 1996 before the "the first known Linux virus" [FSESTAOG] named Staog was seen (see also [FSESTAOG] for a description). While the first Linux viruses have not gained that much attention, recently the Worm Slapper spread quickly through Linux systems (see [FSESLAPPER]).

Another example is Palm OS. Although there have been no worms for Palm OS, there has been other malware, for example the virus Palm/Phage and the Trojan horse Palm/Liberty [FSELIBERTY].

This shows that even if none of the incentives mentioned will lead to malware being in-the-wild (ITW) by tomorrow, it is only a matter of time until there will be malware for the Pocket PC platform as well.

6.2 Requirements for Malware Distribution

With no Pocket PC malware in existence today, malware distribution is at the moment a greater threat than malware contamination. In a Microsoft Whitepaper on Pocket PC security it is listed as "the biggest threat that mobile devices pose today regarding viruses is passing them into a corporate network." [DEDO2002]. This chapter will look at the threat in more detail.

Pocket PC 2002 devices are used in a MS Windows environment, where especially document embedded virus are quite common, wildlist.org reports 72 macro viruses as being in-the-wild in November 2002 [WILDLISTNOV2002], and documents are exchanged and will find their way onto the mobile device. This can take place with documents attached to e-mail messages, copied from and pasted onto network file servers or with documents opened and saved to and from storage cards.

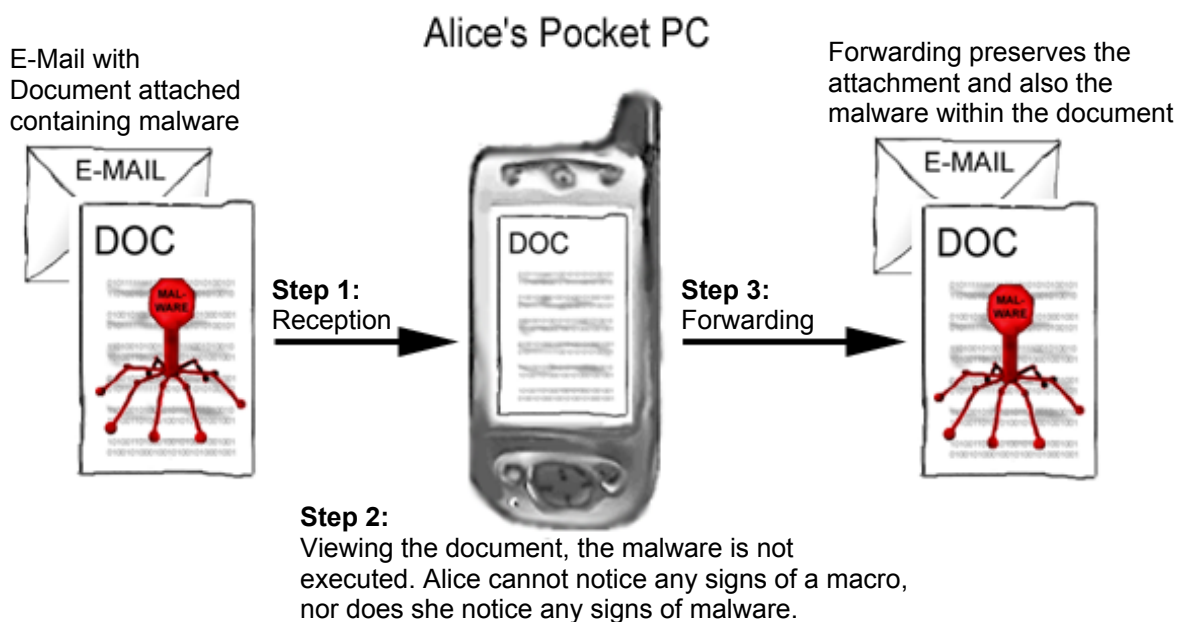


Figure 50: Malware distribution through an e-mail attachment

The example in Figure 50 shows malware distribution through e-mail: *When using Pocket PC 2002 applications like Pocket Word users can view malware infected documents without the execution of the malware. But users will also do not get any hints that some "functionality" the document has to offer has not been made available. Even non malicious macros are not noticeable. Users might therefore think the document is safe, just because nothing happened in the Pocket PC 2002 environment. And they might send the document to a desktop machine, where the document's embedded malicious content can be harmful.*

Of course, not only documents can be distributed in such a way, and not only the exchange via e-mail could be used. And also important: Also Pocket PC 2002 specific malware could be distributed.

During the process of malware distribution no malware is executed. First malware needs to be brought onto the mobile device and stored there in some object, usually a file. Then this malware must not be recognized as such, and finally transferred to another system. All forms of malware can be saved in some form on the mobile device and they can then be distributed. This means that the type of malware is unimportant for malware distribution.

The requirements for malware distribution to take place on a mobile device are then:

- Entry
- No detection
- Transfer

If all of the above requirements can be fulfilled, then distribution of any type of malware can take place.

6.3 Requirements for Malware Contamination

The threat of malware contamination (see 2.4 for definitions) is analysed here in more detail. Malware contamination requires that the malware actually executes and then contaminates the mobile device. Four different types of malware are differentiated in this work and each malware type has specific requirements to carry out its malicious offence. One might argue that all malware types of course need to be executed for malware contamination, but looking at it at a more detailed level the execution requirements can be different for different malware types (compare for example Hostile applets and Trojan horses).

First, the malware has to enter the mobile device, then as a second step it has to execute. For this work, I will subdivide the second step further into invocation and execution. When the malware is running it calls different operating system functions to achieve its malicious task. I will distinguish the basic functionalities used for: Writing to files, using network functions, modifying registry entries, and trying to hide their existence or true nature.

So additionally to the requirement of entering the mobile device, which is discussed in chapter 6.4, I will distinguish the following requirements for malware contamination:

- Invocation
- Execution
- File Write Access
- Network Access
- Registry Modification
- Hiding

They are either required for a certain malware type or for all malware types. They might not cover all the specialised functions malware might use, but they mirror basic functionality needed by certain malware types to contaminate, as a comparison with desktop based malware indicates. They allow this work to check if the mobile device fulfils at least this set of basic functionality required for malware contamination, to assess the risk of malware contamination later in chapter 6.6. For each malware type, the requirements are listed and explained in more detail. Later in the chapter 6.5 the Pocket PC 2002 functions and properties that malware could use to satisfy its requirements are explained in more detail.

Functionality that a certain type of malware could optionally use is listed under the options of this malware type. Again, it shall be noted, that malware can use other more specialised functionality not listed in this work.

6.3.1 Virus Requirements: Invocation, Execution, File Write Access

6.3.1.1 Virus Requirement: Invocation

As a first step, the virus must find a way to be initially invoked. This invocation can be either automatically or with the help of the user.

6.3.1.2 Virus Requirement: Execution

A contamination with a virus is only possible if the virus code can then be executed on the mobile device. An executable or interpretable virus needs to be specifically targeted at the mobile device's instruction set. If the virus however is a script- or a macro-virus, then it needs to be written in a script- or macro- language and use only functions that are understood on the mobile device.

6.3.1.3 Virus Requirement: File Write Access (Infection)

In order to replicate the virus needs the possibility to acquire write access to infect other objects in RAM or on inserted removable storage media. This would then be the contamination of the mobile device.

6.3.1.4 Virus Options: Registry Modification, Hiding

A virus also tries to ensure that it is started after a system is reset. This is normally achieved by adding or modifying registry settings (especially auto-run entries). Viruses also like to stay covertly in memory to infect as much objects as possible, and to avoid an early detection by the user.

For example: *The virus W32/Elkern.cav [NAIVILELKERN] uses a registry key, which will start the virus each time windows is started. W32/Elkern.cav [NAIVILELKERN] marks the file containing the virus as hidden and contains no icon, to hide itself.*

6.3.2 Worm Requirements: Invocation, Execution, Network Access

6.3.2.1 Worm Requirement: Invocation

As with the virus, worms also need to get initially invoked, which can be done automatically or through a user's action.

6.3.2.2 Worm Requirement: Execution

Then it needs to get executed on the mobile device. If it is an executable file, the worm must be specifically targeted at the mobile device's instruction set. Worms can also be written in macro- or script-languages, then they need to facilitate only macro- or script-languages running on mobile devices.

For example: *VBS/VBSWG.gen@MM [NAIVILVBSWG], also known as "Anna Kurnikova Virus", is a worm written in VisualBasic Script.*

6.3.2.3 Worm Requirement: Network Access (Propagation)

In order to propagate, the worm needs to have access to the network, to send copies of himself to other systems. Worms can use any form of network access to reach out for other systems, for example they might use e-mails, chat-messages, network shares, or remote server services to name just a few. This propagation will by definition contaminate the mobile device.

Some examples: *E-mail is very often used for spreading (examples are : VBS/VBSWG.gen@MM [NAIVILVBSWG] or "I-love-U"), these worms are mass mailers (@MM⁴⁴) and are sending a lot of e-mails. The worm CodeRed used a vulnerability in MS IIS web servers and spread from one IIS web server to the next.*

6.3.2.4 Worm Options: File Write Access, Registry Modification, Hiding

Worms like viruses often drop copies of their malicious code on the contaminated system and modify the registry to get re-executed and to continue their propagation. So write access to files and the registry is often used in the desktop environment. Of course, they also try to hide their existence, by using their own SMTP engine, or by disguising their name.

For example: *VBS/VBSWG.gen@MM stores a copy of the worm in the file AnnaKurnikova.jpg.vbs and stores information in the registry [NAIVILVBSWG]. This also shows that the worm tries to hide its extension, by using a double extension (" .jpg .vbs").*

⁴⁴ according to CARO naming convention "@MM" is added to the name for mass mailing malware

6.3.3 Trojan Horse Requirements: Invocation, Execution, Network Access (Online) / File Write Access (Offline), Hiding

As defined in [Brunnstein1999] the actual process of “trojanization” is done outside the user’s environment.

A lot of software for the Pocket PC 2002 environment (from freeware, over shareware to commercial applications) exists, to this software “Trojanic functions” could be added by a bind process.

In the Windows desktop environment a number of Trojan horses exist. The Trojanic functions are sometimes configurable by additional programs, this could be seen as a Trojanization toolkit, as they create “new” Trojan horses. No such Trojanization toolkits are available for the Pocket PC 2002 environment today, but they are not needed in order to generally create Pocket PC 2002 executable Trojan horses.

For example: *Back Orifice [NAIVILBO]* or *Sub7*, both *Backdoor Trojan horses*, allow attackers to configure the server component, which is then the malicious program sent to the attacked system, offering malicious backdoor functionality, if installed.

Network access is not generally a requirement for all Trojan horses, but a lot of the existing Trojan horses in the desktop environment fall into the category of online Trojan horses (see 2.4.3 for definition). Online Trojan horses of course need access to a network.

Trojan horses that fall into the category of offline Trojan horses (see 2.4.3 for definition) require access to the local device’s file system to do harm.

6.3.3.1 Trojan Horse Requirement: Invocation

After the Trojan horse has found its way onto the mobile device it needs to get invoked as all malware. Invocation of Trojan horses is often done by tricking the user into invoking it like a regular application. For the trick to work, the Trojanic functions might be integrated into a harmless software, which will give the user the expected functionality, while hiding the execution of the malicious functions. Or the Trojan horse uses other hiding functions to disguise their true nature and get invoked by the user (giving a false name, looking like harmless known applications, or pretending to be an update/patch).

6.3.3.2 Trojan Horse Requirement: Execution

An execution is only possible if the Trojan horse is written in code that is executable on the mobile device. They are often executables, but also Trojan horses written in macro- or script-languages are possible (for example VBS/PWStroy [NAIVILVPWSTROY] is written in VisualBasic Script). An executable Trojan horse on a mobile device needs to be specifically targeted at the instruction set and a Trojan horse written in a macro- or script language must only use languages supported by the mobile device.

6.3.3.3 Online Trojan Horse Requirement: Network Access

Backdoor or Password stealing Trojan horses facilitate network connections to either report to the attacker or to let the attacker that spread the Trojan horse connect.

For example: *Trojan horses can report their findings (passwords or keystroke-logs) by e-mail to the malicious user. In the case that the Trojan is used to install a backdoor on the system, a network connection is used to inform the malicious user, when the “trojanized” system is online, and to allow the malicious user to remotely control the system. Back Orifice (see [NAIVILBO] for details) is such a Backdoor Trojan horse.*

6.3.3.4 Offline Trojan Horse Requirement: File Write Access

Offline Trojan horses do not require a network connection (see 2.4.3 again for a definition), all of their actions are predefined. They often maliciously abuse write access to the file system, and attempt to delete valuable files from the mobile device.

For example: *The Palm OS Trojan horse Liberty (see [FSELIBERTY] for details) simply deletes all applications from the Palm device.*

6.3.3.5 Trojan Horse Requirement: Hiding

Initially all Trojans hide their true nature, so that the user initially invokes them. But especially Trojan horses that provide remote access, spy on the user or that wait some time after execution before they carry out malicious tasks (also known as “time-bombs”) shall for a long time run undetected on the infected system. So they need to hide their presence while they are running, waiting to get active or spying on the user.

For example: *Back Orifice* (see [NAIVILBO] for details) hides its task from the list of tasks and the executable file’s name is just a space (“ .exe”) to hide in directory listings.

6.3.3.6 Trojan Horse Options: File Write Access, Registry Modification

Also online Trojan horses might access files on the system, either to retrieve/steal user data or to store their own log files and information.

All Trojan horses also would like to stay on the system and get re-invoked, so they might modify registry keys or system files to keep themselves running even after system re-starts.

6.3.4 Hostile Applet Requirements: Invocation, Execution

6.3.4.1 Hostile Applet Requirement: Invocation

The hostile applet is downloaded with the web page in which the applet is embedded. The user must somehow request the malicious page to be loaded inside the browser, for example by following an Internet link. When the browser downloads the web page it will also download the hostile applet’s malicious code. Once the code is successfully downloaded the hostile applet is invoked by the browser, this requires that the browser understands the directions embedded in the web page and the language in which the hostile applet is written.

6.3.4.2 Hostile Applet Requirement: Execution

All a hostile applets need special capabilities of the internet browser application to get executed. Java Applets for example need a virtual machine.

6.3.4.3 Hostile Applet Options: File Write Access, Network Access, Registry Modification

As hostile applets are ran in the internet browser environment they often require the presence of a network connection to report back, but not in general.

Some hostile applets also just modify files on the system or they change browser settings, for example add malicious sites to the favourites menu or change the start and search pages of the Microsoft Internet Explorer. In these cases the hostile applet needs access to the files or the registry.

6.4 Points of Entry for Malware on Pocket PC 2002

There are different ways how malware, in general, can find its way on a mobile device.

The different points of entry, distinguished for later analysis in this work, are:

- Malicious executable or interpretable files accessed through the file system
- Malicious content embedded in documents
- Malicious content on Internet web sites
- Malicious content embedded in e-mails

For the first entry point, the malware is stored in an executable or interpretable file in the file system and can also be executed or interpreted directly by the operating system. This execution or interpretation can be triggered by various factors. Of course, it can also be executed or interpreted when the user taps on it.

For the last three entry points, applications are used to receive and execute or interpret the malicious content. This work will assume that the default programs already integrated into the Pocket PC 2002 platform are used to open an execute or interpret the malicious content.

6.4.1 Entry Point: File System

Files and directories can be browsed with File Explorer, or a third-party file management application. This way executable files can be executed by the user, even if they are not grouped into the start menu. Interpretable files are by default associated with the interpreter. Using for example File Explorer (see 5.1) the user can invoke executable or interpretable files by tapping on them with the stylus.

But the operating system can also invoke executable or interpretable files stored in the file system, for example on system restarts or on certain other events like the insertion of an external storage medium.

The malicious executable or interpretable files can be accessed or transferred on the mobile device in different ways:

- removable storage media,
- transfers from a network or
- transfers from the home system via ActiveSync.

6.4.1.1 File Access to Executable or Interpretable Files on Removable Storage Media

I think that the supported removable storage media, is used as the floppy disks were used in the early days of desktop computing to exchange data and applications between users and mobile devices. What used to be 3.5-inch floppy disks, holding just 1.44 Mbytes for the desktop machines, are SD- or CF-cards for mobile devices. They can carry large amounts of data, for example today's CompactFlash cards already provide 4 GByte on a single card [SANDISK]. Nearly all mobile devices allow some form of removable storage media to be inserted (see Appendix B for information on different mobile devices).

Applications, and also malware, can be executed or interpreted directly from the removable storage media as the operating system provides seamless integration, so that the access to an object from a storage card is the same as the access to an object in RAM (see 4.3.3.2.4 or 4.3.5 for details). This means that malware must not go through some secured application install process, taking place on for example on the malware protected desktop.

Additionally the operating system might invoke an auto-run executable automatically, once the storage medium is inserted (see 4.3.5.2 and 6.5.4 for further details).

6.4.1.2 File Access over Networks

If the mobile device is in a connected state it might have access to shared files and folders on other systems over the network. However executable or interpretable files cannot be directly invoked over a network using file management applications; the file either needs to be copied locally to the mobile device or a shortcut to the remote file needs to be created (see 4.4.6).

It should be noted, that the files and folders located on the mobile device itself, cannot be accessed over the network without installing additional server type applications (for example FTP or HTTP servers). The concept of network shares, known from Windows desktop systems, is not known to Pocket PC 2002. Hence for files to enter the mobile device without additionally installed software over the network, the transfer needs to be initially requested by the mobile device.

6.4.1.3 File Transfer to the local File System over ActiveSync

Additionally malicious executable or interpretable files can be transferred from the desktop by ActiveSync, if the device is in cradled state. They can be automatically copied from and to a desktop folder into the mobile device's RAM if the user enabled File Synchronization through ActiveSync.

ActiveSync does not only synchronize files, but it also allows remote controlling the mobile device from the desktop (see 4.4.13.4 for details on RAPI). As a result also any desktop based malware can transfer malicious files or maliciously modify files on the mobile device using RAPI functions. These RAPI functions are designed to work in one direction only: from the desktop to the mobile device, and could be seen as special remote procedure calls (RPCs).

6.4.2 Entry Point: Documents

Documents can carry embedded malware, especially macro based malware. This malware is executed by the application that opened the document and interprets and executes the embedded malicious code.

Documents can be opened in two ways:

- Through a file management application (like File Explorer) as their extension is associated with an application, or
- through the open file dialog of the application that is used to work with the document type

For the Pocket PC environment Word and Excel documents are the most common ones as they are widely used also in the Windows environment of desktop systems. The common desktop file extensions (like *.doc, *.exe) are associated with the Pocket PC versions of the known Office products Pocket Word and Pocket Excel (see 5.2 or 5.3 for more information about Pocket Word and Excel).

Pocket Word and Pocket Excel do not support any macros, so there is no chance that macro malware can contaminate a mobile device through macros in documents.

6.4.2.1 Documents on Removable Storage Media

Opening documents via a file management application from removable storage media is not a problem and works with a single tap. However the open dialog in Pocket Word and Pocket Excel only display documents that are located either in the \My Documents folder of the mobile device's RAM or in a \My Documents folder on the storage medium.

6.4.2.2 Documents on Local Area Networks (LANs)

Also documents are affected by the restriction that files cannot be opened directly from the network (see 4.4.6). Thus for opening a document over the network a shortcut needs to be created for file management applications. But this does not work for the open dialogs of Pocket Word and Pocket Excel, as they do not show shortcuts. As a result only documents residing locally can be opened through the applications' open dialogs.

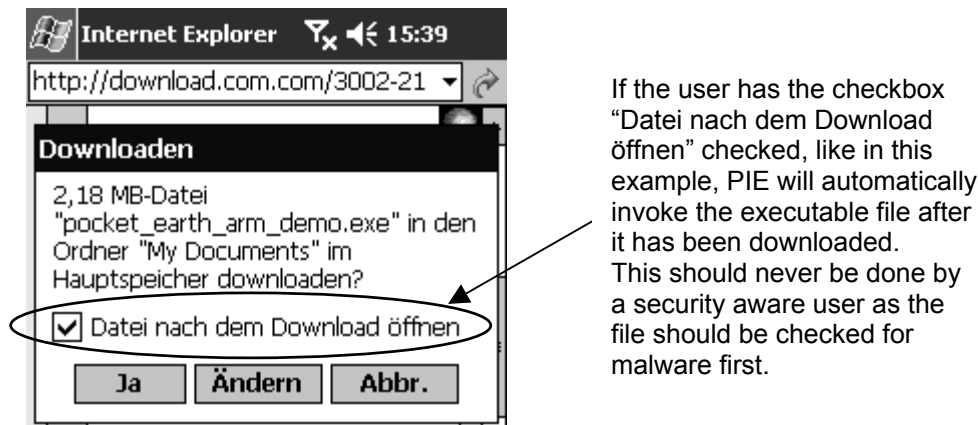
6.4.3 Entry Point: Internet Web Pages

Mobile devices now have the connection bandwidth and the screen size to show standard HTML, CHTML and WAP based web pages rather than specialised web pages only (like some mobile phones only allow the access of WAP content). As a result mobile device users can and probably will access nearly all the internet content that is available for desktop users.

For the Pocket PC environment Pocket Internet Explorer (see 5.4) is used for web browsing and allows displaying HTML and XML contents from the network. Also locally stored files with the *.html extension are associated with Pocket Internet Explorer and are opened and displayed once the user clicks on them.

The internet content can contain active embedded content (ActiveX, Flash, Scripts etc.), this content will also reach the browser of a mobile device. The content can be downloaded from the Internet with the browser, when the mobile device is in a connected state. Additionally content can be opened from the local file system even if the mobile device is not connected. If such embedded content is malicious it is referred to as a hostile applet (see 2.4.4 for a definition).

The browser usually executes active content, without additional user interaction, once it is completely downloaded. But not only embedded content can be downloaded through internet Web Pages, also executable files or documents can be downloaded. Additionally they also get executed or opened once they are completely downloaded by default (see Figure 51). The dialog to execute or open the file is not displayed when the user taps on links or enters the URL to files with for example the extensions vb or cab, but when the user taps on files with the extension doc or exe the dialog appears. No other extensions have been tested, as this shows that only executable files (*.exe) can be downloaded and executed this way.



If the user has the checkbox "Datei nach dem Download öffnen" checked, like in this example, PIE will automatically invoke the executable file after it has been downloaded. This should never be done by a security aware user as the file should be checked for malware first.

Figure 51: Downloading, saving and by default also executing files with Pocket Internet Explorer

6.4.4 Entry Point: E-mails

E-mails are normally just plain ASCII text, but they can contain attachments. Sometimes the whole body of an e-mail could be seen as an attachment, if it is not plain ASCII text, but a HTML formatted text.

E-mail is one of the functions that mobile devices are often used for. Specialised e-mail delivering PDAs, such as the Blackberry [BLACKBERRYWEB], are showing that e-mail support is a key function to mobile devices, as e-mail communication is important especially to corporate users.

As Pocket PC 2002 uses Pocket Outlook (see 5.5) to allow rich⁴⁵ e-mail messaging it is possible to receive malware as an attachment. If the user taps on an e-mail attachment it is opened directly if an executable was attached or with the associated application if it is a document or an interpretable file. The user can also save attachments received by email to the mobile device's storage, to later use it from another application.

6.5 Pocket PC Functions and Properties Exploitable by Malware

To carry out its malicious task malware needs to perform a couple of steps:

1. Malware must first find its way onto the mobile device using any of the entry points explained in the previous chapter 6.4.
2. Then it needs to get invoked, because from just being accessible by the mobile device the malware is not executed. There are functions that will automatically invoke malware, like an auto-run from removable storage media (see 6.5.4) or the malware needs to disguise itself and trick the user to invoke it, for example by using two extensions hoping that the last one is hidden in the user's file management application (see 6.5.12).
3. Then, when the malware is running it can facilitate other functions to do malicious tasks like infecting other objects or propagate through network connections.

This chapter will provide an overview of the main functions and properties of the Pocket PC 2002 platform that could be used by malware to achieve its malicious task. In risk terminology these functions and properties are the vulnerabilities. The threat of malware contamination could exploit these vulnerabilities.

Additionally to listing possible Pocket PC vulnerabilities, I will also describe how each vulnerability could be exploited by future malware.

This exploit description already contains vital details, but does not provide any source code or any further ready-to-use exploits, so it cannot be abused as an instruction on how to write Pocket PC malware.

⁴⁵ "rich" in the positive meaning of e-mail attachments containing more than just plain text, or "rich" in the negative meaning of malware rich.

6.5.1 Operating System Vulnerability: Autostart Functionality

Like other Windows operating systems Windows CE 3.0 and so also Pocket PC 2002 mobile devices offer the option to automatically start programs and applications when the operating system starts. “Start” in this case means that the complete operating system is started from the scratch, for example after a reset. Pressing the power button just wakes the system from sleep, it does not start the operating system.

There are two ways to enable the autostart functionality in the Operating System:

1. Autostart folder `\Windows\AutoStart`:
All programs located in the autostart folder are executed once the operating system has successfully started.
2. Registry key `KEY_LOCAL_MACHINE\Init`:
Putting an entry for an application here has the same effect as the `AutoStart` folder, but is not so transparent to users. [CEGADGETS] describes in more details how to add an application to the autostart registry key.

Exploiting: Autostart Functionality

Using this would allow malware to get automatically executed every time the device is restarted, so that the malware will stay in control of the mobile device, even after system restarts. Especially the second function is not easily observable for end-users as it does require additional third-party software to look at registry keys.

6.5.2 Operating System Vulnerability: No Registry Access Protection

The Pocket PC operating system does not restrict access to the mobile device’s registry as the trust model (see 4.6.6) is not implemented. The registry stores all the configuration information of the operating systems and Microsoft applications, and a lot of third-party applications have followed Microsoft storing their configuration in the registry. If the registry is corrupted or the mobile device is completely reset, default values from ROM are loaded into the RAM.

Exploiting: No Registry Access Protection

Malware could use the registry to reconfigure the mobile device itself or applications, retrieve vital information, or just corrupt the registry causing reboots. Malware can also store its own information in the registry, as this is a place that is not easily observed by the user and access to the registry requires additional third-party software to be installed.

6.5.3 Operating System Vulnerability: No Restriction on Application Execution

There are no security layers, allowing to distinguish which applications can do what, or even which applications are allowed to run at all. The Windows CE 3.0 trust-model (see 4.6.6) is only implemented in Smartphones. The undocumented policy restriction (see 4.6.7) only cares for the transfer on the device. As a result all applications can be executed on the Pocket PC, if their code runs on the processor’s instruction set and uses Windows CE 3.0 functions.

Additionally this holds true for the applications written in eMbedded Visual Basic (see 5.6) as they are also always interpreted, until a fatal error causes the interpreter to stop further interpretation of the code. The needed interpreter is installed in ROM and the interpretable code’s extension is associated with the interpreter.

Exploiting: No Restriction on Application Execution

Any application can execute, also malicious applications. Users cannot limit the execution to known malware-free applications only.

6.5.4 File System Vulnerability: Auto-Run from Removable Storage Media

The automatic execution of `autorun.exe` from removable storage media on insertion (see 4.3.5.2) allows also malicious code to automatically execute when a storage card is transferred to a mobile device. On some mobile devices it might be possible to disable the by default enabled auto-run function through a GUI. The following screen shot was taken from the ASUS A600 manual [A600MANUAL]:

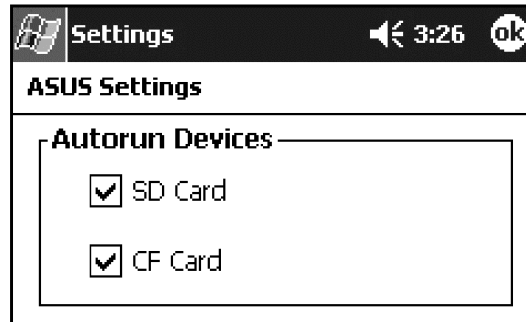


Figure 52: A600 Settings allow to disable Auto-run from removable storage media [A600MANUAL]

This shows, that on the ASUS MyPal A600 (see Appendix B for details) the default auto-run function can be disabled through the GUI, also on iPAQs this menu reachable from the settings menu is available [DEDO2002]. On the test device used for the tests an T-Mobile MDA such GUI functionality did not exist in the standard configuration, and so the MDA is an example that disabling the auto-run functionality through the GUI is not a general option for all Pocket PC users.

No registry setting is documented that will disable the automatic execution, but an undocumented registry setting⁴⁶ exists (see 4.6.7). But to set such a registry setting, third-party tools would be needed to set it.

Exploiting: Auto-Run from Removable Storage Media

If the auto-run function cannot be disabled or the user did not take the required actions, malware can exploit it. The malware could initially create or modify the `autorun.exe` of a removable storage medium. This allows the malware to get automatically invoked on the mobile device when the contaminated storage medium is inserted and the malicious `autorun.exe` is executed. The malware can then try to stay on the just entered mobile device even after the storage medium is removed, for example by copying itself onto the mobile device, before it is deleted by operating system. In the case of a virus, this would be the first replication on the mobile device.

If you compare removable storage mediums with the desktop floppy disks, the auto-run functionality makes virus spreading through removable storage media easier than spreading old boot viruses through floppy disk. Infected floppy disks needed to be left in the boot floppy drive, so the next time the system would boot the malicious code would be executed. With mobile devices the malware does not need to wait for system restarts and is already invoked on insertion.

6.5.5 File System Vulnerability: No File Access Protection

There is no access control (like DAC) based on different users in the installed file systems. No third-party file system is known that offers any additional protection (see 4.3.5.2). For example the file system NTFS, which could offer access control, is not supported [MSDNCHAT2003].

As a result all files in file systems can be accessed and modified by all applications. A sufficient write protection is also not offered, there is a file attribute read-only, but this attribute can be ignored by write operations. As known from Windows desktop environments the read-only flag rather has an informational purpose. Real write protection can only be achieved using hardware's write protection features.

For example: *SD-cards offer a small switch, which allows write protecting the data on it.*

⁴⁶ HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Shell\NoAutoRun=0x01

Exploiting: No File Access Protection

Malware can access all files in RAM or ROM based file systems and even overload ROM files (see vulnerability 6.5.6). This way malware can unhindered read, modify and delete all files on the mobile device.

6.5.6 File System Vulnerability: Overloading ROM Files

By storing a file in RAM using the same name as an existing file in ROM, malware can modify ROM files on the mobile device, as files with the same path residing in RAM have precedence over a ROM based version (see 4.3.5.2).

Exploiting: Overloading ROM Files

Malware can disguise themselves this way, so when the user searches for unknown directory entries, the Malware looks like a legitimate program or system file and is not suspicious.

Overloading a system file also has a second malicious benefit: When malware overloads an often used operating system library (for example a networking DLL) it gets invoked whenever that operating system function is used. If the malware will give the calling application the same functionality as the original operating system function, the malware is transparent to the calling application and can get access to internal application data. Trojan horses sometimes use this to steal data or viruses use network libraries to attach a copy of themselves to outgoing traffic.

6.5.7 Connection Manager: Auto Establish Pre-Defined Connections

The connection manager enables all applications to establish user defined connections. An application can check if a connection is already established or acquire a new application. When new connections are established the user is only asked for confirmation before a connection is established if he has not stored the log-in password. For more details see chapter 4.4.14.

The connections of a mobile device can reach from communication with the home system through a cradle, or to inter mobile device communication over Bluetooth, to last but not least Internet access through WLAN, GSM or just using the home system as a proxy in the cradled state. All these are if configured usable through the connection manager.

Exploiting: Auto Establish Pre-Defined Connections

Malware can establish user defined outside connections (see 4.4.14) through the connection manager and facilitate these connections to propagate or show Trojanic behaviour. It can also easily find out if a connection is available by inquiring the connection manager, so malware can sit still and wait for a connection.

If the user has supplied all the necessary information and also the credentials malware can open connections without user interaction (the user can still observe the initiation of a new connection somehow). Malware could of course also directly access the networking APIs to establish and use the connection solely for its needs (for example RAS and WinSocks, see 4.4.4 and 4.4.5). Using the connection manager makes it easier and more reliable, as malware will access connections that the user uses regularly.

Another exploit could attack the centrally stored settings. Malware could manipulate the RAS parameters. For example: *Malware could change the stored telephone number for dial-up connections to an expensive service number (with 0900 or 0190 prefix).*

6.5.8 ActiveSync Vulnerability: RAPI

Once an ActiveSync connection is established desktop applications can be automatically started or notified (see 4.4.13.3). The RAPI functionality (see 4.4.13.4) can be used to remotely control the mobile device, even before it starts synchronizing.

Exploiting: RAPI

ActiveSync allows desktop based malware to get notified, or even get invoked when the malware could spread to a connected mobile device by executing RAPI functions. This would allow desktop based malware to take control of the connected mobile device.

6.5.9 Standard Application Set Vulnerability: No List of All Running Processes

With the build-in applications users cannot determine, which processes are running in the background, nor are they able to end them. The problem is that the list of running programs that a user can access through the settings does only list running applications that have an open window. Figure 9 shows such a list.

Exploiting: No List of All Running Processes

Malware often has no need to output dialogs, so window-less malware can easily hide from the user. The user must install third-party software (for example [TLISTKILL]) on the mobile device or can use remote tools like the process monitor (see Figure 11) from the development tool set [EVT3] to get the malware's process listed.

This is like the older desktop Windows versions that had no Task-Manager that would list all running processes. Additionally to the hiding, the user has no means of stopping (kill) the malware's process without additional third-party software.

6.5.10 Pocket Outlook Vulnerability: Shortened Attachment's Filename

The standard e-mail application Pocket Outlook does shorten long file names of attachments. If shortened the file's extension is not being revealed, but by the file's extension the executing process is determined. More details and pictures can be seen in chapter 5.5.2.

Exploiting: Shortened Attachment's Filename

This allows malware to trick users into believing that they open a harmless file as they cannot see the file's last extension, by using long file names or names with a lot of spaces before the final extension. It is complicated for users of Pocket Outlook to find out about the extension, if the attachment's name is shortened:

The user needs to invoke the context menu bearing the danger that the user accidentally only taps instead of tap-and-hold (see 4.3.7.2 for tap-and-hold).

A tap would be enough to invoke the malicious attachment.

6.5.11 Pocket Internet Explorer Vulnerability: Scripting

The standard Internet browsing application Pocket Internet Explorer (PIE) allows scripting. Even though support for scripting is limited (see 5.4 for more information about PIE) it might enable malware to enter the mobile device.

Exploiting: Scripting

Scripting allows malware to embed itself in harmless looking web content and so get invoked by the user. It is problematic that the security relevant setting to allow or disallow scripts cannot be adjusted using the GUI. On a default configured mobile device scripting is enabled and cannot be disabled using default programs. The registry needs to be changed using third-part programs to disable the scripting (see 5.4.6).

The functions offered by JScript 3.0 are limited (see 5.4.3). But the first bug shows that the implementation of JScript is not perfect, and it generally allows internet content to control ActiveX and other objects via scripts. No ActiveX applets can be downloaded and installed directly in the PIE (see 5.4.1). Instead an installation routine must be invoked by the user first. And in order to then get used from a script the object must be marked as safe-for scripting.

But once a third-party ActiveX applet, which is marked as safe-for-scripting, is installed it can be scripted from any script in any HTML content. This applet could already be malicious, as the user cannot use the Microsoft Authenticode to gain confidence in the origin or the author. Authenticode would allow to check the authors digital signature (see [MSAUTHENTICODE] and [VERISIGN] for more information). But even if it is not an intentionally malicious ActiveX object, a malicious script can abuse it, if it is safe-for-scripting.

An example:

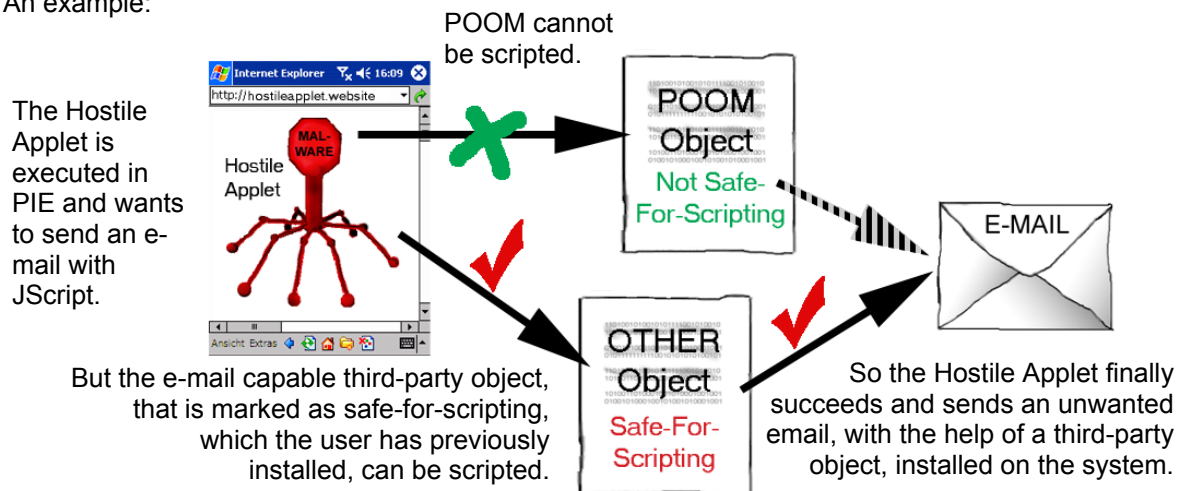


Figure 53: Example of a Hostile Applet, that misuses a safe-for-scripting ActiveX object to send e-mail

In a sample code from [FELDMAN2003] an ActiveX object is used to access Pocket Outlook in order to trigger e-mail functions from a script within a Macromedia Flash movie. This is a complicated way to send e-mail, but as noted in chapter 5.4.1 the POOM object is not marked as safe-for-scripting, and so cannot be accessed from a good script. That is why the user is encouraged to install this safe-for-scripting ActiveX object which allows sending e-mails.

If a user installs such an ActiveX object in order to use the good third-party application that it came with, he or she opens the system up for malicious scripts, that can abuse the good safe-for-scripting ActiveX object for sending e-mails.

6.5.12 File Explorer Vulnerability: Hidden Files Not Shown by Default

File Explorer, the standard file management application does by default not list files with the hidden attribute. Although the user can disable this and can view all files, some users might not facilitate this option.

Exploiting: Hidden Files Not Shown by Default

By setting the hidden attribute for files, malware might prohibit a default configured File Explorer from listing its file name. This allows malware to completely hide from at least the standard file management application. Third-party applications can also be configured to list files with the hidden attribute, but for example in the third-party file management application Resco Explorer 2003 [RESCOWEB] it is also not the default setting.

6.5.13 File Explorer Vulnerability: File Extension Hidden

The standard file management application does not list a file's extension. As a result, not all the information about files on the mobile device that is necessary to clearly identify objects or to detect suspicious objects is given. The user has no option to enable the listing of the extension within File Explorer (see 5.1. for details and a description on how to find out about the file extension without additional software).

Exploiting: File Extension Hidden

The missing file extension allows malware to trick users into believing that they open a harmless file when they see a harmless file extension (for example `Frog.jpg.exe` or `Frog.jpg.vb` is displayed as `Frog.jpg`). More details and pictures in chapter 5.1.

The last file extension is the one that determines the executing process, so malware can use this to get invoked by users or to hide its true extension.

This allows malware to hide the extension from at least the standard file management application, making it difficult for the user to find out about the true extension.

Because to find out about the true extension without additional software (described in chapter 5.1) a tap-and-hold gesture (see 4.3.7.2) is necessary, and that bears the risk of execution.

For example: *If a file is suspicious to the user, and the user tries to investigate the true extension and uses the tap-and-hold gesture. The user might accidentally just tap the file with the unknown extension, and an executable will get executed. If it truly was a malicious file as suspected, the user has now accidentally executed the malicious file.*

6.5.14 Pocket Word and Excel Vulnerability: Embedded Content Hidden

Pocket Word and Pocket Excel cannot handle all of the contents that might be stored in documents created on the desktop. As Figure 40 shows, graphical elements are not displayed correctly, but at least a place holder is displayed, indicating that the user misses something. At least the following embedded content is not displayed at all (no place holder), so they cannot be noticed by a user:

- Macros
- Embedded Excel Objects in Word
- Embedded Graphic Objects
- Embedded Executable Objects

Exploiting: Embedded Content Hidden

Malware, especially macro based, can hide in documents while being on the mobile device and wait until they are in the desktop environment to do harm. This makes it hard for user to prevent malware distribution.

6.6 Risk Evaluation

This chapter assesses the risk that exists if no additional malware defences are installed or enabled on the mobile device. The risk is evaluated for a default configured mobile device, nearly as if it came out-of-the box. Only the pre-installed applications are taken into account. Further this work will assume that the applications and the operating system behave as specified, and have no additional bugs, that allow to circumvent the documented features, as described in this work.

First the risk of malware contamination is evaluated for each malware type. The malware distribution risk is then evaluated independently of the malware's type.

Three ratings are used to roughly categorize the risk:

- high risk,
- medium risk, or
- low risk.

Rating: High Risk

The risk is categorized as high risk, if all of the following is found to be true:

- Malware can enter the mobile device easily and from different entry points.
- All the malware's requirements can be fulfilled by exploiting one or more of the vulnerabilities listed in chapter 6.5.
- Some of the malware's options can also be fulfilled.
- For malware contamination only:
Malware can get invoked and executes automatically with no user interaction.

This rating means that malware can contaminate the mobile device or be distributed through the mobile device very easily. It further can do damage to the data, application, or operating system files by exploiting also the vulnerable functions listed as optional.

Rating: Medium Risk

The risk is categorized as medium risk, if all of the following is found to be true:

- Malware can enter the mobile device.
- All the malware's requirements can be fulfilled by exploiting one or more of the vulnerabilities listed in chapter 6.5.
- For malware contamination only:
Malware can get invoked and executes with limited additional user interaction.

This means that there are chances that malware could contaminate or distribute using the mobile device's vulnerabilities for example with the user's help.

Rating: Low Risk

The risk is categorized as low risk, if all of the following is found to be true:

- Malware can enter the mobile device, but needs to be transferred or installed onto the mobile device with help of desktop programs or user interaction.
- Malware's requirements are not fully met, as there are limiting factors in the vulnerabilities listed in chapter 6.5.
- For malware contamination only:
Malware can only be invoked and executes under certain circumstances, that require extensive user interaction.

This means that only under very special circumstances, for example with excessive help of the user, malware could contaminate or be distributed.

6.6.1 Risk of Malware Contamination

Again, I will treat each of the four categories of malware separately, as the risk of malware contamination depends on the different prerequisites of different malware types, as laid out in 6.3

Three tables will show the entry points of that malware, how the malware's requirements can be fulfilled, and a final table will show, which options are possible. In the table's last row, a "✓ Yes" indicates that it is entirely possible for malware to fulfil a requirement, that the option is possible or that the entry point can be used. A "× No" on the other hand indicates that it is not possible for malware to fulfil its requirements, the option is not possible nor is this entry point usable.

Between these clear-cut assessments, there are in-between assessments, while a red "Yes, but ..." indicates that the limitations are only small, a greenish "Yes, but ..." is used to indicate that the limitations are serious.

The results from the checks of each table are then compared with the rating catalogue to finally categorize the risk of malware contamination for each malware type.

6.6.1.1 Virus Contamination Risk: High

A virus cannot be a macro embedded in a document, as macros are not recognized in Pocket Word or Pocket Excel (see 5.2 and 5.3).

Entry Point:	Useable by Virus:	
File System (6.4.1)	If written in mobile device specific code, malware can either enter the mobile device as an executable file or as an script.	✓ Yes
Documents (6.4.2)	Macros or objects in documents are not recognized in Pocket Word or Excel.	✗ No
Internet Web Pages (6.4.3)	Executable and script malware can be downloaded, and may be automatically invoked via PIE.	✓ Yes
E-mails (6.4.4)	Executable and script malware can be attached to an e-mail, and may be invoked when the user taps on it.	✓ Yes

Table 5: Entry Points for Viruses

This eliminates the entry over documents, apart from that, all the other entry points can be used by an executable or script virus, as Table 5 shows.

The requirements for virus contamination are: Invocation, execution and file write access.

Requirement:	Vulnerability:	Exploitation / Description:	Fulfilled:
Invocation (6.3.1.1)	Auto-Run from Removable Storage Media (6.5.4)	With a malicious <code>autorun.exe</code> , the virus gets invoked automatically, even if it has never entered the mobile device before.	✓ Yes
	User interaction: tap on malware (to disguise use 6.5.13 and 6.5.10)	The user can tap on a malicious file and the file is invoked. The true nature of the file tapped, could be hidden or disguised. Limitation: A user interaction is required.	
Re-Invocation	Autostart Functionality (6.5.1)	Once initially invoked on a mobile device, the virus can use this to get re-invoked if the mobile device is restarted.	
Execution (6.3.1.2)	No Restriction on Application Execution (6.5.3)	Malicious executable code can be written, which runs directly on the mobile device. Or it is interpreted, if it is a malicious scripts written in a scripting language.	✓ Yes
File Write Access (6.3.1.3)	No File Access Protection (6.5.5)	The virus can modify (read, write, delete) all files with no restriction.	✓ Yes
	Overloading ROM Files (6.5.6)	Also ROM files (mostly operating system files) can be modified and will stay modified until the RAM is erased.	

Table 6: All Virus Contamination Requirements are met

The Auto-Run from Removable Storage Media vulnerability allows viruses to get automatically invoked, which is the main reason why the risk of virus contamination is rated high. If the `autorun.exe` of a storage medium is the target for virus infections this will be dangerous. The automatic invocation can be disabled, however this is not always a trivial task. If for whatever reason the autostart cannot be disabled, additional safeguards need to be deployed to prevent at least an automatic start of a malicious `autorun.exe`.

Additionally to this automatic invocation viruses might disguise themselves by making use of the fact that the File Explorer hides the extension (see 6.5.13) or that the file name of an e-mail attachment is shortened (see 6.5.10). So that an unaware user might invoke the virus.

Once the virus is invoked it can do its malicious damage with no further restrictions. Write access to the complete mobile device's RAM and Storage Cards is granted.

But not only all the requirements are met, also all the functionality that was listed as optional for virus contamination can be fulfilled by functions and properties of mobile devices.

Option:	Vulnerability:	Exploitation / Description:	Possible:
Registry Modification (6.3.1.4)	No Registry Access Protection (6.5.2)	The registry can be accessed (read, write, delete) with no restriction.	✓ Yes
Hiding (6.3.1.4)	No List of All Running Processes (6.5.9)	Without additional programs, windows-less viruses can run unseen.	✓ Yes
	Overloading ROM Files (6.5.6)	Viruses might have a name like a legitimate operating system file, which means that also the total number of files might seem unchanged.	
	Hidden Files Not Shown by Default (6.5.12)	The File Explorer is configured by default not to list any file with the attribute hidden, viruses could use this file attribute to deliberately be hidden from the user	
	File Extension Hidden (6.5.13)	File Explorer will not list the last extension, which viruses could use to disguise itself, by using a faked doubled extension.	
	Shortened Attachment's Filename (6.5.10)	When arriving in an e-mail the malware can hide the full name, by making use of the limited screen estate used to display the filename.	

Table 7: All Virus Contamination Options are possible

So to sum up, a virus could facilitate nearly all the entry points (see Table 5). All the requirements can be fulfilled automatically (see Table 6). Also all the options can be met exploiting vulnerabilities (see Table 7), which means that a virus contamination can easily be achieved. This all makes the risk of virus contamination high.

6.6.1.2 Worm Contamination Risk: Medium

A worm could only facilitate entry points that are reachable from the network, so a worm can only enter the mobile device via email or web download, as the overview in Table 8 shows.

Entry Point:	Usable by a Worm:	
File System (6.4.1)	The worm would need to infect the desktop with a desktop malware that could then use RAPI to plant a Pocket PC worm onto the mobile device. No network access to the mobile device's file system from outside is otherwise possible.	✗ No
Documents (6.4.2)	Macros or objects in documents are not recognized in Pocket Word or Excel.	✗ No
Internet Web Pages (6.4.3)	Executable or script malware can be downloaded, and may be automatically invoked via PIE.	✓ Yes
E-mails (6.4.4)	Executable or script malware can be attached to an e-mail. But must be invoked manually by the user.	✓ Yes

Table 8: Entry Points for worms

First limitation is the restricted network access to the mobile device's file system (Entry Point: File System 6.4.1): The user cannot open network shares or allow network user otherwise access the local file system of the mobile device. This makes it impossible for worms to spread from a contaminated mobile device to another via access to the victim's file system over the network.

Access to local files and folders is only possible if the user installs additional software, which might allow network access to the mobile device from other systems. One example would be installing a HTTP or FTP server on the mobile device. Just recently a bug in the beta version of a the Microsoft web server for Pocket PC has been discovered (see [BUGTRAQWEBSVR]) by which attackers could gain full remote access, this would then be an entry point. But that software is not part of the default Pocket PC 2002 platform and therefore not considered in this work.

Also it is impossible for a worm to enter the mobile device through a macro embedded in a document, as macros are not recognized in Pocket Word or Pocket Excel (see 5.2 and 5.3). This eliminates the entry over embedded documents completely.

However, a worm can still reach the mobile device from the network in a default Pocket PC 2002 configuration via two entry points: either via web download or via an e-mail. A lot of today's desktop worms use email to propagate, an example is the worm VBS/VBSWG.gen@MM [NAIVILVBSWG].

Requirement:	Vulnerability:	Exploitation / Description:	Fulfilled:
Invocation (6.3.2.1)	User interaction: tap on malware (see also 6.5.13 and 6.5.10)	The user can tap on an executable or interpretable file and the file is initially invoked. The true nature of the file tapped, could be hidden or disguised. Limitation: A user interaction is required.	Yes, but requires user interaction
	Automatic Execution after PIE Download (see 6.4.3)	If the user initiates the download of an executable worm (*.exe) from the Internet, the worm can get executed automatically after the download has finished. Limitation: The user has to initiate the download.	
Re-Invocation	Auto-Run from Removable Storage Media (6.5.4)	Once initially invoked the worm can infect the autorun.exe. The worm then is then re-invoked automatically when the storage card is inserted.	
	Autostart Functionality (6.5.1)	Once initially invoked on a mobile device, the worm can use this to get re-invoked if the mobile device is restarted.	
Execution (6.3.2.2)	No Restriction on Application Execution (6.5.3)	Malicious executable code can be written, which runs directly on the mobile device. Or it is interpreted, if it is a malicious script written in a scripting language.	✓ Yes
Network Access (6.3.2.3)	Auto Establish Pre-Defined Connections (6.5.7)	The worm can first check with the connection manager if usable outside connections exist and if not request that such outside connections are established. Limitations: <ul style="list-style-type: none"> • Connection must be fully predefined by the user to allow a connection to be established without any user interaction. • Connection must be available to the mobile device, which might not always be in a position to reach a network (cable unplugged, no signal, etc.) • Connections might have only limited bandwidth (i.e. BT, GSM, Infrared, etc.). 	Yes, but requires some user interaction and has some limitations
	ActiveSync Proxy to Internet (4.4.13.3)	The worm can get internet connectivity, if configured by the user, while the mobile device is in the cradled state. Limitations: <ul style="list-style-type: none"> • Connection is only available while cradled • Connection relies on the desktop's internet connection • Connection must be defined by the user 	
	Network connectivity of the operating system (chapter 0)	The mobile device's operating system is build with connectivity in mind, and actively supports a variety of networking functions and protocols.	

Table 9: Not all Worm Contamination Requirements are fully met

The first thing to mention is that a worm will not be automatically executed on a standard mobile device, it requires user interaction. Again it should be mentioned, that this work assumes that there are no bugs that would allow for undocumented behaviour. The only entry points reachable from a worm via the network would be to arrive in an email attachment or be downloaded from a website (see Table 8). Pocket Outlook does not support scripting, and so the user has to tap on the attachment to open it. Also if it is a link that initiates a download through PIE the user's interactions are required to first open the link and then save (and execute) the downloaded file. But desktop worms (see [NAIVILVBSWG]) have shown that spreading is still possible even with such user interaction, as they use social engineering techniques to get the user's help.

As Table 9 further shows there are limitations on the network access from a mobile device: First of all the connection settings must be pre-defined by the user in the connection manager, but for connected devices this is often the case, as other legitimate applications will build their connectivity upon these settings (for example PIE).

They might lack some authentication information, and the request to enter those for a connection establishment could make a user suspicious. But on the other hand the worm could wait with the spreading until it detects a user established connection, which can be done using connection manager functions.

For example: *A worm using mail for spreading could simply queue e-mails for later delivery.*

Another limitation is that even if a network connection exists, as the user is in reach of a wireless LAN access point, GPRS network, or has plugged in a cable, the bandwidth might be limited. A user might then notice that another than the legitimate application is consuming bandwidth.

For example: *A mass mailing worm might consume a noticeable amount of bandwidth for some time, while it propagates copies of him self to all e-mail contacts.*

But this will not generally stop the worm, it might only lead to its detection.

So even with the limitations in the network access, the mobile device in general supports networking, and at some point in time the mobile device will be connected to the internet or at least to another system:

It will be connected to a home system to synchronize or transfer data once in a while, at least at that point in time a worm will have access to an outside connection. If the home system has internet access and the proxy function (see 4.4.13.3 for more details on ActiveSync) of the home system is configured correctly the worm then has access to the Internet while the device is in a cradled state. The device from the above scenario could be seen as a rarely connected mobile device.

The contrast would be a highly connected mobile device, which for example uses wireless LAN to stay connected, and is used in a region were wireless LAN hot spots can always be reached. This mobile device will be in a connected state most of the time, which will be a favour for the speed of the worm's malicious activity.

The limitations will still allow the worm to propagate, though they make it nearly impossible for a worm to spread very fast, as there can be large time offsets, due to the fact that on a disconnected mobile device the worm cannot propagate until it is brought into a connected state again.

Option:	Vulnerability:	Exploitation/Description:	Possible:
File Write Access (6.3.2.4)	No File Access Protection (6.5.5)	The worm can read, write and delete all files with no restriction.	✓ Yes
	Overloading ROM Files (6.5.6)	Also ROM files (mostly operating system files) can be modified and will stay modified until the RAM is erased.	
Registry Modification (6.3.2.4)	No Registry Access Protection ..(6.5.2)	The registry can be accessed (read, write, delete) with no restriction.	✓ Yes
Hiding (6.3.2.4)	No List of All Running Processes (6.5.9)	Without additional programs, window-less worms can run unseen.	✓ Yes
	Overloading ROM Files (6.5.6)	Worms might have a name like a legitimate operating system file, which means that also the total number of files might seem unchanged.	
	Hidden Files Not Shown by Default (6.5.12)	The File Explorer is configured by default not to list any file with the attribute hidden, malware could use this file attribute to deliberately be hidden from the user	
	File Extension Hidden (6.5.13)	File Explorer will not list the last extension, which malware could use to disguise, by using a faked doubled extension.	
	Shortened Attachment's Filename (6.5.10)	When arriving in an e-mail the malware can hide the full name, by making use of the limited screen estate used to display the filename.	

Table 10: All Worm Contamination Options are possible

All the additional options listed for worm contamination can be fulfilled by functions and properties of mobile devices, as shown in Table 10.

So the risk of worm contamination is evaluated as a medium, due to the limitations. But even with the limitations in the network access and entry points it is still possible to spread worms and get contaminated. If users keep opening and executing attachments they receive via email without verifying beforehand that they are not malicious, it does not matter that a worm cannot automatically get invoked on a mobile device.

6.6.1.3 Trojan Horse Contamination Risk: Online: Medium / Offline: High

As the requirements are different the two sub categories of online and offline Trojan horses also get different ratings (see chapter 2.4.3 for the Trojan horse’s definition).

Macros are not recognized in Pocket Word or Pocket Excel (see 5.2 and 5.3). This eliminates the entry of a Trojan horse through an embedded document, but all the other entry points can be used by an executable or script Trojan horse, as the following table shows:

Entry Point:	Usable by a Trojan Horse:	
File System (6.4.1)	Written in mobile device specific executable code, the worm can enter the mobile device as an executable file or a script file through the file system.	✓ Yes
Documents (6.4.2)	Macros or objects in documents are not recognized in Pocket Word or Excel.	✗ No
Internet Web Pages (6.4.3)	Executable or script malware can be downloaded, and maybe automatically invoked via PIE.	✓ Yes
E-mails (6.4.4)	Executable or script malware can be attached to an e-mail.	✓ Yes

Table 11: Entry Points for Trojan Horses

The requirements for the online Trojan horses are: Invocation, execution, network access and hiding.

The following table will show that there are the network access that online Trojan horses can get from a mobile device is limited. These are the same limitations that also apply to worms, so read the previous chapter for a more detailed description of the limitations.

The limitations in the network access affect online Trojan horses in the following way: Backdoor Trojan horses allow the attacker to remotely control contaminated mobile devices. The Trojan horse can only receive remote commands if and as long as the user’s mobile device is connected to the Internet. To stay hidden the Trojan horse could be configured to detect a user established connection and signal the attacker once it is connected.

Even though this makes attacks still possible it means that an attacker cannot always control mobile devices as they might not be constantly connected. For rarely connected mobile devices this reduces the time an attacker has remote control over the mobile device so much that it might render the remote control feature useless.

Password stealing Trojan horses, secretly report findings (i.e. found passwords). They can only report those findings back when the user is connected to the Internet. Data like that is not really time critical and is normally also not transmitted all the time, but rather aggregated over some time. So the Trojan horse can wait with the transmission until the user initiates the connection and then report the findings, in order to stay hidden.

The requirements for the offline Trojan horses are: Invocation, execution, file write access and hiding. The offline Trojan horses on the other hand are not affected by any limitation and find all their requirements met .

Requirement:	Vulnerability:	Exploitation / Description:	Fulfilled:
Invocation (6.3.3.1) Re-Invocation	Auto-Run from Removable Storage Media (6.5.4)	With a malicious <code>autorun.exe</code> , the Trojan horse can even get invoked automatically, even if it has never entered the mobile device before.	✓ Yes
	User interaction: tap on malware (see also 6.5.13 and 6.5.10)	Trojan horse are created to get invoked by the user as their true functionality is hidden or disguised. They often come under a false name and use social engineering tactics to trick the user into invocation. The user can tap on any executable file and the executable file is invoked.	
	Autostart Functionality (6.5.1)	Once initially invoked on a mobile device, the Trojan horse can use this to get re-invoked if the mobile device is restarted.	
Execution (6.3.3.2)	No Restriction on Application Execution (6.5.3)	Malicious executable code can be written, which runs directly on the mobile device. Or it is interpreted, if it is a malicious script written in a scripting language.	✓ Yes
Online Trojan Horse: Network Access (6.3.3.3)	Auto Establish Pre-Defined Connections (6.5.7)	An online Trojan horse can first check with the connection manager if a usable outside connections exists. If not it can request that such an outside connections is established. Limitations: <ul style="list-style-type: none"> • Connection must be fully predefined by the user to allow a connection to be established without any user interaction. • Connection must be available to the mobile device, which might not always be in a position to reach a network (cable unplugged, no signal, etc.) • Connections might have only limited bandwidth (i.e. GSM, Infrared). 	Yes, but only partly
	ActiveSync Proxy to Internet (4.4.13.3)	The online Trojan horse can get internet connectivity, if configured by the user, while the mobile device is in the cradled state. Limitations: <ul style="list-style-type: none"> • Connection is only available while cradled • Connection relies on the desktop's internet connection • Connection must be defined by the user 	
Offline Trojan Horse: File Write Access (6.3.3.4)	No File Access Protection (6.5.5)	The offline Trojan horse can modify (read, write, delete) all files with no restrictions.	✓ Yes
	Overloading ROM Files (6.5.6)	Also ROM files (mostly operating system files) can be modified and will stay modified until the RAM is erased.	
Hiding (6.3.3.5)	No List of All Running Processes (6.5.9)	Without additional programs, windows-less Trojan horses can run unseen in the background.	✓ Yes
	Overloading ROM Files (6.5.6)	Trojan horses might have a name like a legitimate operating system file, which means that also the total number of files might seem unchanged.	
	Hidden Files Not Shown by Default (6.5.12)	The File Explorer is configured by default not to list any file with the attribute hidden, malware could use this file attribute to deliberately be hidden from the user	
	File Extension Hidden (6.5.13)	File Explorer will not list the last extension, which malware could use to disguise, by using a faked doubled extension.	
	Shortened Attachment's Filename (6.5.10)	When arriving in an e-mail the Trojan horse can hide the full name, by making use of the limited screen estate used to display the filename.	

Table 12: Not all Trojan horse Contamination Requirements are fully met

All the functionality that were listed as optional for Trojan horses, both online and offline, can be fulfilled by functions and properties of mobile devices.

Option:	Vulnerability:	Exploitation/Description:	Possible:
Online Trojan Horses:	No File Access Protection (6.5.5)	The Trojan horse can modify (read, write, delete) all files with no restriction.	✓ Yes
File Write Access (6.3.3.6)	Overloading ROM Files (6.5.6)	Also ROM files (mostly operating system files) can be modified and will stay modified until the RAM is erased.	
Registry Modification (6.3.3.6)	No Registry Access Protection ..(6.5.2)	The registry can be accessed (read, write, delete) with no restriction.	✓ Yes

Table 13: All Trojan horse Options are possible

So the risk to get contaminated by an online Trojan horse is evaluated as a medium risk, due to the limiting factor of network connectivity (see Table 12). This still allows Trojan horses to contaminate the device, but the severity of the network attack is limited as mobile devices are not always connected.

While offline Trojan horses will find all requirements fully met, and can even get automatically invoked, even though Trojan horse do not generally require that. So contamination with offline Trojan horses is rated as high risk.

6.6.1.4 Hostile Applet Contamination Risk: Low

Hostile applets are by definition downloaded and executed in the Internet browser application, but that does not necessarily mean that Internet web pages are the only entry point. Except for documents, where active content is ignored, hostile applets might come in form of locally stored HTML files, which means they can use all other entry points as well.

Entry Point:	Usable by Hostile Applet:	
File System (6.4.1)	The hostile applet could be saved into a file (for example an HTML file) and then transferred to the mobile device. It then can be opened through the file system, as *.html files are set to be opened in PIE.	✓ Yes
Documents (6.4.2)	Word or Excel documents are not opened with the internet browser.	✗ No
Internet Web Pages (6.4.3)	Web pages can contain hostile applets and web pages are downloaded via PIE.	✓ Yes
E-mails (6.4.4)	Hostile applets saved into a file can be attached to an e-mail. It can then be opened with PIE (see File System).	✓ Yes

Table 14: Entry Points for Hostile Applets

The requirements for the hostile applet are: Invocation and execution. But invocation and execution must both be carried out by the browser, which downloads such applets. Pocket Internet Explorer (PIE) is the standard internet browser on the Pocket PC 2002 and PIE does not completely fulfil these requirements.

In the following table severe limitations on invocation and on execution are shown. This means that if dangerous hostile applets find there way onto the mobile device, they are not fully automatically invoked and executed. The bug in JScript (see 5.4.3) shows that through bugs hostile applets can do annoying, unwanted things (like closing the browser application), but that was a bug, not an intentional feature of PIE.

Requirement:	Vulnerability:	Exploitation / Description:	Fulfilled:
Invocation (6.3.2.1)	Scripting (6.5.11)	PIE allows scripting and ActiveX objects. Limitations: <ul style="list-style-type: none"> • PIE does not download and install new ActiveX objects automatically, only objects that are already installed can be scripted. • Only objects marked as safe-for-scripting can be accessed by a script (most operating system objects like POOM are not scriptable). 	Yes, but very limited
Execution (6.3.2.2)	Scripting (6.5.11)	Malicious code is interpreted, if it is written in the JScript scripting language. ActiveX objects are written in executable code and are triggered by <object> tags. Limitations: <ul style="list-style-type: none"> • New ActiveX objects need to be installed outside PIE • Only JScript code can be executed directly in PIE, no JavaScript and also no Java. • Only objects marked as safe-for-scripting can be accessed by a script (most operating system objects like POOM are not scriptable). 	Yes, but very limited

Table 15: None of the Hostile Applet requirements are fully met

Hostile applets could still do annoying things even with JScript only, even without using bugs (like opening a lot of messages boxes). But without the access to a malicious ActiveX objects, the JScript language is very limited and hostile applets will not be very harmful. Those malicious ActiveX objects cannot be downloaded and installed automatically with PIE.

There might be the chance that a highly deployed third-party ActiveX objects might provide unwanted functionality to hostile applets (see Figure 53 for an example), but the installation of such an object cannot be controlled or automatically forced on the user by an hostile applet.

Only with great user assistance, which would require that the user takes the necessary steps to install the malicious ActiveX script outside the browser, a hostile applet can do really harmful things on the mobile device.

For Hostile applets the functionality listed as options must be also performed through functions and properties accessible from the browser, in this case the Pocket Internet Explorer.

Option:	Vulnerability:	Exploitation/Description:	Possible:
File Write Access (6.3.1.3)	Scripting (6.5.11)	Malicious ActiveX objects, once installed and registered on the mobile device, have full access to files. Limitation: ActiveX object must be installed and registered outside PIE, this is not done automatically.	Yes, but only with user interaction
Network Access (6.3.2.3)	Scripting (6.5.11)	The hostile applet is usually downloaded from the Internet, so a network connection is active. Malicious ActiveX objects, once installed and registered on the mobile device, can access the network. Limitation: ActiveX object must be installed and registered outside PIE, this is not done automatically.	Yes, but only with user interaction
Registry Modification (6.3.1.4)	Scripting (6.5.11)	Malicious ActiveX objects, once installed and registered on the mobile device, have full access to the registry. Limitation: ActiveX object must be installed and registered outside PIE, this is not done automatically.	Yes, but only with user interaction

Table 16: All Hostile Applet Options are only possible with additional user interactions

Through JScript, hostile applets cannot access vital system functions, as the objects are not marked as safe-for-scripting.

The hostile applet will only get access to objects like POOM (for email, or contacts), FileControl (for file manipulation) and ADOCE (for database manipulation) if it can install a malicious ActiveX object, which is marked safe-for-scripting. The whole idea of hostile applets is that they only abuse browser functions or functions that can be executed within the browser, or functions that can be downloaded through the hostile applet. So with this definition, none of the optional functions can be used by hostile applets, as they all require user operations outside the browser prior to the malicious execution.

So the risk to get contaminated by a hostile applet is evaluated as a low risk, due to the inability of Pocket Internet Explorer to download and install new ActiveX objects automatically.

6.6.2 Risk of Malware Distribution: High

The requirements for the distribution of malware, as laid out in 6.2, are: Entry, no detection and transfer.

Entry Point:	Usable by for malware distribution:	
File System (6.4.1)	All file system functions can be used to transfer malicious files to and from the mobile device.	✓ Yes
Documents (6.4.2)	Malicious macros or objects in documents are not recognized in Pocket Word or Excel, but they can be saved and distributed to desktop systems. There they can do damage.	✓ Yes
Internet Web Pages (6.4.3)	Malware can be downloaded via the Internet.	✓ Yes
E-mails (6.4.4)	Malware can be attached to an e-mail and so enter the mobile device and also used to distribute it.	✓ Yes

Table 17: All entry points can be used for malware distribution

Malware can find its way onto the mobile device through all the entry points explained in chapter 6.4 as the overview in Table 17 shows. Documents and files might be transferred to the mobile device even if the file or documents are not intended to be ever used mobile. The mobile device is then just used as mobile storage medium for desktop systems.

Also storage cards might used to carry data, like documents, can contain malware. Most Pocket PC devices allow some storage medium to be inserted and they can be used to transfer data from and to removable storage cards.

For example: *The mobile device can act as a copy station and transfer data from one removable storage medium to another. One scenario would be that a user wants to share some movies or pictures with another user that gives him his removable storage medium.*

Requirement:	Vulnerability:	Exploitation / Description:	Fulfilled:
No Detection (6.2)	Embedded Content Hidden (6.5.14)	The standard applications for handling documents will not notify the user, that embedded content exists and has just not been displayed.	✓ Yes
	Special mobile device instruction set (see 3.1.2)	The instruction set used in mobile devices is different from that of desktop systems. As a result it is desktop specific code cannot execute.	
	Hidden Files Not Shown by Default (6.5.12)	The File Explorer is configured by default not to list any file with the attribute hidden, malware could use this file attribute to hide from the user.	
	File Extension Hidden (6.5.13)	File Explorer will not list the last extension, which malware could use to disguise, by using a faked doubled extension.	
	Shortened Attachment's Filename (6.5.10)	When arriving in an e-mail, malware can hide the full name, by making use of the limited screen estate used to display the filename.	
Transfer (6.2)	Standard File System or Network functions	Using the functions from file management applications or network applications the user can transfer malware to other removable storage media or to other systems.	✓ Yes

Table 18: The requirements for malware distribution are fulfilled

Due to the restrictions of the Pocket PC applications Pocket Word and Pocket Excel malware embedded in documents is not recognized, they do not reveal macros or other embedded objects (see vulnerability 6.5.14). Additionally malware not written for the mobile device's instruction set, especially the malware written for desktop systems, is not executed and cannot be recognized as malicious.

Apart from these properties of the mobile device that will especially make it hard to detect the malware for desktop systems, there are several other properties of mobile devices that all malware could use to not get detected in order to be distributed. Hiding the file extension makes it possible to hide the true nature of the file, or with the hidden attribute even the whole malicious file can be hidden, which malware might use this to disguise itself.

For example: *The mobile device might be used to copy directories from one storage card to another, and the user did not recognize the malicious file hiding among other harmless pictures in the copied directories. This means that the malware has just been distributed.*

This shows that recognizing, especially desktop malware is not easy on mobile devices. And a lot of such desktop based malware exists. Finally the mobile device can be used to transfer the unrecognised malware to another system. All the file handling functions of the mobile device can be used for transfers. Some examples are the file management application File Explorer (see 5.1) or the e-mail application Pocket Outlook (see 5.5) that could be used to transfer any file also a malicious file.

So as all the requirements are fulfilled and especially the existing desktop malware might be distributed with the help of mobile devices, the risk of malware distribution is rated as high.

7 Malware Defence on Pocket PC 2002

So after identifying and evaluating the risks of malware for mobile devices this chapter will look at the safeguards that Pocket PC 2002 devices have to offer.

In the previous chapter the areas virus contamination (see 6.6.1.1) and malware distribution (see 6.6.2) were identified as areas of high risk. So in these areas safeguards are especially needed to reduce the risk.

A safeguard against malware is to deploy and effectively use anti-malware software. The detection of existing desktop malware by anti-malware products for the Pocket PC 2002 environment is tested in chapter 7.2. But first this chapter will start with looking at build-in safeguards of the mobile device.

7.1 Build-in Defences

This chapter will show how achieve more security on the mobile device. A build-in defence would be to disable a function that was identified as vulnerable in chapter 6.5. But there are not many functions that can be configured easily through a GUI settings dialog, most of them need changes in the registry. Although these build-in defences can have a positive impact on the risk of malware contamination they might impact the mobile device's usability.

The following listed functions are not ordered in any form. Users shall check, which defence they could enable without loosing too much usability in their environment.

7.1.1 Disable Auto-Run from Removable Storage Media Functionality

On some mobile devices the auto-run functionality can simply be disabled through a GUI dialog (see Figure 52). On others the user can modify the registry to disable it⁴⁷. More details can be found in the chapters 6.5.4 and 4.6.7.

The user should keep this functionality disabled to enhance the security, as this greatly reduces the risk that malware from a storage card is invoked and executed automatically.

Especially before unknown storage cards, or storage cards that have been exposed to other systems (whether mobile or desktop) are inserted, the automatic execution of the `autorun.exe` shall be disabled.

It should not render a non-malicious application residing on a storage card facilitating an `autorun.exe` unusable. However, it can have an impact on the ease of use, as the user now has to manually invoke the `autorun.exe` of such an application.

7.1.2 Enable Undocumented Policy Restrictions

In chapter 4.6.7 an undocumented function that allows to restrict the transfer of executable (`*.exe` or `*.cab`) files to the device is described. This function can be used to further limit the entry points, as the user will no longer be able to accidentally transfer `*.exe` or `*.cab` files to the mobile device using the standard application set.

This reduces the risk of all executable malware (`*.exe` or `*.cab`) as it blocks the entry in all the standard applications like Pocket Internet Explorer, ActiveSync or Pocket Outlook. And restoring the original extension of renamed executables is also not possible with File Explorer.

It is quite convenient to disable or enable this additional entry protection through the GUI, once enabled through a registry change, using the additional password, whenever a new application is willingly transferred and installed. It will not interfere with already transferred executables. It will not catch eVB code (`*.vrb`), but see the next defensive setting in chapter 7.1.3 to achieve that. So after enabling the policy restrictions it is not less convenient to work with the mobile device and its installed applications and so this build-in defence should be enabled.

⁴⁷ The registry key `HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Shell\NoAutoRun` must be set to `0x01`.

7.1.3 Remove Association between vb Extension and eVB Interpreter

Files with the extension `vb` are normally interpretable eVB code created for example with eMbedded Visual Basic 3.0 (see 5.6.1 for some details). By default they are associated with the eVB interpreter `pvbload.exe`. If malware written in eVB becomes a big threat or no eVB applications are used on the mobile device at all, this default association can be removed.

To remove this association the registry key `HKEY_CLASSES_ROOT\.vb` should be renamed to something like `.HKEY_CLASSES_ROOT\.vbsec`.

Then the user needs to rename all existing non-malicious eVB application's extensions also from `vb` to `vbsec`. This reduces the risk of manually invoking unchecked eVB applications, which might be malicious and might use disguising techniques to trick the user into tapping on them. It introduces the inconvenience of having to rename all the eVB applications that shall stay usable, but after that the user can be relatively sure that he can only execute the non-malicious eVB applications automatically.

7.1.4 Check "View all ..." in File Explorer

File Explorer can be configured to list also files that have the attribute hidden (see 5.1.3 for details). This will remove the vulnerability: 6.5.12 Hidden Files Not Shown by Default. This will make it harder for malware to hide from the user when the Pocket PC default file management application, File Explorer, is used.

So this can help detecting hidden malware and might reduce the malware contamination risk. Also seeing possibly malicious, hidden files an aware user might not accidentally transfer them from and to storage and so prevent malware distribution. It is a bit inconvenient, as some directories might now contain a whole bunch of files that previously have been hidden. But the user might also learn a bit about the operating system's inner working, by now being able to see the complete operating system's file structure.

7.1.5 Do Not Save Passwords in the Connection Manager

While it never is a good idea to store passwords, not storing passwords for network connections prevents the mobile device instantiating an outside connection without the user's acknowledgement. This will reduce the risk that comes from the vulnerability: 6.5.7 Auto Establish Pre-Defined Connections, because this will further limit the network access capabilities of malware. The user will be made aware of a newly initiated pre-defined connection, because the password is requested. This helps to reduce the risk of malware contamination for worms, online Trojan horses, hostile applets and other malware that need network access and might try to establish connections.

This will only work in environments where such connections are rare. Otherwise, the usability is reduced to far by requiring too much user interactions. If the mobile device however is nearly constantly connected, then this option will not provide additional awareness.

7.1.6 Enable the Power-On-Protection

The user should set a password and an appropriate locking time, as described in chapter 4.6.1. This will protect the data and applications from other users as they can only access the device if they know the PIN or the password. Even though the PIN can be guessable or brute forced it will reduce the risk that the mobile device can get remote controlled via RAPI (see 4.4.13.4) when cradled at other stations. Some people might just want to charge their Pocket PCs at a different system, and without the knowledge of the a password also the guest partnership cannot be established. Putting the mobile device in another cradle might only rarely happen, but with the password enabled it requires the user's interaction every time a connection over ActiveSync is initiated, so that the user made aware of this fact. Additionally it will also protect the access to the Pocket PC once it is lost and it should be enabled, as the user would enable the PIN on his or her mobile phone.

Additionally it adds security against the unauthorized access to the mobile device if the device is lost or stolen, this should encourage the user to enable the power-on-protection. Although it lengthens the time until a user can access his or her device, this option should be enabled for the given reasons.

7.1.7 Disable Automatic Download in Pocket Internet Explorer

Pocket Internet Explorer (PIE) by default executes and opens certain file types (see 6.4.3). This option shall be disabled and the downloaded file shall be checked for malware first.

7.1.8 Disable Pocket Internet Explorer Scripting and ActiveX

Pocket Internet Explorer (PIE) can be configured to disallow scripts and active content (see 5.4.6 for details). This will totally stop hostile applets which already were categorized as low risk (see 0 for the risk evaluation) from entering the mobile device in the first place. It will also prevent any form of scripted automatism when malware tries to enter through web pages and will so help to generally reduce the risk of malware contamination.

It will on the other hand also render any installed useful non-malicious ActiveX objects unusable, and turning of scripting might reduce the usability of non-malicious, but scripted web content.

7.2 Test of Anti-Malware Products for Pocket PC 2002

As there are still a lot of vulnerabilities that malware could exploit after the few build-in safeguards are enabled, this chapter will look at the protection offered by external third-party anti-malware software. The anti-malware products that are in the market for Pocket PC 2002 will be tested and compared.

7.2.1 Test Goals

Mobile devices shall be protected at least with the same safeguards against malware contamination and distribution as desktops and servers are today.

A holistic approach to malware defence has to avoid a weak link in the chain of protection.

If malware can be distributed through mobile devices also the desktop and server environment is at risk. So as desktop and server systems have malware protection through anti-malware software, the mobile devices shall also be protected by such software.

The test shall allow identifying the protection offered by the anti-malware products running on a mobile device in general and for each individual product. This shall be done in a comparable way, so that it can be contrasted with the protection offered by desktop or server anti-malware products.

The goal of the test is to measure the detection rate and detection quality of malware with anti-malware products running stand-alone on mobile devices⁴⁸.

7.2.2 Test Conditions

To allow the testing of an anti-malware product the product must meet some requirements, the test conditions. If the anti-malware product meets the test conditions it will be able to produce a measurable output.

The following test conditions do only slightly differ from the test conditions for normal aVTC tests [VTCWEBSITE].

Compared to the normal aVTC test conditions, it shall be noted, that some conditions are left out, as they do not apply to mobile devices in general or to the tests conducted on the mobile device for this work, as for example the test conditions for file, boot, or packed viruses.

Also some conditions have been made optional or their strictness is lowered to suit the mobile device requirements better.

Conditions that are not essential are indicated by putting the condition name (for example "AA") in square brackets (for example "[AA]"). If the condition is not essential a product can be tested even the condition is not fulfilled by the product. In some cases the strictness of a condition can only be lowered. A less strict condition is then given in square brackets and must be fulfilled instead of the stricter original condition. If any condition is not met this is listed in the detailed product description for each product in chapter 7.3.

⁴⁸ Again it shall be stressed that the mobile device used for this tests runs the Microsoft Pocket PC 2002 operating system.

Common conditions:

- [AA] Essential parameters or options under which the scanner produces optimum detection results should be available to the tester.
[Otherwise suitable default options are used.]
- AB) The scanner must perform its detection tasks within reasonable time, compared to similar products.
 - A1) The scanner must be able to create a report file.
- [A2] The full path of scanned files must be present in the report file. Long paths must not be abbreviated, e.g. by using "..." instead of several intermediate directory names. Shortening file paths is acceptable when displaying them on the screen, but not in the report file.
[At least the full path of reported files must be present in the report file, but the total number of scanned files must then be reported.]
- A3) The scanner must be able to run in "scan-only" mode. If its default mode is to disinfect automatically all viruses found, there must be an option to run it in "scan-only" (i.e. no disinfection) mode.
- A4) The scanner must be able to run unattended - and they must not stop on each infected object and request user input.
- A6) If the scanner issues an audible alarm each time when it detects a virus, there must be a way to turn the sound off. This is not necessary if the alarm is issued only once - at the end of the scanning, but the alarm should be able to stop on its own, i.e. without requiring user intervention.
- A7) The only limit of the size of the report file that the scanner creates must be the amount of free space.
- [A8] The scanner must be able to test objects on netdrives and obey the given user rights (i.e. read only, access denied).
- A9) The scanner must not move any file which it regards as infected to another drive or a specified directory.

M) Conditions for tests against macro viruses:

- M1) The scanner must be able to scan macro viruses.
- M2) The report file must contain the directory path, the file name of the suspicious or infected file.

Q) Conditions for other classes of viruses:

- In test "2000-08": testbed for script viruses (VBS, JS, mIRC) added.
Same conditions as for macro viruses apply.
- In test "2001-04": testbed for exotic viruses (OS/2, Linux, Java) added.
Same conditions as for macro viruses apply.

Additionally the following new test conditions must be met for the test on the mobile devices running MS Pocket PC 2002 operating system:

C) Conditions for test on a mobile device

- C1) The scanner must be able to work as a stand-alone product, running autonomously on the mobile device.**
- C2) The scanner must be able to perform its task (especially obeying AB) also on mobile devices of the first generation, as used in the test environment.**

These two additional conditions ensure that the tested products run stand-alone on all Pocket PC 2002 devices.

7.2.3 Test Measurements

The detection rate and the detection quality shall be measured to provide comparison criteria for different anti-malware products.

The following will give an overview of the different measurements used in the test (see also [VTCTESTPROTO2002] or [SEEDORF2002]) and how they relate to each other.

7.2.3.1 Detection Rate

The detection rate can be calculated in two ways:

- on the basis of the malware existing in the testbed, or
- on the basis of malicious samples (e.g. files) existing in the testbed.

The detection rate on a malware basis, shows which malware (i.e. viruses) can be detected out of all the malware in the testbed. It is calculated by dividing the number of reported malware by the number of all the malware in the scanned testbed.

$$\text{DetectionRate (malware basis)} = \frac{\text{Reported malware}}{\text{All malware}}$$

As there can be several malicious samples of each malware in a testbed, a second detection rate can be calculated on the basis of detected malware samples. So it is calculated as the number of reported malicious samples divided by the number of all the malicious samples in the testbed:

$$\text{DetectionRate (malicious sample basis)} = \frac{\text{Reported malicious samples}}{\text{All malicious samples}}$$

A value of 100% means maximum detection.

7.2.3.2 Detection Quality

The detection quality of anti-malware products is measured by two values:

- Unreliable Detection Rate
- Unreliable Identification Rate

As the testbeds contain multiple malicious samples for each malware (e.g. ten Word documents all infected with the same virus) the anti-malware product shall detect and report all the malicious samples as being malicious. The unreliable detection rate reaches zero percent, if all malicious samples are reported as malicious, and then the Detection Rate (both on object and on malware basis) is 100%. But it can also reach zero if no malware is reported at all.

The unreliable detection rate is calculated on a malware basis, counting the malware were not all malicious samples were reported and dividing it by the number of all malware in the testbed.

$$\text{Unreliable Detection Rate} = \frac{\text{Reported malware with not all samples detected}}{\text{All malware}}$$

A second quality criterion is how consistently the anti-malware product identifies different malicious samples containing the same malware (e.g. 10 different Word documents all infected with the same virus Virus.A) by reporting the same malware name in all malicious samples. If an anti-malware product reports different malware as found even though it is the same malware in any malicious sample (e.g. reporting two of the ten Word documents as being Virus.B, while reporting the rest correctly as Virus.A) this counts as an unreliable identification. Therefore even if an anti-malware product reaches 100% in the detection rates, it can still reach a poor 100% unreliable identification giving no indication what malware has been detected for example by just reporting "virus found" for all malicious samples. It is also calculated on a malware basis, and can also reach zero if no malware is reported at all.

$$\text{Unreliable Identification Rate} = \frac{\text{Reported malware with inconsistently identified samples}}{\text{All malware}}$$

An unreliable identification and an unreliable detection rate of 0% indicate best detection quality. But they shall be used only if the malware based detection rate is non-zero, as these measurements assess the quality of the malware detection.

A further criterion is to test if the products raise false alerts, referred to as false-positive test in regular aVTC tests. This work has not conducted tests if non malicious objects are false identified as malicious by the Pocket PC scanners, as this would have introduced further testbeds, especially crafted for mobile devices, which would exceed the time constraints of this work.

7.2.3.3 Calculating the Measurements

The anti-malware products have to generate a report file (see condition A3 in 7.2.2). This report file (or log-file) is used to calculate the measurements.

As the amount of free RAM is a very valuable resource on mobile devices all products only log the path and name of infected objects, not the path and name of all the objects scanned in their reports. This partly violates test condition A2⁴⁹ (see 7.2.2) as this states it should report scanned files, but most important of A2 is that the report file shall contain the path and filenames as this is used to identify the malicious samples that are reported by the product.

Reporting only the path and filename of objects that have been identified as malicious, but not of all the objects scanned, introduces a problem:

If any malicious samples have not been touched by the scanning process at all, this will be more difficult to detect. Detecting which samples have not been scanned is a very complicated task.

Luckily all products tested do list the total number of objects scanned in their report, allowing calculating if all samples from the test set were touched.

For calculating the measures I will assume the following:

- If the product reports the same number of objects as scanned, as there are samples in the test set, that the product has scanned them all, and only those listed in the report have been identified as malicious.
- If the product reports a lower number of objects as scanned than there are objects in the test set, I will assume that the product has missed some objects. The product will then be retested on a sub test set. This sub test set will only contain the objects that have not been reported as obviously scanned by being identified as malicious in the reports. Thereby I will reduce the number of objects that need to be scanned. This will be called a "re-test". Each product is allowed two such re-tests.

Another problem that occurred with some anti-malware products is their inability to scan only a defined portion of the mobile device, as the testbeds are transferred into a subdirectory on the mobile device. These products scan all the objects on the mobile device. To reliably calculate the number of samples scanned a pre-scan is initiated. The pre-scan shall identify the number of objects reported to be scanned with the scanner options without the samples from the testbed on the mobile device. The total number of objects scanned reported in the regular scans will then be decremented by the number of objects reported by the pre-scan.

Any such problems and others are reported in the test results in chapter 7.4, or for each product in the chapter 7.3.

7.2.4 Test Procedures

As the platform of mobile devices is relatively young and the mobile devices just gain the memory and processing requirements enabling them to perform tasks such as scanning for malware, the products will only be tested against a limited set of malware.

Malware that is circulating among systems and reaches a lot of users is a greater threat to mobile devices, than malware that is not actually spreading and has therefore not reached many systems. This circulating malware is referred to as in-the-wild (ITW).

The WildList organization [WILDLISTORG] maintains a list of viruses that have been reported to be in-the-wild, by special WildList members, so-called reporters. In order to receive the status in-the-wild two incidents, concerning this malware, must be reported by WildList reporters within a month.

⁴⁹ A2) The full path of scanned files must be present in the report file. Long paths must not be abbreviated, e.g. by using "..." instead of several intermediate directory names. Shortening file paths is acceptable when displaying them on the screen, but not in the report file.

As the processor for the mobile devices (based on ARM architecture) differs from the processor used in most desktop or server systems, the executable malware for desktops and servers can never run on mobile devices, as they are compiled to run on that one platform only. This work will further not test the detection of file malware.

This test will limit the malware used in the testbeds to test Pocket PC 2002 anti-malware products for the above reasons to in-the-wild script and macro malware.

This test is the first anti-malware product test run on mobile devices. Hopefully more anti-malware product vendors will recognize the need for malware protection on mobile devices, as malware writers will recognize the vulnerabilities in the future for sure. Then this first anti-malware product test on Pocket PC 2002 will be followed by others, also wider tests, probably not only from the aVTC.

In the following I will outline in more detail the malicious objects used to test the scanners, the so-called testbeds, and the environment in which the test will be performed (see 7.2.4.2). Finally the test steps are described in detail (see 7.2.4.3).

7.2.4.1 Testbeds

The testbeds contain malicious samples (files) from different script and macro viruses that were reported to be in-the-wild (ITW). As the compilation of the testbeds is a complicated process that also very much determines the quality of the test results the most recent testbeds were not available right away when the products from different vendors were available for testing in February 2003.

So I conducted a preliminary test using older aVTC testbeds from earlier aVTC tests. This pre-test was done in May 2003 and will be referred to as “pre-test” or “Pocket PC Scanner Test 2003-05” [PPCTEST2003-05].

For the pre-test the following testbeds were used:

- aVTC ITW testbeds from the Heureka III Test:
 - Script ITW frozen 31st January 2002 (scr_itw.021)
 - Macro ITW frozen 31st January 2002 (mac_itw.021)
 - Script ITW frozen 30th April 2002 (scr_itw.024)
 - Macro ITW frozen 30th April 2002 (mac_itw.024)
- aVTC ITW testbeds from old Test 2002-12 [VTCTEST2002-12]:
 - Script ITW frozen 31st October 2001 (itwskri.002)
 - Macro ITW frozen 31st October 2001 (itwmac.002)

Each testbed is frozen at a specific date, frozen means that the testbed will include only samples of malware before that date. For the aVTC tests the testbeds are regularly frozen about 6 weeks before the submission deadline of the anti-malware products, so that the products are not confronted with brand-new viruses. The submission deadline for the products for this test was 17th of February 2003 midnight, so the samples from the testbeds used for the pre-test are rather old.

For the actual test of anti-malware products on mobile devices, referred to as Pocket PC Scanner Test 2003-07, the following testbeds were used:

- aVTC ITW testbed for actual Test (probably named Test 2003-09):
 - Script ITW frozen 31st December 2002 (scr_itw.dec02)
 - Macro ITW frozen 31st December 2002 (mac_itw.dec02)
- Testbed containing samples of all ITW viruses from November 2001 till December 2002:
 - Script ITW Nov.2001 till Dec.2002 (scr_itw.304)
 - Macro ITW Nov.2001 till Dec.2002 (mac_itw.304)

The name in brackets is the name of the testbeds as used internally, it is used later for a shorter reference to each testbed and is also present in the scanner reports.

From the very large collection of all malware that was collected, filtered, identified and ordered by the aVTC Team until 31st of December 2002, only samples of viruses that were reported by wildlist.org to be in-the-wild in December 2002 [WILDLISTDEC02] were used for the testbed of the ITW script and macro viruses December 2002 (scr_itw.dec02 and mac_itw.dec02). So they include samples of viruses that were in-the-wild 6 weeks before the product's submission deadline.

I additionally used two testbeds (scr_itw.304 and mac_itw.304) that contain samples of viruses that were in-the-wild in the time between the testbeds from the pre-test and the December 2002 testbed. These second testbeds contain samples from all script and macro viruses that were listed to be in-the-wild by wildlist.org from November 2001 to December 2002.

So the second two testbeds (based on wildlists from Nov.2001 till Dec.2002) include also older viruses and also samples from the first two testbeds (based on the wildlist from Dec.2002 only).

A detailed list of the malicious samples contained in each testbed can be seen in Appendix C.

For a better understanding I will briefly explain the contents of the script testbed. On the right hand side of the following table I have listed the contents of the scr_itw.dec02 testbed (details in Appendix C 2.2) and on the left the script viruses as listed in the wildlist from December 2002:

Wildlist December 2002 (script entries only):	Path in Testbed scr_itw.dec02:
JS/Kak.A-m	..\scr_itw.dec02\jvs\K\Kak\A
VBS/Freelink-mm	..\scr_itw.dec02\vbs\fl\Freelink\A_MM
VBS/Haptime.A-mm	..\scr_itw.dec02\vbs\h\HAPTIME\A_MM
VBS/Haptime.D-mm	..\scr_itw.dec02\vbs\h\HAPTIME\D_M
VBS/LoveLetter.AS-mm	..\scr_itw.dec02\vbs\ll\LOVELETT\AS
VBS/LoveLetter.A-mm	..\scr_itw.dec02\vbs\ll\LOVELETT\A_MM ..\scr_itw.dec02\vbs\ll\LOVELETT\BG
VBS/LoveLetter.C-mm	..\scr_itw.dec02\vbs\ll\LOVELETT\C
VBS/Netlog.A	..\scr_itw.dec02\vbs\n\NETLOG\A
VBS/Redlof.A-m	..\scr_itw.dec02\vbs\r\REDLOF\A_M
VBS/SSIWG2.A-mm [Daira.A-mm]	..\scr_itw.dec02\vbs\d\Daira\A_MM ..\scr_itw.dec02\vbs\s\SSIWG\U ..\scr_itw.dec02\vbs\s\SSIWG2\A
VBS/Stages.A-mm	..\scr_itw.dec02\vbs\s\STAGES\A ..\scr_itw.dec02\vbs\s\STAGES\A_14559 ..\scr_itw.dec02\vbs\s\STAGES\A_2543
VBS/Tam.A-m	..\scr_itw.dec02\vbs\t\TAM\A_M
VBS/VBSWG.AQ-mm	..\scr_itw.dec02\vbs\v\VBSWG\AQ
VBS/VBSWG.J-mm	..\scr_itw.dec02\vbs\v\VBSWG\J_MM
VBS/VBSWG.K-mm	..\scr_itw.dec02\vbs\v\VBSWG\K_MM
VBS/VBSWG.X-mm	..\scr_itw.dec02\vbs\v\VBSWG\X
..VBS/VBSWG.Z-mm	..\scr_itw.dec02\vbs\v\VBSWG\Z

Table 19: Comparison of the Testbed structure and the Wildlist

The script testbed contains different variants⁵⁰ for each of the virus families⁵¹, which makes a total of 22 different viruses in the testbed that the anti-malware products shall detect.

The names of viruses as listed in the wildlist might differ from those in the testbeds (i.e. SSIWG2 vs. Daira) and there are also cases where viruses have been identified in more detail by the procedures used at the aVTC labs (i.e. Stages.A), but the testbed reflects the ITW viruses as reported by the wildlist. A clear identification is not always possible, and different up-to-date scanners are used to identify the malware, as they might produce different identifications there are sometimes additional entries in the testbed. For more information on the mapping of inconsistent identifications from different vendors see [VGREP] and also [HEIMANN2002] for details on the compilation of testbeds.

For each of the 22 virus variants at least one malicious sample is contained in the testbed, for some variants there are a lot of samples, for some just one. In total the testbed scr_itw.dec02 contains 178 different malicious samples, which shall all be detected by the anti-malware products.

More macro than script viruses have been reported to be in-the-wild, so the testbed of ITW-macro viruses contains 976 malicious samples, but from 76 different macro virus variants.

See the appendix C for a complete listing of each testbed and [HEIMANN2003], [SEEDORF2001] or the aVTC website [VTCWEBSITE] for further details on the testbed compilation for aVTC tests.

⁵⁰ For example “Loveletter.A@MM” and “Loveletter.C” are two variants of the “Loveletter” family.

⁵¹ An example of a virus family is the family of “Loveletter” viruses.

7.2.4.2 Test Environment

The test will be performed on a mobile device of the first generation (see 3); the test device is a T-Mobile MDA (see Appendix B).

To recall, first generation Pocket PC 2002 devices have

- 32 MB RAM,
- 32 MB ROM (mainly used by the operating system and pre-installed applications) and an
- ARM SA-1110 processor running at 206 MHz.

The MDA was running version 3.012039 (Built 11178) of Microsoft Pocket PC, the ROM-Version was 3.16.36 GER (from February 18, 2003) and the Radio-Version was 3.17.01.

The MDA's built-in GSM phone was not used, no SIM card was inserted, and once the mobile device was turned on, the software was used to turn the built-in phone off.

The mobile device is connected to a desktop machine, running Windows 2000, via the USB connection. The desktop is running ActiveSync Version 3.6 (Build 2148) to facilitate the connection to the mobile device. It has no further outside connection (i.e. no internet or network access).

The testbeds come from a CD-ROM or FLOPPY disk and no anti-malware is installed on the desktop. The tests are conducted in six steps, which will be described next.

7.2.4.3 Test Steps

The test procedure can be separated in six different steps, as shown in the following figure.

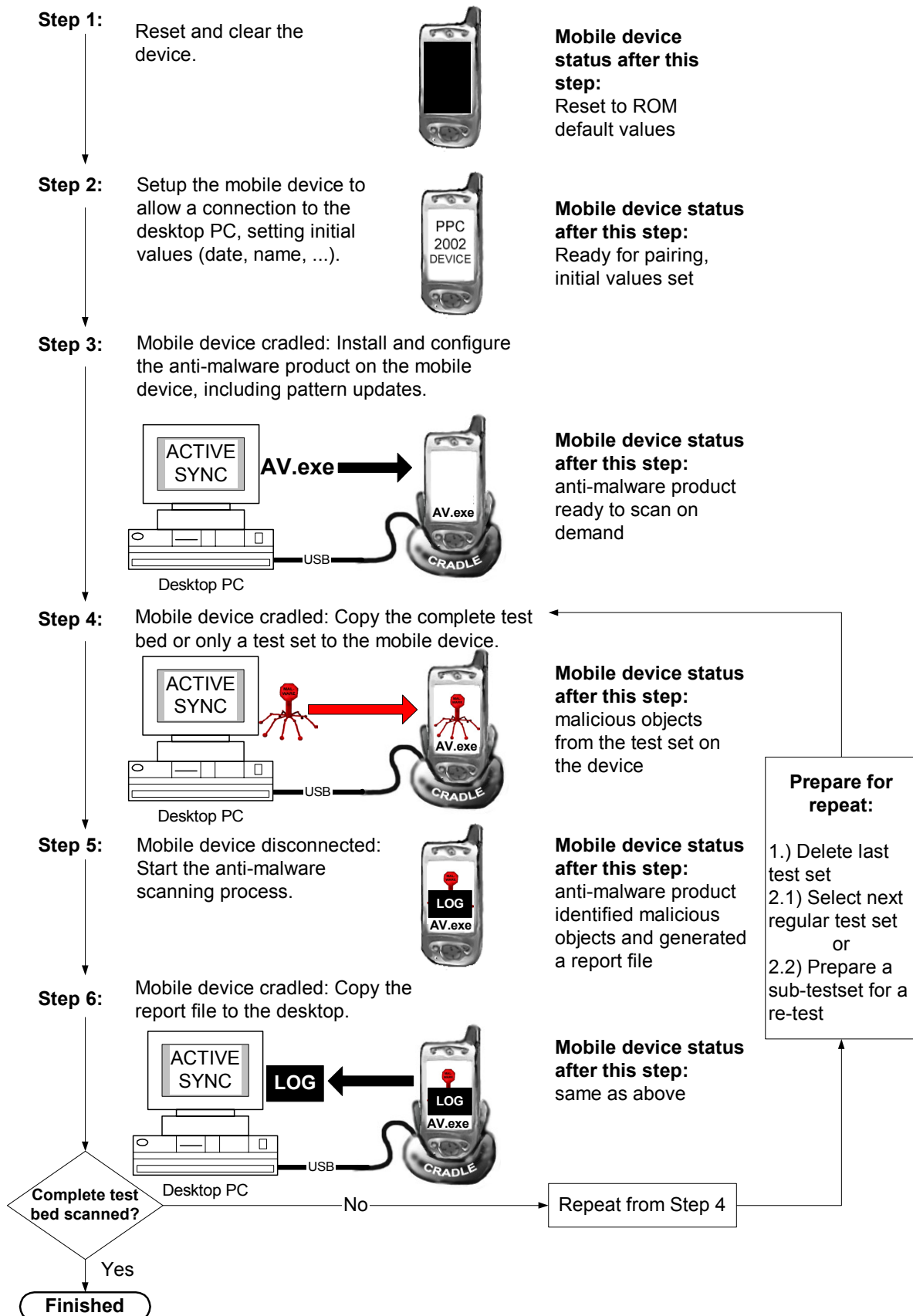


Figure 54: Six different steps of the test procedure

Step 1: Reset and clear the mobile device

No other software than the anti-malware software and the default software from the MS Pocket PC 2002 operating system and the MS Pocket PC 2002 default applications will be on the mobile device during the test.

To bring the mobile device into a default state, the information held in RAM or any other user memory needs to be erased, so it only loads the default application and operating system code and values from the ROM.

At first, all the data residing in other permanent storage facilities (like SD-cards or the iPAQ FileStore⁵²) shall be disconnected or erased (through operating system functions) to have a clean mobile device. Then the main memory (RAM) is erased performing a so-called hard reset.

The most secure way to achieve this is disconnecting the battery powering the RAM for several minutes so that all RAM based data is erased permanently. As this does not allow automating the process of installing anti-malware products on mobile devices, there are third party software products, for example PHM Task Manager [PHMTASK] or certain API functions [MURATOV2001], which can initiate a hard reset through software. This might be a way to automate the process of installing the anti-malware products on the mobile device, but has not been looked at further for this work.

However, not only the mobile device needs to be reset, also the desktop computer's settings concerning connected mobile devices are removed.

In the test case of MS ActiveSync, this is done by deleting the all partnerships for prior mobile devices and restarting the MS ActiveSync application.

For the test mobile device, an MDA, the method of manually disconnecting the battery was used to perform the hard reset; this is achieved by pressing the main battery on/off switch [MDAMANUAL]. For other mobile devices such as an iPAQ, other more complicated button press sequences are needed [HPFAQRESET]. The test mobile device RAM was left without any battery power for at least two minutes.

Step 2: Setup the mobile device

The mobile device is being restarted. Only the settings necessary to get the device to be connected to the host computer are made.

The following settings are made:

- Set the test time and date. The date and time was set to shortly after the submission deadline, but randomly chosen: 19th February 2003 - 12:00 for the pre-test and 18th February 2003 - 13:00 for the final test.
This time is also been set for the desktop computer, as during the cradled connection state the desktops clock might be used to synchronise the mobile device's time.
- Changing the device name⁵³ of the mobile device to "ppc".
- Then establishing the necessary outside connections: Especially the MS ActiveSync connection must be established before the installation of the anti-malware product, this includes the pairing process⁵⁴ to allow the mobile device to get into the connection state: cradled.

Step 3: Install and configure the anti-malware product

The mobile device is cradled and the anti-malware product is installed on the mobile device, by using the installer or other routines provided by the anti-malware vendor. The anti-malware product is then configured, especially not to monitor the exchange of files over cradled connections, as used to transfer the malicious samples in the next step. The installation and configuration steps are described for each product participating in the test in more detail in chapter 7.3.

Finally, the anti-malware software is terminated using the standard dialog for closing running applications (see Figure 12: Closing running programs).

⁵² The iPAQ FileStore is an installable file system that uses unused ROM-space in iPAQs to store files to provide a backup for battery losses or hard-resets.

⁵³ On German Pocket PCs "Gerätename", found under Start → Settings → System → Info → Device ID

⁵⁴ For the ActiveSync connection the so-called "Standard partnership" is chosen.

Step 4:

The test shall mimic a real life environment as best as it can, therefore the scanning will take place on the mobile device.

The total amount of memory available on the mobile device used for testing is limited to the 32 MB RAM. Only a portion of this can be used to store the malicious samples because already the anti-malware program files and the report file will also consume RAM. Additionally other system files and other operating system data occupy RAM after the device restarts.

After a hard reset the testing mobile device reports a total of 25.89 MB of RAM as free.

For the tests, I have limited the size of malicious samples being on the mobile device to 16 MB. This leaves more than 8 MB for the anti-malware products and their report data. Memory is a spare resource on mobile devices, but under the given circumstances of the size of anti-malware products (see 7.3 for details for different products) leaving 8 MB is a fair amount of memory.

This might make it necessary to split the testbed in chunks, referred to as test sets (see definition in 0).

The malicious samples from the actual test set are then copied to the mobile device's RAM via the cradled connection. The ActiveSync connection is set up in such a way that no conversion of exchanged files takes places. No anti-malware scanner is running on the desktop. The mobile device's anti-malware software was also setup in Step 3 not to touch or modify the transferred files.

Step 5:

The mobile device will then be detached from the cradled connection with the desktop, in this case the MS ActiveSync USB connection. It will be set into disconnected state. The mobile device will however remain connected to an external power supply to prevent battery exhaustion failures. The anti-malware software is started and the scan process is performed on the stand-alone mobile device. After the completion of the scan process, the report file is saved and the anti-malware software is terminated.

Step 6:

As last step, the mobile device will then be brought back into a cradled state, so its re-connected through MS ActiveSync, and the report file is copied to the desktop PC for analysis.

If the analysis shows the necessity for a re-test, the test set for the re-test is prepared. The old test set is removed and the process is repeated from step 4, with the copy of the re-test test set. Two such re-tests are permitted for each anti-malware product.

If another test set or a different testbed needs to be scanned with this scanner, the old test set is removed and the process is again repeated from step four with the new.

If all test sets from all testbeds have been scanned the process finishes and is repeated for another anti-malware product from step 1.

7.3 Anti-Virus Products for Pocket PC 2002

The following table gives an overview about the anti-malware products that are targeted at or marketed for PDA devices. For the compilation of this table, I checked the anti-malware vendor websites, of all vendors who submitted products for testing in previous aVTC tests [VTCWEBSITE], or vendors who submitted products for testing at virus bulletin [VBENDORLIST]. I also checked [DEDO2002] and [HERRERA2001] for further references to anti-malware products running on Pocket PC 2002.

Anti-malware products that were tested are indicated with "Yes tested" in the column PPC 2002. Only a "Yes" in the PPC column means that this vendor's malware product is for various reasons not included in the test, but that this vendor has a product targeted at the Windows CE 3.0 operating system or for the Pocket PC 2002 platform.

Name or aVTC-Abbreviation ⁵⁶	Vendor Name	Products for PDAs ⁵⁵ ?			Name of PPC 2000 product
		Palm	Symbian	PPC 2002	
AVP	Kaspersky Lab	Yes	⁵⁷	Yes tested	Kaspersky Antivirus for Pocket PC
BDF (AVX)	Softwin	Yes		Yes	BitDefender for Windows CE
FSE	F-Secure		Yes	Yes tested	F-Secure Anti-Virus for Pocket PC 1.5
INO	Computer Associates	Yes		Yes tested	eTrust Antivirus 7.0 for Pocket PC
NAV	Symantec	Yes			
PCC	TrendMicro	Yes	Yes	Yes tested	PC-cillin for Wireless 2.0
SCN	Network Associates	Yes	Yes	Yes	VirusScan Wireless
AhnLabs V3	AhnLabs	Yes			

Table 20: List of anti-malware products vendors and their support of PDA operating systems.

A call for products was made to all anti-malware vendors, who regularly submit products for the aVTC tests, to submit products that run on Pocket PC 2002 and especially meet the conditions C1 and C2. The deadline for product submission was 17th of February 2003 at 24:00 GMT+1:00 hrs. This was also the latest date until any virus definitions can be updated to be used for the test.

The following three products were submitted for test: AVP, FSE, INO. Additionally PCC and BDF can be downloaded from their vendor’s website, where they are freely available. BDF and SCN were not tested for various reasons.

This makes a total of six products out of which four could be tested against several testbeds. In the following chapters 7.3.1 to 7.3.6 I will look at each product in more detail.

7.3.1 AVP: Kaspersky Anti-Virus for Windows CE [tested]

Kaspersky Anti-Virus was submitted for testing by the vendor, together with a license key file (KAVScanner.key).

The following statement from the user guide [AVPMANUAL] should be noted:
“Kaspersky Anti-Virus for Windows CE is able to detect only those viruses developed specially for the PDA running Windows CE. Files infected with Windows viruses and viruses developed for other operating systems cannot affect your PDA. If you copy a file infected with a Windows virus to your PDA, or if you move a message infected by a Windows virus into your pocket device e-mail program, Kaspersky Anti-Virus for Windows CE will not detect it.”

Not scanning for malware from other platforms lies within the requirements for submitted products, so it is tested, but the results are of limited use as it is tested against malware written for the desktop Windows platforms.

It is expected that Kaspersky Antivirus for Windows CE will not stop the distribution of today’s mostly Windows based malware, and that it will not detect any malware that is contained in the testbeds used.

7.3.1.1 Version Information

Version / Release:	Version 4.0.0.4
Build:	n/a
Pattern / Database:	File timestamp: 27.01.2003

Table 21: Kaspersky version information

⁵⁵ This does not necessarily mean that the products meet condition C1 and C2.

⁵⁶ A three-letter abbreviation indicates that the product has participated in prior aVTC tests.

⁵⁷ Empty fields indicate “No”.

7.3.1.2 Mobile Device's System Requirements

		Test-Device: MDA
Operating System:	Pocket PC 2000, Pocket PC 2002 (Windows CE 3.0)	Yes
Devices:	Based on MIPS or StrongARM CPU	Yes
Memory:	At least 150 Kb	Yes

Table 22: Kaspersky system requirements

7.3.1.3 Installation

Installation is done through an installer from the desktop system, which transfers the following files into the program files directory on the mobile device:

...\Kaspersky Anti-Virus\KAVBase.kvb	[size: 534b	timestamp: 27.01.2003]
...\Kaspersky Anti-Virus\KAVScanner.exe	[size: 93,5k	timestamp: 04.02.2003]
...\Kaspersky Anti-Virus\KAVScanner.key	[size: 1,9kb	timestamp: 22.04.2003]

The pattern is in the file KAVBase.kvp.

The install process also installs a component on the desktop PC, which establishes a connection to the Kaspersky server, downloads the latest pattern, and transfers it to the connected mobile device. More details on this in the update process section (7.3.1.6).

7.3.1.4 Scanner Options

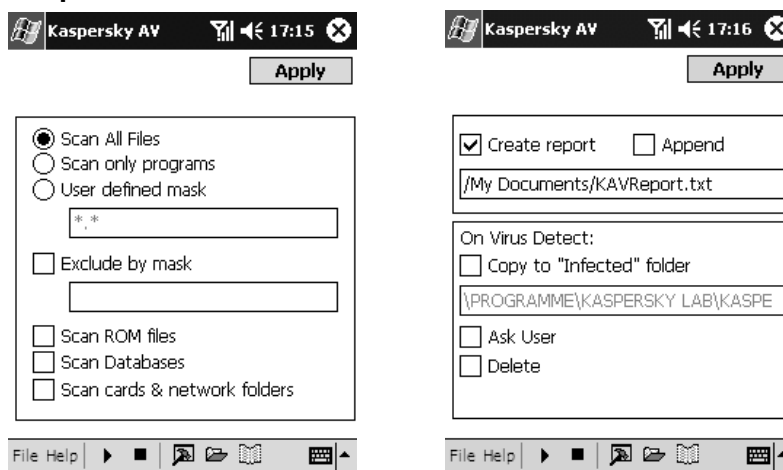


Figure 55: Kaspersky AV scanner options

The following settings were used for the tests:

Action:	No action, but create a new report file
Files:	Scan All Files
Scan ROM files:	No
Scan Databases:	No
Scan files on storage card & networks:	No
Automatic Protection:	Not available

Table 23: Kaspersky AV scanner options

7.3.1.5 Scan Process and Scan Report

The scanning process is started by tapping at the play symbol in the menu bar, once started the complete mobile device is scanned. So no explicit folder can be selected and the entire mobile device is scanned, so a pre-scan is necessary (see 7.2.3.3). As Kaspersky AV does not recognize the seven EICAR test files, used as a test set to evaluate the scanner report. Scanning the mobile device with the EICAR test files generates the following report, which can be saved as a text file:


```
Scanned:
Files: 187
Folders: 42
Cards: 0
DBases: 0

Viruses found 0
Deleted: 0
Moved to "Infected" 0
Errors: 0
*****
```

Figure 56: Kaspersky AV scan report

The manual [AVPMANUAL] gives an impression what is to be expected in the report file, if Kaspersky AV will find malware, this example can be seen in Figure 57. Instead of recognizing the EICAR test file, two test files infected with TestVirus1 and TestVirus2 are used.

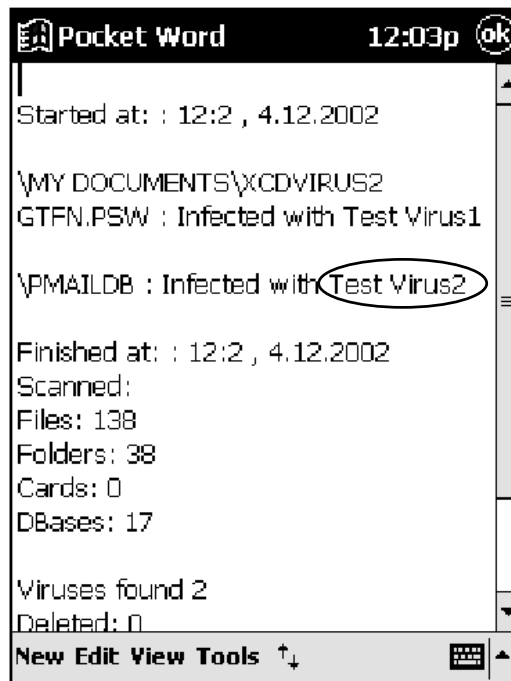


Figure 57: Kaspersky AV report example from the manual [AVPMANUAL]

As can be seen, the Kaspersky AV report meets the condition A2 and contains the necessary information, including the identification, to calculate the measurements.

7.3.1.6 Update Process

During the installation of Kaspersky Anti-Virus for Windows CE, a component is installed on the desktop PC, which is responsible for updating the pattern on the mobile device. The main screen of this application is shown in Figure 58.



Figure 58: Kaspersky Anti-Virus Updater application on the desktop PC

The Kaspersky Anti-Virus Updater for Windows CE will check and if required download an updated pattern file from the Internet, either on demand (when instructed with the “Get Update” button), on every connect or when the scheduled interval has elapsed. If a new pattern was downloaded, it is transferred to the cradled mobile device without user interaction [AVPMANUAL].

The only way to update the pattern is while the mobile device’s connection is in a cradled state is connected to a desktop that is running the update application.

7.3.2 BDF: Softwin BitDefender for Windows CE [not tested]

BitDefender for Windows CE Version 1.01 can be downloaded from the vendor’s website [BDFDOWNLOAD] as a free tool. It seems to have problems running on Pocket PC 2002 devices. As the vendor also submitted products for other operating systems for a test at the aVTC, but did not submit this product for running on Pocket PC 2002, although it got the call for Pocket PC products, it will not participate in this test.

7.3.3 FSE: F-Secure Anti-Virus for Pocket PC Version 1.5 [tested]

F-Secure Anti-Virus for Pocket PC resides locally on the mobile device, so the condition C1 is met and the software is tested. The software was submitted by the vendor, including the license key and a download to a pattern, which was up-to date at the submission deadline.

Different e-mails exchanged with the vendor established the fact that F-Secure Anti-Virus for Pocket PC will only detect malware especially targeted at the Pocket PC platform.

The requirements for submitted products are still met, although F-Secure Anti-Virus does not scan for malware from other platforms. Therefore, it is tested, but the results are of limited use as it is tested against malware written for the desktop Windows platforms.

So F-Secure Anti-Virus for Pocket PC is expected not to stop the distribution of today’s mostly Windows based malware, and not to detect any malware that is contained in the testbeds used.

7.3.3.1 Version Information

Version / Release:	F-Secure Anti-Virus for Pocket PC Release 1.5
Build:	Build 11
Pattern / Database:	Database timestamp: 16.12.2002 – 12:13

Table 24: F-Secure version information

7.3.3.2 Mobile Device’s System Requirements

According to the user's guide [FSEMANUAL] and the release notes [FSERELNOTES] the following requirements must be met by the mobile device:

		Test-Device: MDA
Operating System:	MS Windows CE 3.0	Yes
Devices:	Compaq iPAQ Pocket PC 2000 and 2002 handhelds H3600/H3700/H3800/H3900 series Symbol Pocket PC 2002 handheld HP Jornada Pocket PC 2002	No, but tested as all Pocket PC 2002 devices run on the same hardware platform (ARM).
Memory:	600 Kb	Yes

Table 25: F-Secure system requirements

7.3.3.3 Installation

The installation is done through an installer from the desktop system. I also downloaded an updated pattern file from the F-Secure website, which was transferred manually (which is why the timestamp differs). After the installation has finished the following files were found in the program files directory on the mobile device:

```

... \F-Secure Anti-Virus\fsav.dll           [size: 7k      timestamp: 11.07.2002]
... \F-Secure Anti-Virus\fsav.exe          [size: 153k    timestamp: 14.08.2002]
... \F-Secure Anti-Virus\fsavppc.dat      [size: 379b    timestamp: 29.04.200358]
    
```

The pattern is in the file fsavppc.dat.

The F-Secure scanner also installs a link in the today screen, which allows the user to quickly scan all files as pictured in Figure 59.

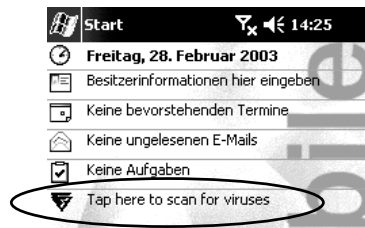


Figure 59: Today screen with F-Secure Anti-Virus

Tapping "Tap here to scan for viruses" starts a scan of all files on the mobile device.

7.3.3.4 Scanner Options

The scanner does not allow to only log the scan process results, but the option "Ask after scan" allows to scan the complete test set, and then to dismiss any action, which will lead to the report file generation (so condition A3, A4 and A9 are met).

The automatic start of the scanning process on different events will be disabled for the test.

⁵⁸ This is just a file date, representing the day when it was manually transferred to the mobile device, this is not the actual date of the pattern, which is reported by the software as being 16.12.2002 meeting the submission deadline.

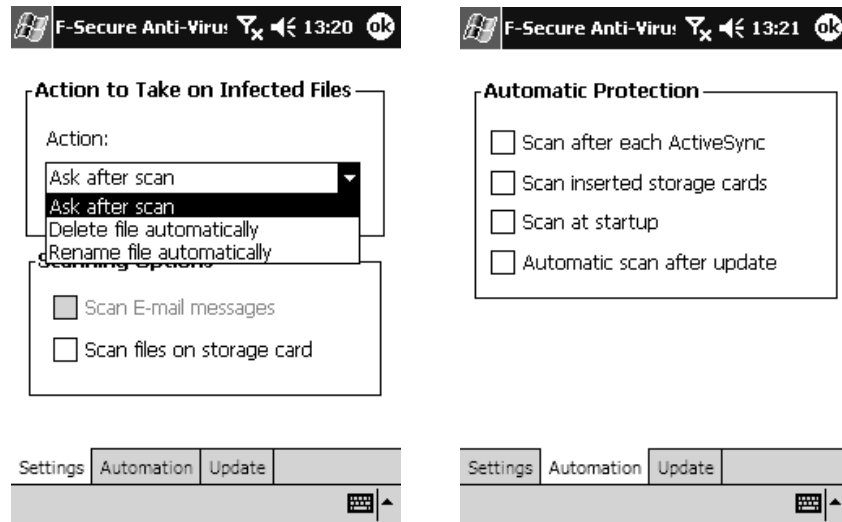


Figure 60: F-Secure scanner options

The following settings were used for the tests:

Action:	Ask after scan
Files:	
Scan files on storage card:	No
Automatic Protection:	All No

Table 26: F-Secure scan options

7.3.3.5 Scan Process and Scan Report

The scanning process can be set to only scan the folder containing the test set, using the menu “Tools -> Scan Folder ...”. After the scan has finished the log is presented, and can be saved in HTML-format only.

To verify that the information provided in the report file is sufficient to calculate the measures, I have also initiated a test scan over a small set of seven EICAR test files in six folders.

This is the report file generated from the test scan (it has been reformatted from the original HTML-format):

Scanning Report 29.04.03 12:16

Database timestamp:

16.12.02 12:13

Scan target:

\My Documents\small test eicar

Scanned files and attachments:

Scanned: 7

Infected: 7

Renamed: 0

Deleted: 0

Infections:

1. \My Documents\small test eicar\e2\eicar2.com
EICAR Test File

2. \My Documents\small test eicar\el\eicar1.com
EICAR Test File
3. \My Documents\small test eicar\el\el_2\eicar1_2.com
EICAR Test File
4. \My Documents\small test eicar\el\el_1\eicar1_1.com
EICAR Test File
5. \My Documents\small test eicar\el\el_1\el_1_2\eicar1_1_2a.com
EICAR Test File
6. \My Documents\small test eicar\el\el_1\el_1_2\eicar1_1_2b.com
EICAR Test File
7. \My Documents\small test eicar\el\el_1\el_1_1\eicar1_1_1.com
EICAR Test File

Scan time 1 seconds

Figure 61: F-Secure scan report

It contains the complete name and path of the identified malicious samples and a summary, so it meets condition A2. In addition, the identified name of the malware is reported.

7.3.3.6 Update Process

After supplying the correct update Internet address⁵⁹, called “Update URL”, the update can be done whenever an Internet connection is available. If and only if required, a new database will be downloaded and saved in the file `fsavppc.dat`, the old database will be saved in `fsavppc.old`.

Figure 62 on the next page shows the update dialogs of F-Secure AntiVirus for Pocket PC.

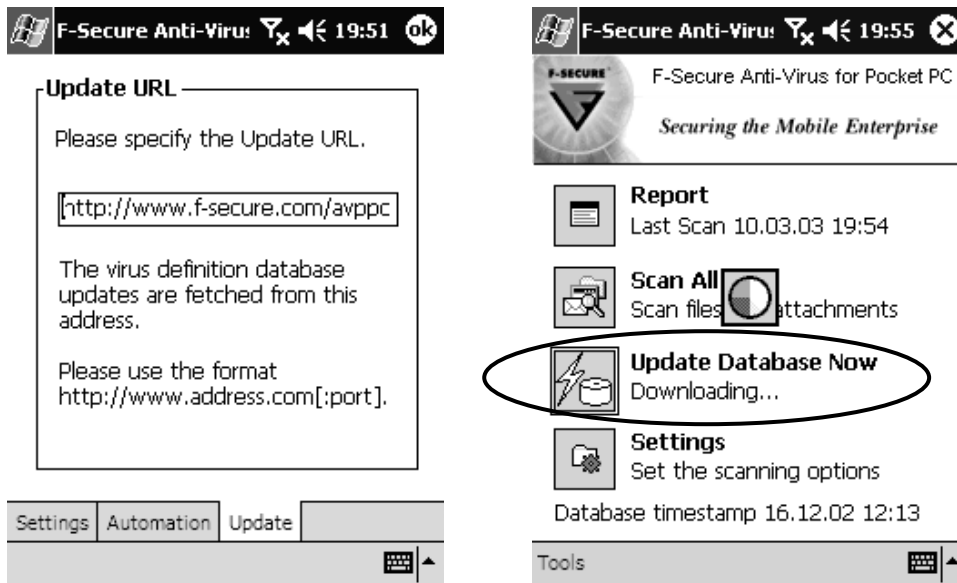


Figure 62: Update URL and Database Update

An update log is also written to document the steps taken; the size of this log seems to be limited to a maximum of about five Kbytes.

⁵⁹ <http://www.f-secure.com/avppc>

7.3.4 INO: Computer Associates eTrust Antivirus 7.0 for Pocket PC [tested]

Microsoft lists this product in the virus protection section of a Pocket PC Security document [DEDO2002], with the following entry: “InoculateIT virus pass through protection”.

eTrust for Pocket PC was submitted by its vendor Computer Associates (CA) as a “release candidate” version to meet the submission deadline, whilst the official release for the Pocket PC platform was the 24th of February 2003. Assured “that there will be no changes needed for these two platforms before GA (general availability)” [e-mail by Shali Hsieh, 18th February 2003], the product was tested as submitted, together with a submitted pattern-update, which dates before submission deadline.

eTrust Antivirus for Pocket PC shall offer “Stand-alone, on-demand antivirus protection against all known Microsoft Word, Excel, PowerPoint macro virus and script viruses/worms on the Pocket PC 2002 platform” according to the user guide [INOMANUAL]. So, it meets condition C1 and is tested.

7.3.4.1 Version Information

Version / Release:	eTrust Antivirus for Pocket PC, Version 2.00.31
Build:	Program: 2.00.31 – 27.11.02
Pattern / Database:	Engine: version 23.59.00 – 17.12.02 Data: version 23.59.36 – 10.02.03

Table 27: eTrust version information

7.3.4.2 Mobile Device’s System Requirements

The manual [INOMANUAL] does not have much information on the mobile device’s requirements, but there have not been any problems running the product on the testing mobile device.

		Test-Device: MDA
Operating System:	“any Pocket PC running the Pocket PC 2002 operating system” [INOMANUAL]	Yes
Devices:		Yes
Memory:	No requirements given	Yes

Table 28: eTrust system requirements

7.3.4.3 Installation

eTrust was installed using the setup, which installs the program on the mobile device. After this, I installed the also submitted signature update. After the complete installation, the following files are present in the mobile device’s program files directory:

... \CA\ eTrust Antivirus\avh32dll.dll	[size: 298k	timestamp: 17.12.2002]
... \CA\ eTrust Antivirus\avoem.dll	[size: 11k	timestamp: 27.11.2002]
... \CA\ eTrust Antivirus\avui.exe	[size: 216k	timestamp: 27.11.2002]
... \CA\ eTrust Antivirus\inoavgz.dll	[size: 37k	timestamp: 27.11.2002]
... \CA\ eTrust Antivirus\InoAVUpd.exe	[size: 169k	timestamp: 27.11.2002]
... \CA\ eTrust Antivirus\license.dll	[size: 21k	timestamp: 27.11.2002]
... \CA\ eTrust Antivirus\license.lic	[size: 1k	timestamp: 25.03.2002]
... \CA\ eTrust Antivirus\update.txt	[size: 1k	timestamp: 17.07.2002]
... \CA\ eTrust Antivirus\virsig.dat	[size: 296k	timestamp: 10.02.2003]

The pattern resides in the file virsig.dat.

7.3.4.4 Scanner Options

The scanner can be configured (see Figure 63) to react in four different ways if an object is found to be malicious: User intervention (“Ask user”), log (“Report only”), repair object (“Cure file”) or destroy object (“Delete file”). For the test, the “Report only” option was chosen to meet the conditions A3, A4 and A9.

eTrust Antivirus can react on the insertion of mobile storage media, and initiate the scan process automatically, once configured to do so.

The option “Monitor file updates” automatically scans objects that are transferred to the mobile device during ActiveSync sessions. For the test, these options are not enabled.

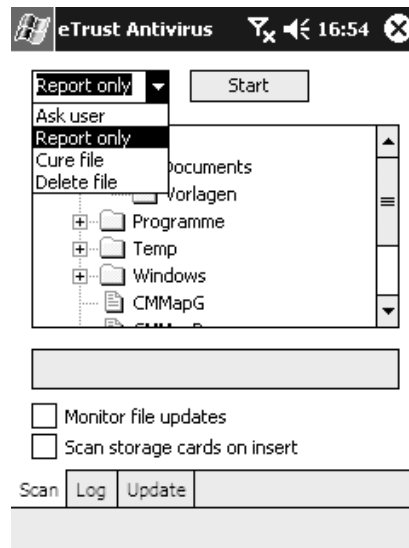


Figure 63: eTrust Antivirus scanner options

The following settings were used for the tests:

Action:	Report only
Files:	Scans all files unless instructed different
Automatic Protection:	
Monitor file updates:	Needs to be enabled to save report⁶⁰
Scan storage cards on insert:	No

Table 29: eTrust scan options

7.3.4.5 Scan Process and Scan Report

The scanning process can be set to only scan the folder containing the test set. After the scan has finished, the log is presented.

This report can then be saved. As already mentioned there is a problem with the version tested, that only if “Monitor file updates” is enabled eTrust Antivirus succeeds in presenting the save dialog, otherwise the program just quits and the report cannot be saved. So the option is enabled. The report file can be saved as text.

To get an impression of the report file generated by eTrust Antivirus see the result of a scan of seven EICAR test files in Figure 64.

⁶⁰ “Monitor file updates” is disabled only during the transfer of the samples, as I discovered a problem that it needs to be enabled to allow the program to save the report. So it is enabled during the actual scanning process.

```

Antivirus engine initialized successfully
-----
*** 12.05.03 12:54:35
*** On demand scan started
\My Documents\small test eicar\e1\eicar1.com is infected with EICAR
test file
\My Documents\small test eicar\e1\e1_1\eicar1_1.com is infected with
EICAR test file
\My Documents\small test eicar\e1\e1_1\e1_1__1\eicar1_1__1.com is
infected with EICAR test file
\My Documents\small test eicar\e1\e1_1\e1_1__2\eicar1_1__2a.com is
infected with EICAR test file
\My Documents\small test eicar\e1\e1_1\e1_1__2\eicar1_1__2b.com is
infected with EICAR test file
\My Documents\small test eicar\e1\e1_2\eicar1_2.com is infected with
EICAR test file
\My Documents\small test eicar\e2\eicar2.com is infected with EICAR
test file
*** On demand scan has been completed
Total Files scanned 7
Total Directories scanned 7
Total Viruses found 7
Total Viruses cured 0
Total Files deleted 0
    
```

Figure 64: eTrust scan report

The report lists the complete path of files found to be infected and gives total sums of all scanned files, as required by condition A2. Additionally eTrust also reports the name of the malware.

7.3.4.6 Update Process

The update function uses an Internet connection to download the latest pattern from a list of servers via FTP. The list of servers is contained in a text file⁶¹ and it seems to allow for a list of 10 different servers from where the patterns and engine updates are downloaded. The patterns are only downloaded when an update is needed (see Figure 65 right), to save bandwidth.

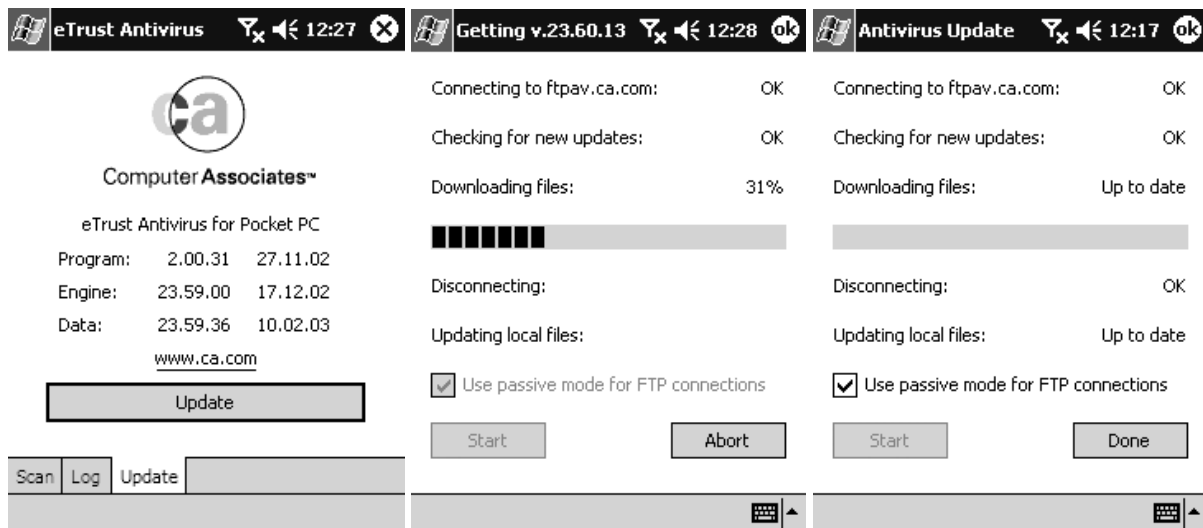


Figure 65: Pattern information and update process for eTrust

Facilitating an Internet connection allows the user to update when ever the mobile device is wirelessly connected and has Internet access, not only when cradled.

⁶¹ Filename is update.txt, it is located within the installation directory.

7.3.5 McAfee VirusScan Wireless [not tested]

This software according to information from the vendor “closes the gap in your security. It scans your handheld for PC viruses each time you attempt to sync with your PC” [NAIWEBINFO].

This anti-malware software does not run stand-alone on the mobile device, and therefore does not meet C1, it was also not submitted by the vendor. It is not tested as part of this test.

7.3.6 PCC: PC-cillin for Wireless [tested]

PC-cillin for Wireless runs locally on the mobile device, so it meets C1 and is tested.

The software is available as a free download from the companies website [TRENDPCCWEB] and was downloaded as a scheduled task minutes before the deadline, I also downloaded the pattern file number 345 [TRENDPATTERNWEB] from that day from the website.

Before interpreting the results the following information from the manual [TRENDREADME] shall be taken into account: “PC-cillin for Pocket PC version 2.0 can only detect Excel-type and Word VB script-type viruses.”

7.3.6.1 Version Information

Version / Release:	PC-cillin Wireless for Pocket PC Version 2.0
Build:	n/a
Pattern / Database:	Engine version: 5.200-0522 Pattern version: 345

Table 30: PC-cillin version information

7.3.6.2 Mobile Device’s System Requirements

According to [TRENDMANUAL] the requirements listed in Table 31: must be met by the mobile device to run PC-cillin wireless.

		Test-Device: MDA
Operating System:	MS Windows CE 3.0	Yes
Devices:	ARM-Processor (Compaq iPAQ; Strong ARM CPU) MIPS-Processor (Casio Cassiopeia) SH3-Processor (HP Jornada) RAM: 16 MB	Yes
Memory:	1 MB	Yes

Table 31: PC-cillin system requirements

7.3.6.3 Installation

After downloading the appropriate installer (depending on the processor type), PC-cillin for Wireless is installed via the installation program on the mobile device. Additionally the appropriate pattern was copied onto the mobile device. After completing the installation the following files can be found in the program files directory on the mobile device:

```

...TrendMicro\PC-cillin for Pocket PC\Common.vir    [size: 58k    timestamp: 10.11.2000]
...TrendMicro\PC-cillin for Pocket PC\PcciWinCE.exe [size: 41k    timestamp: 11.04.2001]
...TrendMicro\PC-cillin for Pocket PC\slm$vpn.345  ..[size: 91k   timestamp: 11.09.2002]
...TrendMicro\PC-cillin for Pocket PC\Vsapice.dll   [size: 644k   timestamp: 11.04.2001]
    
```

The pattern file is slm\$vpn.345.

7.3.6.4 Scanner Options

No options can be set for scanning, but no action is taken other than to report the infected files, so conditions are met.

7.3.6.5 Scan Process and Scan Report

Unfortunately, only all files on the mobile device can be scanned. So to allow to count the number of missed objects a pre-scan is performed just before the test set is copied onto the mobile device, to record the number of “clean”, non test set related, objects.

After PCC has finished scanning a report is presented to the user, a report log is automatically written to a file (condition A1 met). While the report in the PCC application lists the identification of the malicious code, the log file that is written to disk does not. As a result, the identification quality is very poor, as no identification is given at all. It does contain the full path to the identified malicious object though and a summary, so it meets condition A2. The scan report is in text format and looks like this:

```
Found 7 viruses.
Scanned 928 files.
Date: 05/08/2003  Time: 10:54

Action: pass
File: \My Documents\small test eicar\e2\eicar2.com
Date:05/ 08/2003  Time: 10:54

Action: pass
File: \My Documents\small test eicar\e1\eicar1.com
Date:05/ 08/2003  Time: 10:54

Action: pass
File: \My Documents\small test eicar\e1\e1_2\eicar1_2.com
Date:05/ 08/2003  Time: 10:54

Action: pass
File: \My Documents\small test eicar\e1\e1_1\eicar1_1.com
Date:05/ 08/2003  Time: 10:54

Action: pass
File: \My Documents\small test
eicar\e1\e1_1\e1_1__2\eicar1_1__2a.com
Date:05/ 08/2003  Time: 10:54

Action: pass
File: \My Documents\small test
eicar\e1\e1_1\e1_1__2\eicar1_1__2b.com
Date:05/ 08/2003  Time: 10:54

Action: pass
File: \My Documents\small test eicar\e1\e1_1\e1_1__1\eicar1_1__1.com
Date:05/ 08/2003  Time: 10:54
```

Figure 66: PC-cillin scan report

7.3.6.6 Update Process

PC-cillin does not have an automated update process neither on the mobile device nor on the desktop side. The pattern file needs to be downloaded from the web site [TRENDPATTERNWEB]. The web site from which I downloaded the latest pattern at the night from the 17th to the 18th of February for the test is shown in Figure 67.



Figure 67: Trendmicro's pattern download web site

After a download, the pattern needs to be unzipped and then copied into the PC-cillin program folder on the mobile device.

According to [TRENDFAQ] PC-cillin will then use the pattern with the higher version number and delete the old pattern file automatically. This is problematic as the downloaded version comes with an old pattern (dated November 2000), which is numbered 795. PCC will not know that pattern #345 is the newer one, so the user must delete the older pattern #795 manually.

More details on the manual update process can be found in [TRENDFAQ].

7.4 Test Results for AV-Products for Pocket PC 2002

In this chapter, especially in the figures the three letter abbreviations are used to reference the products and the internal name of the testbeds are used to clearly reference the different testbeds. Please see chapter 7.2.4.1 or the Appendix C for more information about the Testbeds. The three letter product abbreviation is resolved in each of the result figures, more details on each product were given in the previous chapter 7.3.

First, the results for the pre-test, that was conducted using the preliminary testbeds from older aVTC tests (see 7.2.4.1), are presented. This pre-test is also available as Pocket PC Scanner Test 2003-05 [PPCTEST2003-05] and is found in the Appendix F. The pre-test results can be used to determine how good the products can detect older viruses and will show that the two products AVP and FSE that were not expected to detect any desktop malware, actually behave as expected and do not detect any malware from the testbeds.

Of course, after the results of the pre-test are evaluated, the results on the latest testbeds containing also in-the-wild viruses from as late as December 2002 are stated and commented. From those the products ability to detect new and actual viruses can be determined. Actually only the remaining products INO and PCC were tested against those testbeds, as the others are again expected to not detect any of the desktop malware used in all the testbeds.

To provide a better comparison to the desktop test results I will apply the same grading as used in the desktop aVTC tests to rate the two Pocket PC products.

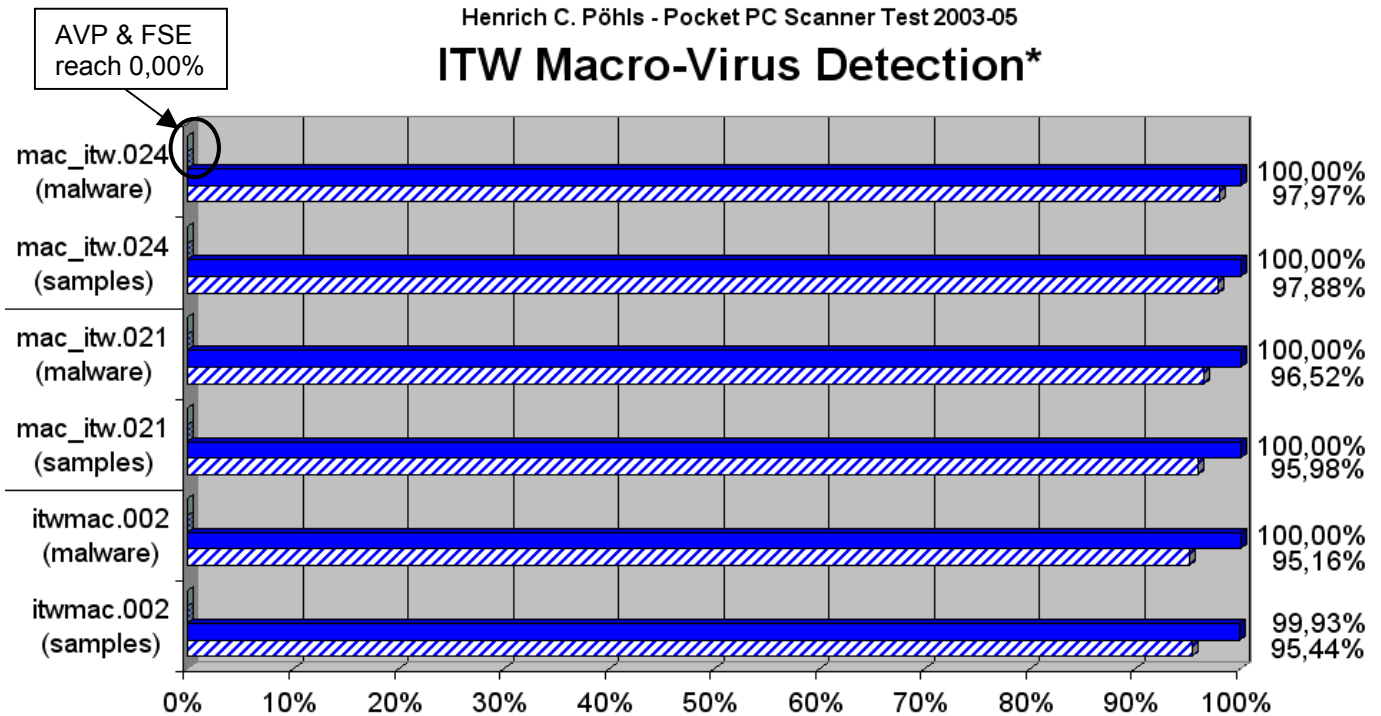
Then the two products are compared head-to-head and against the average of the Windows 2000 desktop products (see desktop test [TEST2002-12]).

In chapter 6.5.4 I identified the vulnerability auto-run from removable storage media. This would allow malware to get automatically invoked and execute from an inserted storage card. In the desktop environment most of today's anti-malware products allow the user to constantly scan for known malicious code using an on-access scanner; and so identify known malware before it is executed. This on-access scanning would also prevent the execution of a malicious auto-run code on a storage media. With this in mind I additionally analysed if any of the tested programs will allow to mirror such behaviour on Pocket PC mobile devices, the results are shown in 7.4.4.

7.4.1 Evaluation of Test Results for the Macro In-the-Wild testbeds

7.4.1.1 Pre-Test: Macro In-The-Wild

The following figure compares the detection rate of the four participating anti-malware products when tested on the three macro testbeds used in the pre-test. Detailed results, including the counts from each testbed are given in Appendix F.



*) Summarized results from the Pocket PC Scanner Test 2003-05, please read the eval.txt for full details. The products AVP and FSE do not attempt to detect Macro-Viruses written for the desktop environment, as used in the test-beds, they claim to detect Pocket PC relevant malware only, which is not part of any test-bed used. PCC states that it will only detect Excel-type and Word VB script-type script viruses. Please read the problems.txt from the full report for further details.

Figure 68: Pre-Test: ITW Macro-Virus Detection Rate (sample and malware based)

AVP & FSE:

AVP's and FSE's inability to detect other malware than those especially targeting the Pocket PC platform is shown, as they both detect no malicious samples from any macro testbed. Nevertheless, they report all the samples as scanned.

INO:

INO shows 100.00% macro virus and sample detection rates in nearly all testbeds; just in the oldest testbed (itwmac.002) it misses one single sample of the Word97 macro virus "W97M/Thus.A", which reduces the macro sample based detection rate to a still high 99.93%.

The detection quality for macro viruses of INO:

The unreliable detection (only the on sample missed in the itwmac.002) is low. Also INO's unreliable identification rate is rather low except a maximum value of 6.5% again on the testbed itwmac.002, which INO had the most problems with.

PCC:

PCC shows far lower detection rates for macro viruses than INO, ranging from below 98% down to 95.1%. Especially in the oldest testbed (itwmac.002) it nearly leaves the above 95% region. It additionally does not include any identification in the saved report, giving it the highest possible unreliable identification rate.

7.4.1.2 Final Test: Macro In-The-Wild

Results are shown with all details in the usual aVTC format in Table 32 and Table 33. They contain the totals and their percentage calculated based on the total number of viruses or samples contained in the testbed.

Results for Macro ITW Viruses (Testbed: mac_itw.304, Nov. 2001-Dec. 2002)								
Scanner	Viruses detected		This includes unreliable			Files detected		
			identified	detected				
Testbed	118	100.00%				277	100.00%	
INO	118	100.00%	5	4.24%	0	0.00%	277	100.00%
PCC	114	96.61%	114	96.61%	2	1.69%	266	96.03%

Table 32: Scanner Results for Macro ITW Viruses (Testbed: mac_itw.304, Nov. 2001-Dec. 2002)

Results for Macro ITW Viruses (Testbed: mac_itw.dec02, December 2002)								
Scanner	Viruses detected		This includes unreliable			Files detected		
			identified	detected				
Testbed	76	100.00%				976	100.00%	
INO	76	100.00%	5	6.58%	1	1.32%	975	99.90%
PCC	74	97.37%	74	97.37%	6	7.89%	950	97.34%

Table 33: Scanner Results for Macro ITW Viruses (Testbed: mac_itw.dec02, December 2002)

INO:

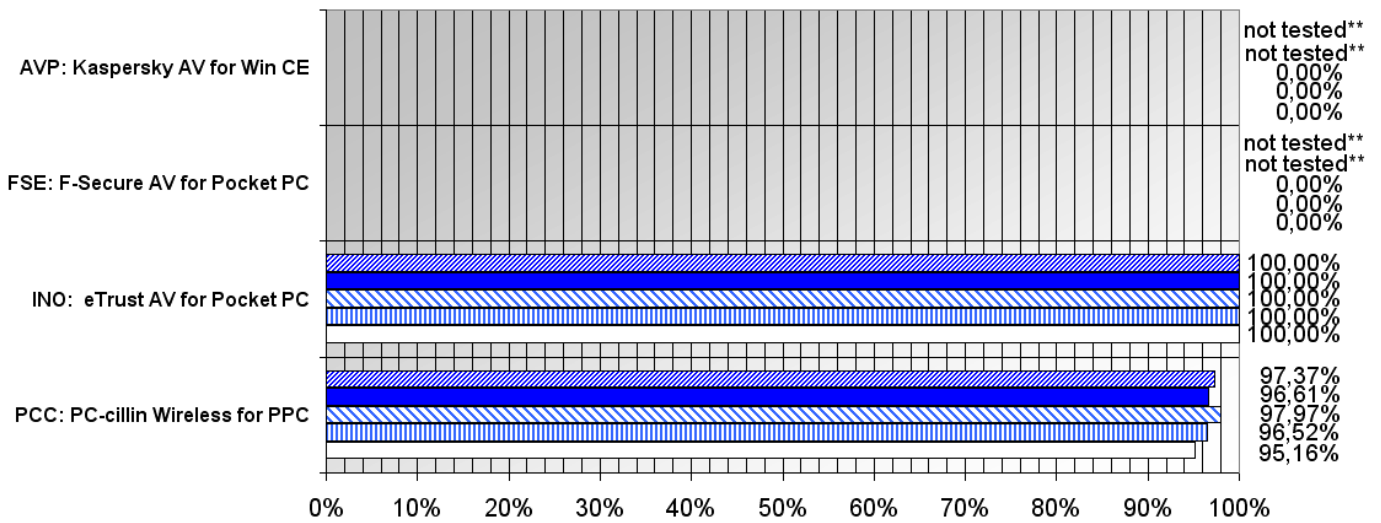
So this shows that INO keeps up the perfect macro detection rate of 100.00%, detecting all macro viruses in both testbeds. However, INO misses one sample of the most recent macro viruses in itw_mac.dec02. The one missed sample is from the virus “W97M/Thus.A”. The number of unreliable identified macro viruses is even higher then the worse value of 6.5% in one of the pre-test testbeds (itwmac.002).

PCC:

PCC shows again a lower detection rate than INO with only 96.6% and 98.6%. The comparison chart in Figure 69 shows more clearly that PCC’s detection rate, measured on a malware basis, is better for the newest macro viruses contained in the testbed mac_itw.dec02 than for older viruses (pre-test testbeds and also the last year’s viruses in mac_itw.304).

Henrich C. Pöhls - Pocket PC Scanner Test 2003-07

ITW-Macro Virus Detection Rate (malware) on all TestBeds*



*) Summarized results from the Pocket PC Scanner Test 2003-07, including the results from the previous Test 2003-05, please read eval.txt or my diploma thesis for full details. Detection Rate (malware) is calculated as the number of malware detected divided by the total number of malware in the test bed. PCC states that it will only detect Excel-type and Word VB script-type script viruses.

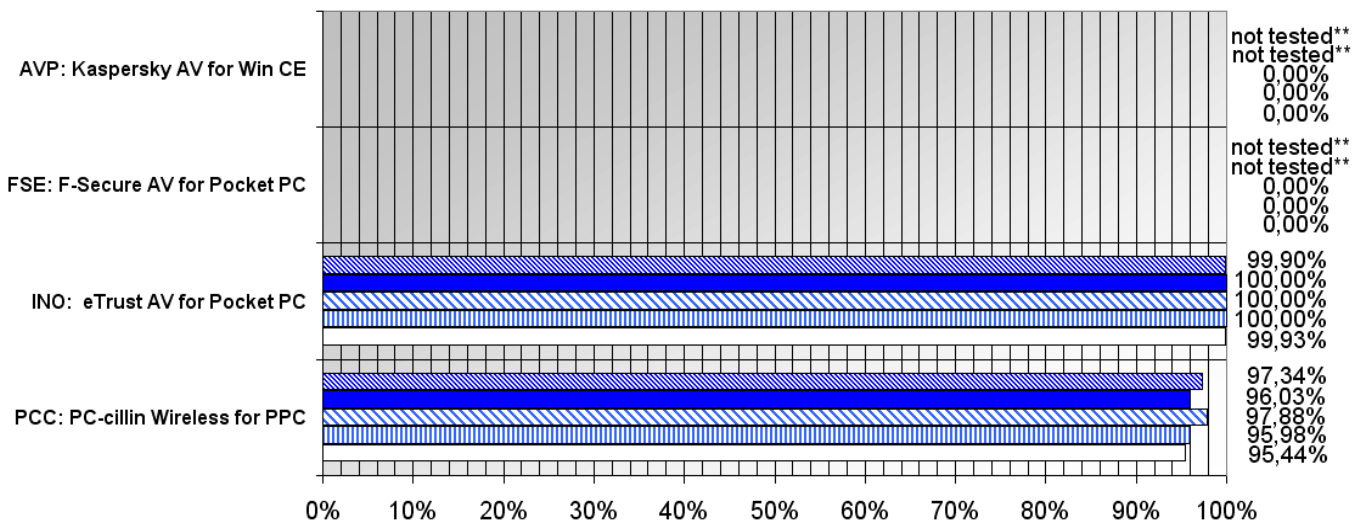
**) The products FSE and AVP do not attempt to detect Macro-Virus written for the desktop environment, as used in the test-beds, they claim to detect Pocket PC relevant malware only. Pocket PC malware does not exist today and is not part of any test-bed, so the products were not tested again against testbed mac_itw.304, after previous tests have shown their inability to detect other platform's malware. Please read the problems.txt from the full report or my diploma thesis for further details.

***)Date shows when the test beds were frozen.

Figure 69: ITW-Macro Virus Detection Rate (malware) on all TestBeds

Henrich C. Pöhls - Pocket PC Scanner Test 2003-07

ITW-Macro Virus Detection Rate (samples) on all TestBeds*



*) Summarized results from the Pocket PC Scanner Test 2003-07, including the results from the previous Test 2003-05, please read eval.txt or my diploma thesis for full details. Detection Rate (samples) is calculated as the number of samples reported divided by the total number of samples in the test bed. PCC states that it will only detect Excel-type and Word VB script-type script viruses.

**) The products FSE and AVP do not attempt to detect Macro-Virus written for the desktop environment, as used in the test-beds, they claim to detect Pocket PC relevant malware only. Pocket PC malware does not exist today and is not part of any test-bed, so the products were not tested again against testbed mac_itw.304, after previous tests have shown their inability to detect other platform's malware. Please read the problems.txt from the full report or my diploma thesis for further details.

***)Date shows when the test beds were frozen.

Figure 70: ITW-Macro Virus Detection Rate (samples) on all TestBeds

The two charts (Figure 69: ITW-Macro Virus Detection Rate (malware) on all TestBeds and Figure 70: ITW-Macro Virus Detection Rate (samples) on all TestBeds) compare the detection rates, calculated on malware and on sample basis, of the most recent and the pre-test testbeds.

7.4.1.3 Summary: Macro In-The-Wild

The detection rate for macro viruses of both products INO and PCC is generally not affected if the testbed contains more recent malicious macro samples (itw_mac.dec02).

The grading grid for desktop products as used in the latest aVTC test (see [VTCWEBSITE] or [VTCTEST2002-12] for details) for ITW detection is:

“Concerning "In-The-Wild" viruses, the following grid is applied:

- detection rate is 100% : scanner is "perfect"
- detection rate is >99% : scanner is "excellent"
- detection rate is >95% : scanner is "very good"
- detection rate is >90% : scanner is "good"
- detection rate is <90% : scanner is "risky"

100% detection of In-the-Wild viruses also esp. detecting ALL instantiations of those viruses is now ABSOLUTE REQUIREMENT, for macro and script viruses to be rated "perfect" (it must be observed that detection and identification is not completely reliable)." [VTCTEST2002-12]

This work will only apply the above grading for the detection rates on the most recent testbed (itw_mac.dec02), because in the aVTC test the grading would also only be applied against this testbed.

INO macro grading: "excellent" for itw_mac.dec02

INO reaches an "excellent" for the ITW macro testbed (itw_mac.dec02), as the malware detection rate is 100.00%, but one sample is missed (99.90%).

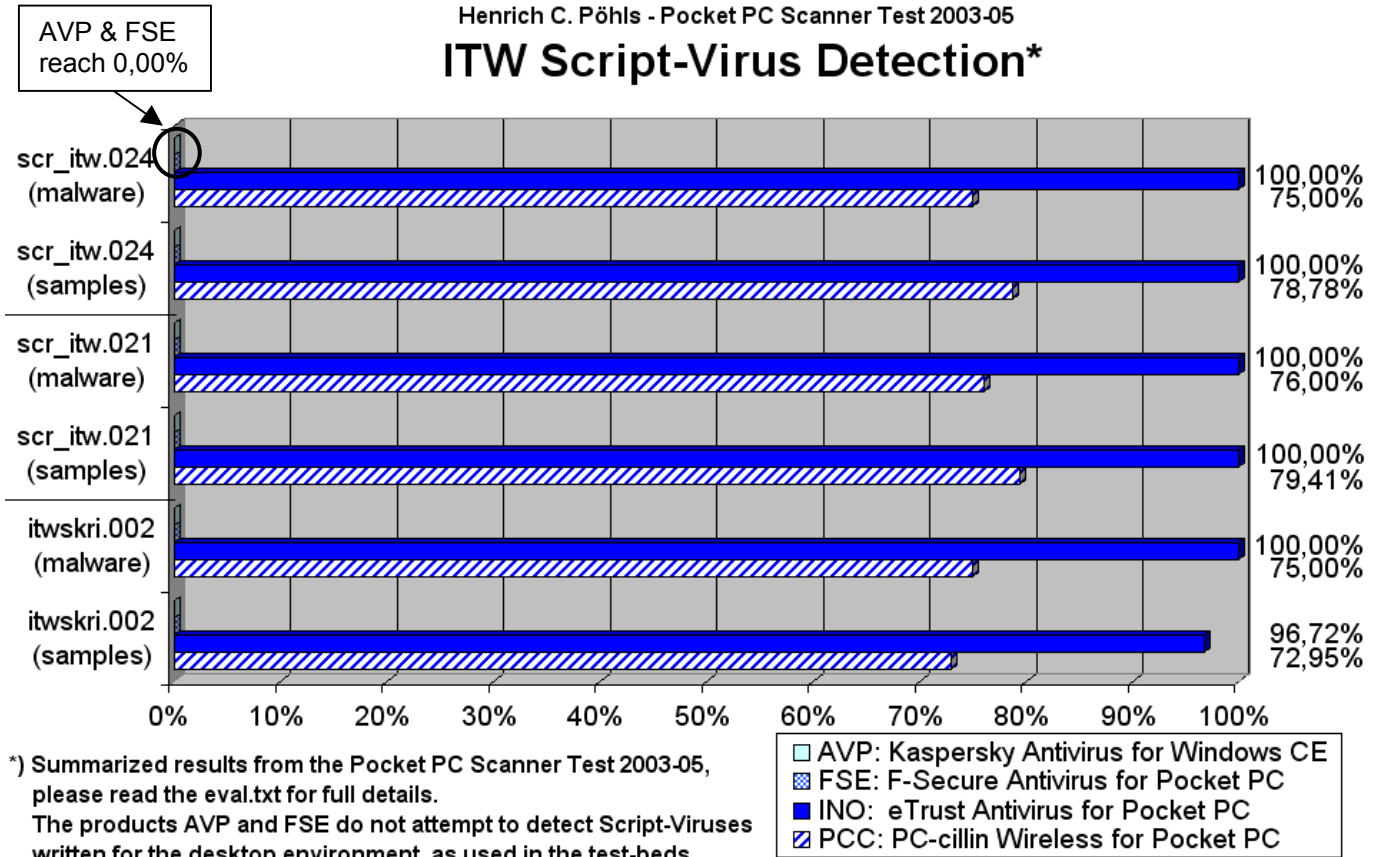
PCC macro grading: "very good" for itw_mac.dec02

PCC will get a grade of "very good" for the recent macro testbeds (itw_mac.dec02 and itw_mac.304), as all detection rates are above 95.00%.

7.4.2 Evaluation of Test Results for Script In-the-Wild Testbeds

7.4.2.1 Pre-Test: Script In-the-Wild

Figure 71 shows the results of the test carried out on the three script testbeds used in the pre-test.



*) Summarized results from the Pocket PC Scanner Test 2003-05, please read the eval.txt for full details. The products AVP and FSE do not attempt to detect Script-Viruses written for the desktop environment, as used in the test-beds, they claim to detect Pocket PC relevant malware only, which is not part of any test-bed used. PCC states that it will only detect Excel-type and Word VB script-type script viruses. Please read the problems.txt from the full report for further details.

Figure 71: Pre-test: ITW Script-Virus Detection Rate (sample and malware based)

AVP & FSE:

The two products AVP and FSE that have not been expected to find any viruses, as they both do not detect non Pocket PC malware, prove their inability by not reporting a single sample. Nevertheless, they do scan all the samples.

INO:

For INO, which in the pre-test's macro results looked perfect, still looks perfect in the detection rate based on malware, where it again reaches 100.00% in all three testbeds. However, this time INO misses four samples from VBS/LoveLetter.A in the itwskri.002 testbed, reaching only 96.72% in the sample detection rate on itwskri.002.

PCC:

PCC is far behind INO in the script detection. The detection rates on all pre-test script testbeds is performs worse, also compared to PCC's own macro results. PCC's detection rates are below 80% and only reach an average malware detection rate of 75.6% on the pre-test testbeds.

7.4.2.2 Final Test: Script In-the-Wild

Using the latest script testbeds, frozen end of December 2002, I conducted a final test, which only included the products INO and PCC, as the two other products proved their inability to detect non Pocket PC malware in the pre-test.

The detailed scanner results are shown in the following tables in the usual aVTC format, containing the actual total numbers and their percentage calculated based on the total number of viruses or samples contained in the testbed.

Results for Script ITW Viruses (Testbed: scr_itw.304, Nov. 2001-Dec. 2002)						
Scanner	Viruses detected		This includes unreliable		Files detected	
			identified	detected		
Testbed	30	100.00%			201	100.00%
INO	30	100.00%	14	46.67%	4	13.33%
PCC	19	63.33%	10	33.33%	19	55.72%

Table 34: Scanner Results for Script ITW Viruses (Testbed: scr_itw.304, Nov. 2001-Dec. 2002)

Results for Script ITW Viruses (Testbed: scr_itw.dec02, December 2002)						
Scanner	Viruses detected		This includes unreliable		Files detected	
			identified	detected		
Testbed	22	100.00%			178	100.00%
INO	22	100.00%	6	27.27%	3	13.64%
PCC	11	50.00%	11	50.00%	7	31.82%

Table 35: Scanner Results for Script ITW Viruses (Testbed: scr_itw.dec02, December 2002)

INO:

This shows that although INO detects all script viruses, it misses 13 samples in four viruses in the scr_itw.304. The samples missed are from the following virus variants: “VBS\Cuerpo.A@MM”, “VBS\Loveletter.A@MM”, “VBS\San.A@MM” and “VBS\Tam.A@M”. Additionally bad is the high unreliable identification rate of 46.67% (scr_itw.304).

Also, in the scr_itw.dec02 testbed, that only contains the December script viruses, INO misses 12 samples in three viruses. The samples missed are from the following virus variants: “VBS\Loveletter.A@MM”, “VBS\SSIWG2.A” and “VBS\Tam.A@M”.

Compared to the pre-test results were INO only missed four samples of “VBS\Loveletter” in one testbeds (itwskri.002), and still got a 96.72% (see Figure 71), this is a decrease. INO still detects 100% of the script viruses in the two recent script testbeds (itw_scr.dec02 and itw_scr.304).

PCC:

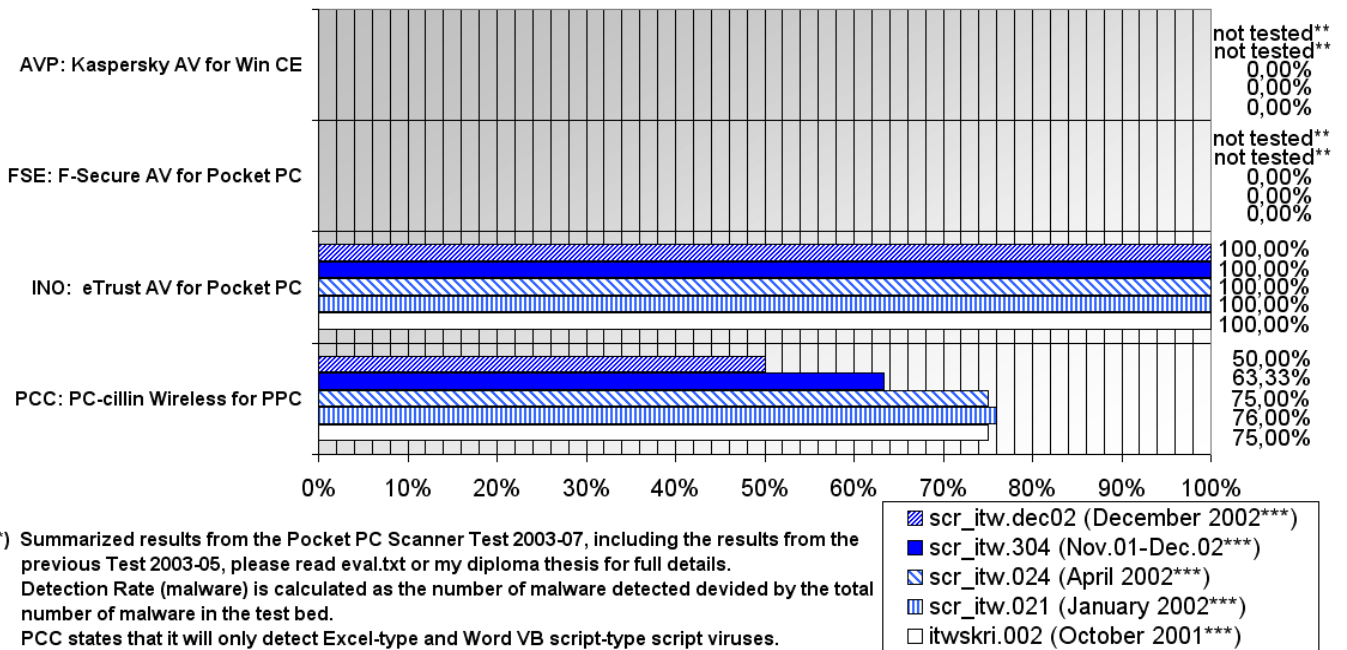
PCC detects roughly only the half of the samples in the recent script testbeds (itw_scr.dec02 and itw_scr.304). Therefore, PCC can inform the user only about 63.33% of the viruses contained in the “long-term” testbed itw_scr.304. It gets worse when only the most recent ITW macro viruses from itw_scr.dec02 testbed shall be detected: PCC only detects half (50.00%) of the December script viruses (itw_scr.dec02).

This detection rate is far from optimal and PCC shows that it has a lower detection rate on most recent script viruses.

Also, for the script detection rate a comparison charts for all testbeds including the ones from pre-test are shown in Figure 72 and Figure 73 on the next page. This leads to the summary of the script evaluation.

Henrich C. Pöhls - Pocket PC Scanner Test 2003-07

ITW-Script Virus Detection Rate (malware) on all TestBeds*

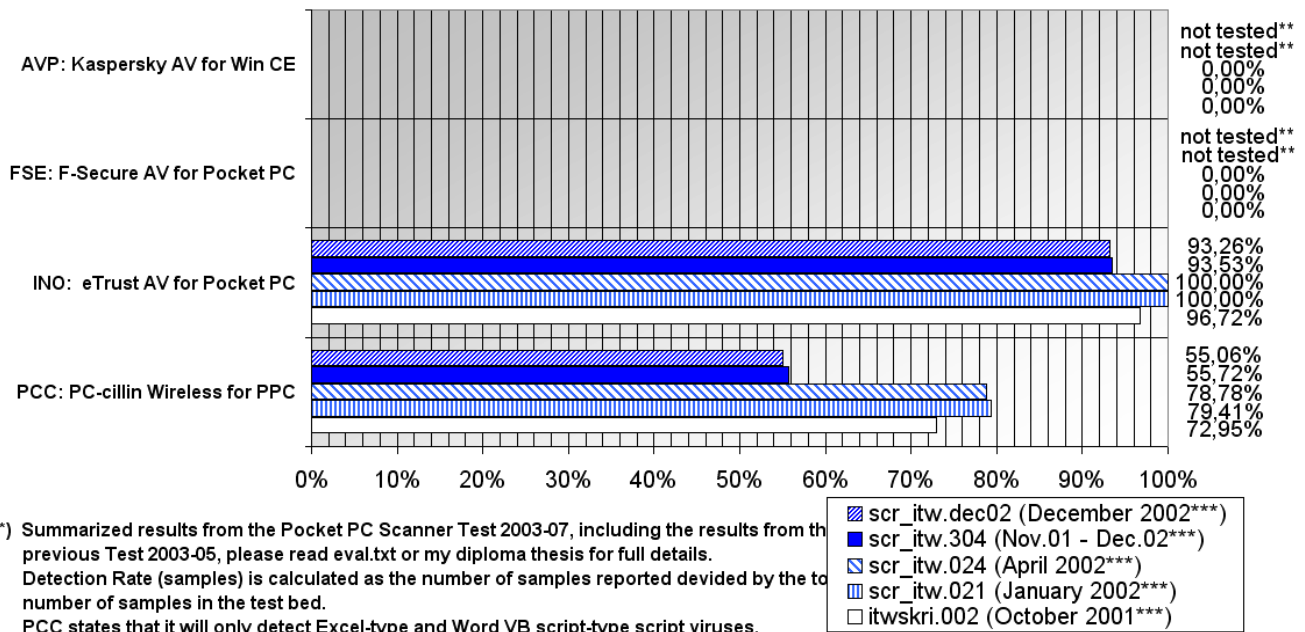


- *) Summarized results from the Pocket PC Scanner Test 2003-07, including the results from the previous Test 2003-05, please read eval.txt or my diploma thesis for full details. Detection Rate (malware) is calculated as the number of malware detected divided by the total number of malware in the test bed. PCC states that it will only detect Excel-type and Word VB script-type script viruses.
- **) The products FSE and AVP do not attempt to detect Macro-Virus written for the desktop environment, as used in the test-beds, they claim to detect Pocket PC relevant malware only. Pocket PC malware does not exist today and is not part of any test-bed, so the products were not tested again against testbed scr_itw.304, after previous tests have shown their inability to detect other platform's malware. Please read the problems.txt from the full report or my diploma thesis for further details.
- ***)Date shows when the test beds were frozen.

Figure 72: ITW-Script Virus Detection Rate (malware) on all TestBeds

Henrich C. Pöhls - Pocket PC Scanner Test 2003-07

ITW-Script Virus Detection Rate (samples) on all TestBeds*



- *) Summarized results from the Pocket PC Scanner Test 2003-07, including the results from the previous Test 2003-05, please read eval.txt or my diploma thesis for full details. Detection Rate (samples) is calculated as the number of samples reported divided by the total number of samples in the test bed. PCC states that it will only detect Excel-type and Word VB script-type script viruses.
- **) The products FSE and AVP do not attempt to detect Macro-Virus written for the desktop environment, as used in the test-beds, they claim to detect Pocket PC relevant malware only. Pocket PC malware does not exist today and is not part of any test-bed, so the products were not tested again against testbed scr_itw.304, after previous tests have shown their inability to detect other platform's malware. Please read the problems.txt from the full report or my diploma thesis for further details.
- ***)Date shows when the test beds were frozen.

Figure 73: ITW-Script Virus Detection Rate (samples) on all TestBeds

7.4.2.3 Summary: Script In-The-Wild

The comparison of all script testbeds shows that while INO again reaches 100% malware detection rate on the most recent testbeds, PCC shows the worst results on the scr_itw.dec02 testbed, with less than two months old viruses.

The detection problems for the December 2002 malware also reduce the detection rate for the long-time ITW testbed (scr_itw.304). This leads to the assumption that PC-cillin Wireless for Pocket PC has not updated the malware information contained in pattern #345 recently enough, as it detects about 10% more viruses each time older viruses are included in testbeds.

On a sample basis INO, though still detecting all script viruses, misses more samples in the most recent testbeds itw_scr.304 and itw_scr.dec02 than in the older testbeds.

Even more dramatically is the decrease in sample detection rates for PCC, which detects far less samples in the most recent testbeds than before on older testbeds.

Again the aVTC grading grid (see [VTCWEBSITE] or [VTCTEST2002-12] for details) can be used to rate the detection for ITW script viruses in the December testbed (itw_mac.dec02):

“Concerning "In-The-Wild" viruses, the following grid is applied:

- detection rate is 100% : scanner is "perfect"
- detection rate is >99% : scanner is "excellent"
- detection rate is >95% : scanner is "very good"
- detection rate is >90% : scanner is "good"
- detection rate is <90% : scanner is "risky"

100% detection of In-the-Wild viruses also esp. detecting ALL instantiations of those viruses is now ABSOLUTE REQUIREMENT, for macro and script viruses to be rated "perfect" (it must be observed that detection and identification is not completely reliable)." [VTCTEST2002-12]

INO grading: "good" for itw_scr.dec02

INO only detects 93.26% of the ITW script virus samples, while it still detects all viruses in the testbeds itw_scr.dec02. This shows that INO has problems identifying the newest samples of script viruses.

PCC grading: "risky" for itw_scr.dec02

PCC has a very bad sample detection rate of less than 60% of all samples from ITW script viruses (itw_scr.dec02). This also results in a bad detection rate on a malware basis of as low as 50.00% for the December 2002 ITW macro viruses (itw_scr.dec02). Therefore, PCC is graded as "risky" for ITW script viruses.

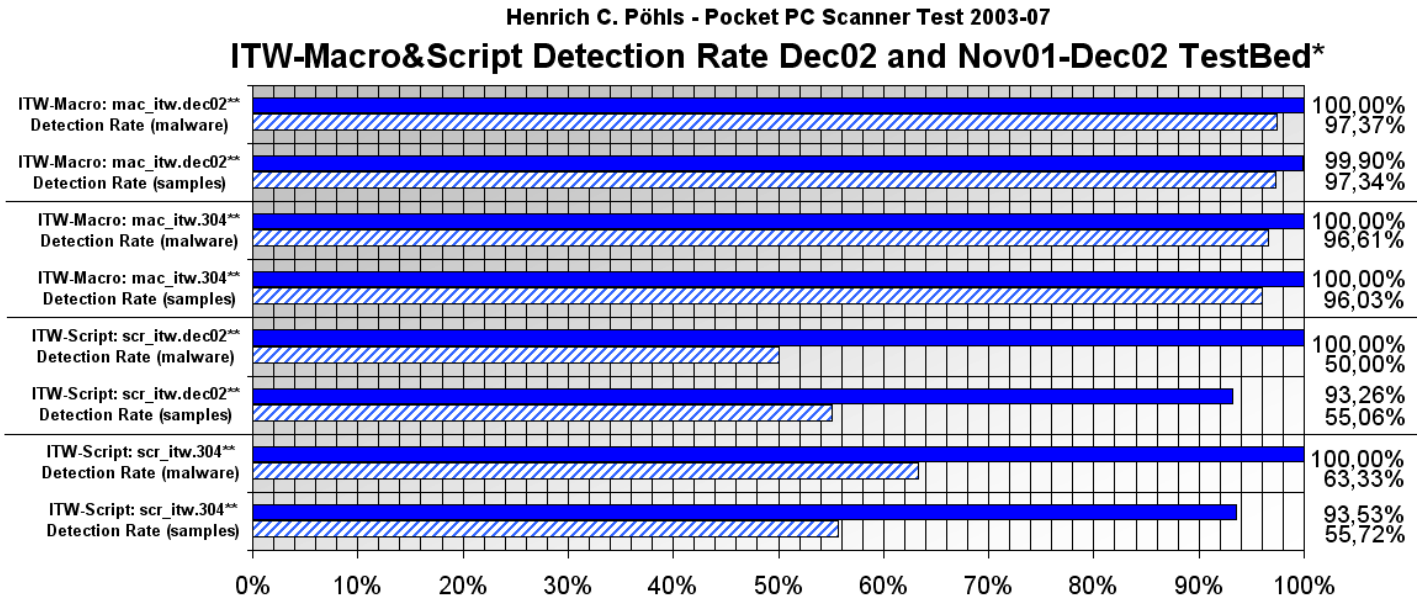
Next I will compare the results of the two Pocket PC anti-malware products INO and PCC to one another and to desktop anti-malware products.

7.4.3 Comparison of Pocket PC Anti-Malware Products and Desktop Products

I will first compare the in-the-wild macro and script detection rates of the two Pocket PC products. Then I will contrast the detection rates of the Pocket PC products with the detection rates of desktop products.

7.4.3.1 Detection Rates of INO vs. Detection Rates of PCC

A direct comparison between the two Pocket PC anti-malware products that were able to detect desktop malware eTrust Antivirus for Pocket PC (INO) and PC-cillin Wireless for Pocket PC clearly shows that INO offers far better detection.



*) Summarized results from the Pocket PC Scanner Test 2003-07, please read eval.txt or my diploma thesis for full details.
 PCC states that it will only detect Excel-type and Word VB script-type script viruses.
 Please read the problems.txt from the full report or my diploma thesis for further details.

■ INO: eTrust Antivirus for Pocket PC

▨ PCC: PC-cillin Wireless for Pocket PC

***) The TestBeds mac_itw.304 and scr_itw.304 were frozen 31st of December 2002.
 Detection Rate (malware) is calculated as the number of malware detected divided by the total number of malware in the test bed.
 Detection Rate (samples) is calculated as the number of samples reported divided by the total number of samples in the test bed.

Figure 74: Head-to-Head Comparison of detection rates from INO & PCC on latest testbeds

The above figure shows the detection rates of INO and PCC in a head-to-head comparison on the two testbeds containing the latest malware. INO detects all of the viruses (100.00%) while PCC does not and detects less in all four testbeds. INO also identifies more samples as malicious than PCC.

7.4.3.2 Detection Rates of Pocket PC Products vs. Detection Rates of Desktop Products

As the two products were tested by the same criteria under which the aVTC tests desktop products, I composed Figure 75 which compares the detection rates of the Pocket PC products with the mean detection rate⁶² of the Windows 2000 desktop products from an earlier aVTC test (Test 2002-12, see [VTCTEST2002-12] for detailed results of that test).

I have chosen to compare the Pocket PC products against the most recent Windows desktop platform that was tested in the aVTC Test 2002-12. So the desktop average is the average detection rate of ITW macro and script viruses from all tested Windows 2000 products, which had a detection rate higher than 10%.

⁶² For the calculation of this average included only products with detection rates >10% are included

There are slight problems with a direct head-to-head comparison:

- The actual version of the Pocket PC scanners have been tested against the old testbeds, but it would be unfair to compare these Pocket PC results (shown in the left half of Figure 75) with the desktop results, as the Pocket PC products have an advantage as they had newer patterns.
- The new December 2002 testbeds, against which none of the desktop products from the Test 2002-12 has been tested, might have introduced very hard to detect viruses.

Nevertheless, even with this restriction the comparison shown in the right half of Figure 75, which have been calculated using the same procedures and periods, allows for a comparison between desktop and Pocket PC products.

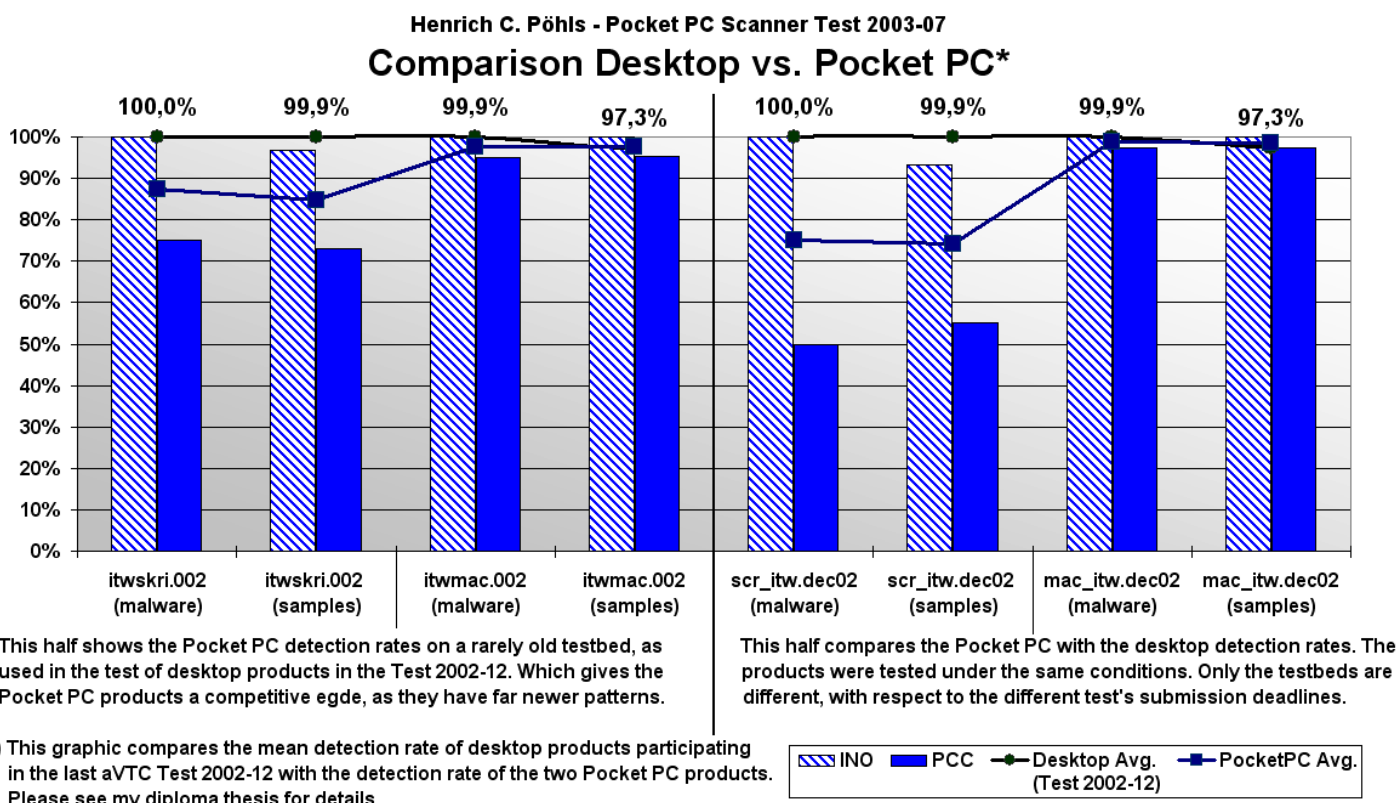


Figure 75 Comparison of the detection rate of Desktop and Pocket PC products

This clearly shows, that INO can compete with the desktop products, although not quite as good as the average Windows 2000 product on the ITW script sample detection. However, PCC performs worse than the average Windows 2000 scanner especially in the script detection. This causes the average script detection rates of the two Pocket PC scanners to be far lower as well.

With Pocket PC being a quite new platform, the Pocket PC scanners do an acceptable good job. This time commercial anti-malware products are available for a platform, which has luckily not seen its first malware yet. This was not the case for DOS or Windows desktop systems. Malware used to be first on different platforms in the past; on the Pocket PC platform the anti-malware programs can be deployed previously. I hope that this lead will help to protect from future Pocket PC malware outbreaks.

7.4.4 Automatic Scan on Removable Storage Media Insertion

Two out of four AV-Products have options to automatically start scanning, once a removable storage medium is inserted into the mobile device. I will refer to this option as “Automatic Scan on Removable Storage Media Insertion”.

The two products having this option are:

- INO: Computer Associates eTrust Antivirus 7.0 for Pocket PC
- FSE: F-Secure Anti-Virus for Pocket PC 1.5

To test if this option works the following files were first transferred to a SD-card, which was later inserted into the mobile test device:

\2577\autorun.exe	This is a harmless renamed Pocket PC application, and it is automatically started on insertion (see 4.3.5.2 and 0. for details).
\eicar test\e1.com	This is an EICAR test file, recognized by both AV-Products, as shown in 7.3.4.5 and 7.3.3.5.
\bitmap.jpg	This is a harmless picture.

Table 36: Auto-Insertion Test Set #1

In a second test set, I tried to simulate a malicious auto-start application, as no Pocket PC malware was available, I renamed the EICAR test file (which is a COM-executable for desktop systems) to `autorun.exe`. It is not only ethically incorrect to write such code, but also the aVTC’s code of conduct [VTCCODE] forbids writing new malware even for test purposes.

This gives the following second test set:

\2577\autorun.exe	This is a renamed EICAR test file; an automatic start on insertion will be tried (see 4.3.5.2 and 0. for details), but will fail, as it is not a Pocket PC executable. So it is an “infected” <code>autorun.exe</code>
\eicar test\e1.com	This is an EICAR test file, recognized by both AV-Products, as shown in 7.3.4.5 and 7.3.3.5.
\bitmap.jpg	This is a harmless picture.

Table 37: Auto-Insertion Test Set #2

The AV-products were set to automatically rename or delete identified malware, in order to allow for an execution prevention.

7.4.4.1 Assumption on the execution of an “infected” `autorun.exe`

I assume that some auto-start on insertion routine, will be executed once the operating system detects the insertion of the removable storage medium. So, the auto-start on insertion routine will open the `autorun.exe` for copying to the `\Windows` directory. The auto-start on insertion routine will then execute the `autorun.exe` from the `\Windows` directory. (See also 4.3.5.2 for details on the auto-start).

If an `autorun.exe` object is found in the `\Windows` directory, even after reports from an AV-product, I will assume that an execution has taken place, and that an “infected” `autorun.exe` has been executed.

If the `autorun.exe` would have been detected and renamed or deleted by an AV-product, before the auto-insertion routine would have searched for it, or copied it, the execution would have been prevented by the AV-product. This prevention would have the observable result, that no `autorun.exe` is present in the `\Windows` folder.

This must be assumed, because the “infected” `autostart.exe` can only be simulated with an EICAR test file. The EICAR test file, recognized by scanners, cannot be observed executing, as it does not execute on the Pocket PC platform.

7.4.4.2 INO: Automatic Scan on Removable Storage Media Insertion

eTrust Antivirus (INO) is configured to delete infected objects and to “Scan storage cards on insert” (see 7.3.4.4).

On the first test set, eTrust Antivirus automatically initiates a scan of the SD-card and deletes the EICAR test file and correctly generates a scan report. However, it shall be noted that the window with the scan report lies below the windows of the automatically executed `autorun.exe` application. The user therefore has no knowledge that the SD-card contains malicious objects, until the user closes the `autorun.exe` application or it ends automatically.

On the second test set, eTrust tries to delete the “infected” `autorun.exe`, but does not succeed and gives an error as depicted in Figure 76. The EICAR test file stored in `\eicar test\el.com` is deleted successfully.



Figure 76: INO gives an error, while deleting the “infected” `autorun.exe`

I have repeatedly inserted the SD-card with the test set #2, but always got the same error message with the same results. I assume that INO is not able to perform the deletion, because the operating system’s auto-run routine still accesses the file.

The “infected” `autorun.exe` is found in the `\Windows` directory. Under the assumption made in 7.4.4.1 INO is not able to prevent the execution of an infected `autorun.exe`, by renaming it before the auto-insertion routine kicks in.

7.4.4.3 FSE: Automatic Scan on Removable Storage Media Insertion

F-Secure Anti-Virus (FSE) is configured to rename infected objects and the option “automatic protection” is set to “Scan inserted storage cards” (see 7.3.3.4).

Generally FSE needs to be started once, to be able to detect insertions. Once started FSE presents the scan report, after the SD-card with test set one has been inserted, on top of the started `autorun.exe`, correctly reporting that the `el.com` EICAR test file has been renamed.

On the second test set, FSE again presents the report to the user, this time correctly reporting two “infected” objects and states that it renamed both. Both “infected” files are indeed renamed successfully.

Looking at the `\Windows` directory still shows an “infected” `autorun.exe`. The execution of an “infected” `autorun.exe` has taken place under the assumption made in 7.4.4.1, and FSE was also not able to prevent the execution.

7.4.4.4 Summary: Automatic Scan on Removable Storage Media Insertion

At a time, where no malware is known that could execute on the Pocket PC platform, the protection offered by FSE and INO is sufficient, as it will scan the objects of an inserted storage medium automatically, once configured to do so. FSE needs to be running in order to successfully achieve this, while INO is started completely automatically.

The option “Automatic Scan on Removable Storage Media Insertion” triggers a scan whenever the event, here the insertion of a removable storage medium, occurs. This could be seen as an “on-event” scan, rather than a manually initiated on-demand scan. This event triggered scanning shall not lead the user to the assumption that he is protected by an always-running on-access scan.

An on-access scan could prohibit the copying and execution of a malicious `autorun.exe`, as already the first attempt to access objects triggers a scan. If this scan identifies the malicious `autorun.exe` as malicious, the access and so also the copying and execution can be prohibited.

Both products do not seem to stop malicious `autorun.exe` applications (tested under assumption 7.4.4.1), and therefore even if they would detect the Pocket PC malware inside the `autorun.exe`, they will not be able to protect against an automatic start from removable storage media. The event triggered scan offered by the products cannot offer such a protection, because the automatic execution is not delayed until the scan process has completely finished. This could only be achieved if Pocket PC products would be able to perform real on-access scanning.

8 Conclusion

This work has shown that the general security functionality of mobile devices based on Pocket PC 2002 is limited. Especially the left out Windows CE 3.0 trust-model (see 4.6.6) greatly reduces the restrictions that can be applied to applications running on the mobile device.

The risk analysis conducted in chapter 6.6 has identified the following areas of high risks:

- virus contamination,
- offline Trojan horse contamination,
- malware distribution.

The following areas were identified as medium risks:

- worm contamination,
- online Trojan horse contamination.

The following area was identified as low risk:

- hostile applet contamination.

The build-in safeguards, as set forth in chapter 7.1 can be applied to make the mobile device more secure and to reduce some of the risk of malware contamination or malware distribution. However, latest with the advent of malware especially targeted at mobile devices, anti-malware products running stand-alone on the Pocket PC shall be used to guard the mobile device against malware. Today's anti-malware products have limitations, and compared to desktop products they need further improvement.

8.1 General Pocket PC Security

The operating system does separate the memory of the different processes running (see 4.3.4), but there is no way to separate stored data in the file system. No notion of access control on the file systems that the Pocket PC supports (see 4.3.5.2), no NTFS that would allow administrators to protect certain files against user access or unwanted modification. Also without the trust-model (see 4.6.6), there is no option for administrators to restrict the applications allowed to run on the Pocket PC 2002 device.

All this limitations in the security functions, used in desktop systems since Windows NT, make it very complicated for administrators to manage mobile devices. Users are given free hand, to do whatever they want once they have access to the device, only specialised security aware applications that would encrypt their data could ensure that they can control at least the read access to that encrypted file, however encrypting a file on an unsecured file system does not protect it against simple deletion or modification.

It is possible for an administrator to protect the device against the execution of unwanted software. With enabling the undocumented policy restrictions, the execution of not already installed software is limited. The administrator must not install third-party file management applications, and no registry editors, the user cannot manually re-rename executables (*.exe and *.cab) or disable the restriction in the registry. Further, the administrator has to completely remove the support for eVB applications, as they can bypass the executable policy. On a mobile device configured as such, the user cannot install new applications.

However, the user can still modify or delete all the files on the mobile device, including operating system and configuration files. All this not only reduces the possibility to completely administer the mobile device and protect against unwanted user actions, but also allows malware to do harm.

8.2 Risk of Malware Contamination and Distribution

The risk of malware distribution is rated as high, especially due to the limitations to detect macros in documents. It is possible for malware, not only document embedded malware, to be distributed through Pocket PC mobile devices.

Malware can also contaminate the mobile device, especially if no safeguards neither the build-in nor the anti-malware products are deployed. Viruses pose a high risk and they can be easily spread over removable storage media. Not only the risk of virus contamination is rated as high, also the risk of contamination with an offline Trojan horse. Trojan horses can cause annoyances or real damage to the user, as a Trojan horse like all malware has unlimited access to all files, once executed on the device.

Worms and online Trojan horses can use the many networking facilities of the mobile device. However, there are limiting factors on the network access, which reduce the risk to medium. And the good news, the risk of contamination with hostile applets is rated as low; due to heavy restrictions by Pocket Internet Explorer (PIE).

8.3 Anti-Malware Products for Pocket PC

This work has conducted the first test of the detection-rate of anti-malware products running stand-alone on Pocket PC 2002 devices. The test yielded interesting results:

Two out of four products running stand-alone on the Pocket PC will not provide any protection against the distribution of desktop malware through Pocket PCs. The two Products AVP and FSE do not detect desktop-based malware. However, AVP and FSE might be able to protect against malware contamination with future Pocket PC malware, as they have proven their ability to scan all the files, though not identifying any of the desktop based malware from the testbeds.

The remaining two products INO and PCC are able to provide also protection against malware distribution, as they are able to detect desktop based malware from the testbeds. So INO and PCC allow to protect mobile devices against future malware contamination and also against malware distribution. From these remaining two products only INO offers a sufficient level of protection and could compete with the average desktop product in the detection of in-the-wild script and macro viruses.

None of the products has on-access scanning capabilities. On-access scanners are deployed to desktop computers to prevent the execution of malicious code on the desktop computer, as they prevent the access to infected objects before they are invoked. The protection offered by on-access scanners can not be offered by the mechanisms included in the Pocket PC anti-malware products, as a test of the "Automatic Scan on Removable Storage Media Insertion" mechanism offered by two products has shown.

The tests have shown that it is possible to build sufficient anti-malware products also on the Pocket PC platform. It must be further noted that although no Pocket PC malware is known today, there are already four commercial products waiting to use new pattern updates to detect the first Pocket PC malware.

This could be the greatest advantage that anti-malware products ever had. However, they will only be able to keep this advantage, if a large number of Pocket PC users install such anti-malware software and if they manage to keep their patterns up-to-date.

The two products FSE and INO offer a pattern update function that can be used from the mobile device, if it has an Internet connection. AVP also offers a pattern update function, but the pattern is only updated when the mobile device is in the cradled state. Only the remaining of the four products, PCC, needs manual pattern updates. The user is advised to regularly update these patterns; no matter how complicated it might be, as they are important for the detection of malware.

Also the very different marketing concepts behind the four anti-malware products will have an impact on how many mobile devices will be equipped with such anti-malware software in the future. The four tested products show various marketing concepts: One product can be downloaded free, two products can be bought as stand-alone versions, and one product is only available as part of a complete product suite. This work has not further investigated the costs of the different anti-malware products, but this already shows that different vendors see the Pocket PC platform with different eyes, when it comes to the prices and marketing.

None of the tested software showed adverse effects on the functionality, speed or stability of the mobile device, so I suggest it is better to have a ready and waiting anti-malware software installed, that regularly updates its pattern than to foolishly wait until it is too late and the first malware is in-the-wild.

8.4 Suggestions for Future Work

The mobile devices that today are more gadgets than mobile clients, will take over client functionality in the future. If mobile devices will increasingly act as mobile clients, they must also offer the security of desktop clients. To achieve desktop client security, the security functionality must be increased in future mobile devices. For example the development of an installable file system (IFS) that supports access control features as NTFS, would greatly enhance the security of mobile devices.

Also further research into the next version of Pocket PC 2002, Windows Mobile 2003 for Pocket PC should be done to show if the highlighted security problems could be fixed using the more advanced Windows CE.net operating system components.

Security relevant products like anti-malware products must be tested like their desktop counterparts. Ongoing tests, like the aVTC tests of desktop anti-malware products shall provide users with information on the detection rate, also for products running on mobile devices. Especially the future of mobile device specific malware must be clearly observed, to deploy countermeasures fast enough.

Mobile devices will be exposed to more security problems known from desktop systems in the future. If for example mobile network connectivity is increasing (i.e. costs come down, coverage and bandwidth go up), network threats (i.e. spoofing, sniffing or denial of service) will become a threat to mobile devices as well. However, mobile devices will also introduce new security problems due to their mobility. The known security problems need to be solved quickly, before new mobile concepts like "nomadic computing" (see [KLEINROCK1995] for a definition and details) can be implemented securely. So future mobile devices shall be built on the security fundamentals already established and implemented for today's desktop computers and then be enhanced to cater for new "mobile threads".

Bibliography

- [A600MANUAL] Asus:
"MyPal A600 User's Manual"
1st Edition, July 2002

[ftp://ftp.asus.com.tw/pub/ASUS/IA/MyPal A600/
E1054 MyPal A600 English User Guide.pdf](ftp://ftp.asus.com.tw/pub/ASUS/IA/MyPal_A600/E1054_MyPal_A600_English_User_Guide.pdf)
Last download: August 2003
- [ALFORQUE2000] Maricia Alforque:
"Creating a Secure Windows CE Device"
October 2000

<http://msdn.microsoft.com/library/en-us/dnce30/html/winsecurity.asp>
Last downloaded: May 2003
- [ANDERSON2001] Tim Anderson:
"Tools in your pocket"
2001, in DNJ Online issue 25

http://www.dnjonline.com/articles/mobility/iss25_mobility_tools.asp
Last download: April 2003
- [ARMCOREWEB] ARM Ltd:
"Core Selection Guide"

http://www.arm.com/armtech/core_selection
Last download: February 2003
- [ARMREFERENCE] ARM Ltd:
"ARM Architecture Reference Manual"

Addison-Wesley, Harlow, 2000
ISBN 0-201-73719-1
- [ARMTRUST] ARM Ltd:
"TrustZone Technology"

<http://www.arm.com/armtech/TrustZone>
Last downloaded: August 2003
- [ARMTRUSTPR] ARM Ltd (Pressrelease):
"ARM Builds Security Foundation For Future Wireless And Consumer Devices"
May 27, 2003

<http://www.arm.com/armtech/TrustZone>
Last downloaded: August 2003
- [ARMWEB] ARM Ltd:
"The ARM Architecture"

http://www.arm.com/armtech/ARM_Arch
Last download: February 2003
- [ARTISTAWEB] Bernina:
"Artista 200"

<http://media.berninausa.com/index.php/artista>
Last download: January 2003

- [AVPMANUAL] Kaspersky Lab:
"Kaspersky Security for PDA 4.5 - USER GUIDE"
Revision date: December 2002

http://www.avp.com.mx/descargas/4.0/doc/kav4.5_securitypdae.pdf
Last downloaded: March 2003
- [AVPPRESSWEB] Kaspersky Lab (Pressrelease):
"Kaspersky Security for PDA - A New Level Of Protection For Handheld Computers"
March 04,2003

<http://www.kaspersky.com/news.html?id=972594>
Last downloaded: March 2003
- [AYALA2002] Jordan Ayala:
"Ultimate Wireless Email"
June, 2002 in Windows & .NET Magazine

<http://www.mobile-and-wireless.com/Articles/Index.cfm?ArticleID=24875>
Last download: April 2003
- [BDFDOWNLOAD] BitDefender:
BitDefender for Pocket PC (Arm) Produkt-Download

http://www.bitdefender.com/download/download.php?file=bdwce_pocketpc_arm.zip
Last download: February 2003
- [BDFMANUAL] BitDefender (Manual):
"BitDefender for Windows CE"
- [BDFREADME] BitDefender (ReadMe):
"BitDefender for Windows CE version 1.01"
- [BLACKBERRYWEB] Research In Motion Limited:
Website

<http://www.blackberry.net>
Last download: January 2003
- [BOLING1999] Douglas Boling:
"Updated with New Kernel Features, Windows CE 3.0 Really Packs a Punch"
in Microsoft Systems Jornal, July 1999

<http://www.microsoft.com/msj/0799/wincekernel/wincekerneltop.htm>
Last download: March 2003
- [BOLING1999-2] Douglas Boling:
"Minimizing the Memory Footprint of Your Windows CE-based Program"
in Microsoft Systems Jornal, May 1999

<http://www.microsoft.com/msj/0598/memory.aspx>
Last download: March 2003
- [BOLING2002] Douglas Boling:
"Windows CE .NET Advanced Memory Management"
August 2002

<http://msdn.microsoft.com/library/en-us/wcedsn40/html/cgconusingramrommassstorage.asp>
Last download: April 2003

- [BRAGINSKI2000] Leonid Braginski & Matthew Powell:
"Windows CE Web Server: Using Web Tools to Monitor and Manage Embedded Devices"
in MSDN Magazine, Issue: May 2000

msdn.microsoft.com/msdnmag/issues/0500/wince/default.aspx
Last download: January 2003
- [BRASH2002] David Brash:
"The ARM Architecture Version 6 (ARMv6)",
January 2002, ARM WhitePaper

[http://www.arm.com/support/56VF7H/\\$File/ARMv6_Architecture.pdf](http://www.arm.com/support/56VF7H/$File/ARMv6_Architecture.pdf)
Last download: January 2003
- [BRUNNSTEIN1999] Klaus Brunnstein:
"From AntiVirus to AntiMalware Software and Beyond: Another Approach to the Protection of Customers from Dysfunctional System Behaviour"
July 23, 1999 in 22nd National Information Systems Security Conference Proceedings

<http://csrc.nist.gov/nissc/1999/proceeding/papers/p12.pdf>
Last download: March 2003
- [BUGTRAQWEBSVR] Bugtraq Posting:
"ZH2003-5SA (security advisory): Windows beta webserver for pocket pc: full remote access."
on BUGTRAQ August 2, 2003

<http://www.securityfocus.com/archive/1/331735>
Last downloaded: August 2003
- [CANALYS2002] canalys.com (Pressrelease) :
"Palm retains mobile lead by units, Compaq top by value"
April 26, 2002

<http://www.canalys.com/pr/r2002041.htm>
Last download: January 2003
- [CAROVNC1991] A. S. Fridrik Skulason and Vesselin Bontchev:
"A New Virus Naming Convention"
CAROmeeting 1991

<http://downloads.securityfocus.com/library/naming.txt>
Last download: December 2002
- [CASEDY2001] Krista Casedy, Matthew Gunderson, Nicholas Idler, Jessica Lee:
"Protecting Against the Emerging Threat of PDA Viruses"
May 2001

<http://198.11.21.25/capstoneTest/Students/Papers/docs/proceedingspaper39198.pdf>
Last download: January 2003
- [CEGADGETS] cegadgets.com:
"Windows CE Developers FAQ"
March 27, 2001

<http://www.cegadgets.com/wincedevfaq.htm>
Last downloaded: June 2003

- [CERTICOM] Certicom (Pressrelease):
"Certicom Launches Industry's First Secure E-mail Solution for Microsoft Pocket Outlook"
March 17, 2003
- http://www.certicom.com/about/pr/03/030317_mmail.html
Last download: March 2003
- [CHIEN2000] Eric Chien:
"Malicious Threats to Personal Digital Assistants"
October 2000
- <http://securityresponse.symantec.com/avcenter/reference/malicious.threats.to.pdas.pdf>
Last download: January 2003
- [CLINICK2000] Andrew Clinick:
"Is That a Script in Your Pocket?"
August 14, 2000
- <http://msdn.microsoft.com/library/en-us/dnclinic/html/scripting08142000.asp>
Last download: April 2003
- [CODY2001] John Cody:
"Creating POOM items using PIE Web pages"
on dEVBuzz in November 2001
- http://www.devbuzz.com/content/zinc_evb_creating_POOM_items_PIE_pg2.asp
Last downloaded: April 2003
- [COMPAQSECENH] Compaq:
"Security Enhancements for Pocket PC"
- <http://compaq.co.uk/globalservices/security>
Last downloaded: Nov 2002
- [DAVIS2003] Andy Davis:
"ActiveSync Version 3.5 Denial of Service Vulnerability"
on BUGTRAQ March 21, 2003
- <http://cert.uni-stuttgart.de/archive/bugtraq/2003/03/msg00304.html>
Last downloaded: May 2003
- [DEDO2001] Douglas Dedo:
"Mobile Enterprise Solutions: What Is the Appropriate Pocket-Size Platform"
October, 2001
- <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/mobile/evaluate/mobilwhy.asp>
Last download: November 2002
- [DEDO2002] Douglas Dedo:
"Pocket PC Security"
May 2002
- <http://www.microsoft.com/mobile/enterprise/papers/security.asp>
Last download: November 2002
- [DEVTOOLS] Microsoft (Whitepaper):
"Development Tools for Mobile and Embedded Applications"
- <http://www.microsoft.com/mobile/enterprise/papers/devtoolsmobileapps.doc>
Last download: November 2002

- [DUDENINFORMATIK] Hermann Engesser, Volker Claus, Andreas Schwill :
"Duden Informatik - ein Sachlexikon für Studium und Praxis - 2. Ausgabe"

Dudenverlag Mannheim, Leipzig, Wien, Zürich, 1993
ISBN: 3-411-05232-5
- [ETFORECAST2002] eTForecasts (Pressrelease):
"Pocket PC PDAs To Surpass Palm OS PDAs in 2004"
September 9, 2002

<http://www.etforecasts.com/pr/pr0902.htm>
Last download: January 2003
- [EVT3] Microsoft:
eMbedded Visual Tools 3.0

from CD ROM, freely downloadable also from:
www.microsoft.com/mobile/downloads/emvt30.asp
- [EVT3HELP] Microsoft:
Helpfile of eMbedded Visual Tools Version 3.0 with Pocket PC 2002 SDK
- [FELDMAN2003] Assaf Feldman:
"Macromedia Flash Integration with Pocket PC E-mail and Contact List"
March 2003

http://www.macromedia.com/devnet/mobile/articles/ppc_email.html
Last download: June 2003
- [FREITAG2000] Sönke Freitag:
"Webbasiertes Auffinden maliziöser Software mit fortschrittlichen heuristischen Verfahren (MWC - Malware Crawler)"
July 2000

http://agn-www.informatik.uni-hamburg.de/papers/doc/diparb_soenke_freitag.pdf
Last download: July 2003
- [FSELIBERTY] F-Secure Virus Descriptions:
"Palm/Liberty"

http://www.europe.f-secure.com/v-descs/lib_palm.shtml
Last downloaded: August 2003
- [FSEMANUAL] F-Secure:
"Anti-Virus for Pocket PC User's Guide"
Version 1.50

http://www.europe.f-secure.com/webclub/hh/av/ppc15/enterprise/avppc_eng.pdf
Last download: February 2003
- [FSERELNOTES] F-Secure:
"Release Notes for F-Secure Anti-Virus for Pocket PC Version 1.50"

<http://www.europe.f-secure.com/webclub/hh/av/ppc15/enterprise/readme.rtf>
Last download: February 2003
- [FSESLAPPER] F-Secure:
"Global Slapper Worm Information Center"
September 26, 2002

<http://www.f-secure.com/slapper>
Last download: August 2003

- [FSESTAOG] F-Secure Virus Descriptions:
"Linux/Staog"

<http://www.europe.f-secure.com/v-descs/staog.shtml>
Last downloaded: August 2003
- [GARTNER2002] Gartner (Pressrelease):
"Gartner Dataquest Says Hewlett-Packard Surpassed Palm in Worldwide PDA Revenue in Second Quarter of 2002"
August 05, 2002

http://www4.gartner.com/5_about/press_releases/2002_08/pr20020805b.jsp
Last download: January 2003
- [GARTNER2003] Gartner Dataquest (Pressrelease):
"Gartner Dataquest Says Worldwide PDA Market Suffers Through a Dismal Year in 2002"
January 27, 2003

http://www3.gartner.com/5_about/press_releases/pr27jan2003a.jsp
Last download: January 2003
- [GESYTEC] Gesytec:
"Developer's Guide Windows CE Embedded PC"

<http://www.gesytec.de/common/pdf-downloads/epc/embedded-pc.pdf>
Last download: October 2002
- [GOLEMNEWS2002] golem.de (News):
"Weltweiter PDA-Markt geht im 3. Quartal 2002 zurück"
November 01, 2002

<http://dyn1.golem.de/cgi-bin/usisapi.dll/forprint?id=22438>
Last download: January 2003
- [HEIMANN2003] Stefan Heimann:
"Analyse und Weiterentwicklung der Testbed-Erstellung im aVTC-Projekt"
April 2003

http://agn-www.informatik.uni-hamburg.de/papers/doc/bacarb_stefan_Heimann.zip
Last downloaded: June 2003
- [HERRERA2001] Chris De Herrera:
"Virus FAQ"
Version 1.00 Revised 7/17/2001

<http://www.cewindows.net/faqs/virus.htm>
Last download: January 2003
- [HERRERA2001RAM] Chris De Herrera:
"RAM FAQ"
Version 1.00 Revised 6/17/2001

<http://www.cewindows.net/faqs/ram.htm>
Last download: April 2003
- [HERRERA2002SEC] Chris De Herrera:
"Pocket PC Security"
Version 1.01 Revised 2/5/2002

<http://www.cewindows.net/reviews/pocketpc2002security.htm>
Last download: Jan 2003

- [HERRERA2002SECFAQ] Chris De Herrera:
"Pocket PC Security" [FAQ]
Version 1.00 Revised 11/14/2002

<http://www.cewindows.net/faqs/ppc-security.htm>
Last download: Jan 2003
- [HP1910FAQ] HP:
"iPAQ model h1910 FAQs"

http://h21007.www2.hp.com/dspp/files/unprotected/PDA/dsppwireless/WCSA_1910_FAQs.html
Last download: April 2003
- [HP3800SPECS] HP:
"QuickSpecs Compaq iPAQ Pocket PC H3800 Series"
Version 17, August 16, 2002

http://h18000.www1.hp.com/products/quickspecs/10977_na/10977_na.html
Last download: April 2003
- [HPFAQRESET] HP (Discussion Group):
"How to full reset a ipaq 5450"
Thread started: January 23, 2003

<http://bizforums.itrc.hp.com/cm/QuestionAnswer/1,,0x73175bd3782dd711abdc0090277a778c,00.html>
Last downloaded: April 2003
- [HPFAQWSYNC] HP (Discussion Group):
"ActiveSync over WLAN"
Thread started: January 28, 2003

<http://bizforums.itrc.hp.com/cm/QuestionAnswer/1,,0x03255bd3782dd711abdc0090277a778c,00.html>
Last downloaded: April 2003
- [HPPRESS2001] HP (Pressrelease):
"HP Unveils New HP Jornada 560 Series Pocket PCs That Offer Limitless Expandability For Wireless Connectivity and Solutions"
September 27, 2001

http://www.hp.com.hk/pcorner/press/20010927_e.htm
Last download: March 2003
- [IBM405LP] IBM:
"PowerPC 405LP Embedded Processor"

http://www-3.ibm.com/chips/techlib/techlib.nsf/products/PowerPC_405LP_Embedded_Processor
Last download: February 2003
- [INOMANUAL] Computer Associates:
"eTrust Antivirus for Pocket PC 2002 - User Guide"
Version 2.0

ftp://ftp.ca.com/pub/Inoculan/docs/eAV_PPC_User_Guide.pdf
Last download: April 2003

- [INTELMEM] intel:
"Microsoft Windows CE Memory Models and Usage"

<http://www.intel.com/design/flcomp/manuals/psm/appendb.pdf>
Last download: September 2002
- [INTELSA1110DEV] intel:
"Intel StrongARM SA-1110 Development Board Hardware Release Notes"*
June 16, 2000

<http://www.intel.com/design/strong/schems/HWRELNOT.pdf>
Last download: February 2003
- [INTELSA1110WEB] intel:
"Intel SA-1110 processor"

http://developer.intel.com/design/pca/applicationsprocessors/1110_brf.htm
Last download: February 2003
- [INTELWLANCE] intel:
"Intel PRO Wireless 2111 LAN PC Card - Windows CE Driver Installation Instructions"

ftp://download.intel.com/support/network/wireless/pro2011/lanpccard/cedriver_install.pdf
Last download: January 2003
- [IRCOMM] Infrared Data Association:
"IrCOMM: Serial and Parallel Port Emulation over IR (Wire Replacement)"
Version 1.0 - 7 November, 1995

<http://www.irda.org/standards/pubs/ircomm10.pdf>
Last download: March 2003
- [ISO13335-1] ISO:
"Information technology -- Guidelines for the management of IT Security -- Part 1: Concepts and models for IT Security"
Draft, May 05, 1999
- [ITASKWEB] iTask:
Website

<http://www.itask.com>
Last download: October 2002
- [ITREVIEW] IT review:
"HP - Jornada 928 WDA review"

<http://www.itreviews.co.uk/hardware/h343.htm>
Last download: March 2003
- [JAVAAPPLET] SUN:
"Security for the Java Platform"

<http://www.sun.com/960901/feature3/javasecure.html>
Last downloaded: June 2003

- [JAVAVM] INSIGNIA (Pressrelease):
"Insignia Ships First Java Virtual Machine Integrated Into Microsoft's
Pocket Internet Explorer"
May 2001
- <http://www.insignia.com/content/about/releases/010531.shtml>
Last downloaded: April 2003
- [JOHANNSEN1998] Lars Johannsen und Kai Müller:
"Kommunikation zwischen PDAs über IrDA"
May, 1998
- <http://www.teco.uni-karlsruhe.de/~markus/dasa/irda/index.htm>
Last downloaded: March 2003
- [JORNADAPR] Leonid Braginski & Matthew Powell:
"Windows CE Web Server: Using Web Tools to Monitor
and Manage Embedded Devices"
in MSDN Magazine, Issue: May 2000
- <http://msdn.microsoft.com/msdnmag/issues/0500/wince/default.aspx>
Last download: January 2003
- [KEUCHEL2001] Rainer Keuchel:
"Windows CE Desktop Toolset"
January 30, 2001
- <http://www.rainer-keuchel.de/wince/wince-desktop-tools.tar.gz>
Last downloaded: June 2003
- [KLEINROCK1995] L. Kleinrock:
"Nomadcity: Characteristics, Issues, and Applications"
Nomadic Working Team of the Cross Industrial Working Team, 1995
- <http://www.xiwt.org/documents/Nomadcity.html>
Last download: August 2003
- [KRELL2002] Bruce E. Krell:
"Pocket PC Developer's Guide"
- McGraw-Hill/Osborne, Berkely, 2002
ISBN: 0-07-213150-0
- [LANGE2001] Ludovic Lange:
"Re: Reverse Engeneering Active Sync"
March 24, 2001
- <http://handhelds.org/hypermail/handhelds/15/1550.html>
Last download: July 2003
- [LUTTERBECK2003] S. Balszuweit, T. Fritsch, R. Gehring, T. Kamp, R. Leiteritz (Projektleiter),
Prof. Dr. iur. Bernd Lutterbeck, F. Pallas, T. Pehl ,N. Yildiz:
"MOBILER ZUGANG ZU GESICHERTEN NETZEN –
LÖSUNGEN FÜR DIE ZUKUNFT"
Pressversion of a study for the BMI, August 2003
- http://ig.cs.tu-berlin.de/forschung/Mobile/Presseversion_BMI-Studie_mobile_Endgeraete_TU_Berlin.pdf
Last download: August 2003

- [MCPHERSON2002] Frank McPherson:
"How to do everything with your PocketPC - Second Edition"

McGraw-Hill/Osborne, Berkeley, 2002
ISBN: 0-07-219414-6
- [MDAMANUAL] T-Mobile:
"Pocket PC Phone Edition Handbuch"
- [MEUNIER2002] Pascal Meunier, Sofie Nystrom, Seny Kamara, Scott Yost,
Kyle Alexander, Dan Noland, Jared Crane:
*"CERIAS Tech Report 2002-17 - ActiveSync, TCP/IP and 802.11b
Wireless Vulnerabilities of WinCE-based PDAs"*
October 12, 2002

[https://www.cerias.purdue.edu/tools_and_resources/
bibtex_archive/archive/2002-17.pdf](https://www.cerias.purdue.edu/tools_and_resources/bibtex_archive/archive/2002-17.pdf)
Last download: April 2003

Also see an earlier report [https://www.cerias.purdue.edu/
papers/archive/2001-04.pdf](https://www.cerias.purdue.edu/papers/archive/2001-04.pdf) for further details.
- [MOBILEVILLAGENEWS] mobilevillage.com (News):
*"Worldwide Handheld Shipments Decline in 2Q; Pocket PC to
Surpass Palm PDAs in 2004"*
September 11, 2002

<http://www.mobilevillage.com/news/2002.09.12/PDAmarket.htm>
Last download: January 2003
- [MSACTIVESYNC] Microsoft Development Network (MSDN):
"ActiveSync Security"
April 22, 2003

[http://msdn.microsoft.com/library/en-us/wceactsy/html/
ceconActiveSyncSecurity.asp](http://msdn.microsoft.com/library/en-us/wceactsy/html/ceconActiveSyncSecurity.asp)
Last download: May 2003
- [MSACTIVEXSECURITY] Microsoft Development Network (MSDN):
"Safe Initialization and Scripting for ActiveX Controls"

<http://msdn.microsoft.com/workshop/components/activex/safety.asp>
Last download: April 2003
- [MSAUTHENTICCODE] Microsoft Development Network (MSDN):
"Frequently Asked Questions About Authenticode"
May 26, 2000

<http://msdn.microsoft.com/library/en-us/dnauth/html/signfaq.asp>
Last downloaded: June 2003
- [MSCE1] Microsoft (Pressrelease):
"Microsoft Announces Broad Availability of Handheld PCs With Windows CE"
November 1996

<http://www.microsoft.com/presspass/press/1996/Nov96/wincepr.asp>
Last downloaded: May 2003

- [MSCE2] Microsoft (Pressrelease):
"Microsoft Announces Release of Windows CE 2.0"
Septemeber 1997

<http://www.microsoft.com/presspass/press/1997/Sept97/WINCE2PR.asp>
Last downloaded: May 2003
- [MSCE3] Microsoft (Pressrelease):
"Microsoft Announces Availability of Windows CE 3.0"
June 2000

<http://www.microsoft.com/presspass/press/2000/Jun00/CELlaunchPR.asp>
Last download: May 2003
- [MSCE4] Microsoft (Pressrelease):
"Microsoft Launches Windows CE .NET"
January 2002

<http://www.microsoft.com/presspass/press/2002/jan02/01-07CENetLaunchPR.asp>
Last download: May 2003
- [MSSEARCHITECTURE] Microsoft:
"Windows CE Architecture in Depth"
October 2001

www.microsoft.com/mobile/assets/Windows_CE_Architecture_in_Depth.doc
Last download: April 2003
- [MSCECOMPARISON] Microsoft:
"Comparing Windows CE 3.0 with Windows CE 2.12"
June 2000

<http://www.microsoft.com/windows/embedded/ce.net/previous/evaluation/compare/ce212v30.asp>
Last download: May 2003
- [MSCECOMPNET] Microsoft:
"Feature by Feature Comparison: Microsoft® Windows® CE .NET 4.2, 4.1, 4.0 and Windows CE 3.0"
April 2003

http://www.microsoft.com/windows/embedded/docs/ce.net/FeatComp_WindowsCE4.2.doc
Last download: May 2003
- [MSCEDOTNET] Microsoft:
"Windows CE.Net Homepage"

<http://www.microsoft.com/windows/embedded/ce.net/default.asp>
Last download: November 2002
- [MSCEOSI] Microsoft Development Network (MSDN):
"Microsoft Windows CE 3.0 - Open Systems Interconnection Model"

http://msdn.microsoft.com/library/en-us/wcecomm/html/_wcesdk_Open_Systems_Interconnection_Model.asp
Last download: August 2003

- [MSCEPROCESSORS] Microsoft:
"Supported Processors"
September 26, 2002

<http://www.microsoft.com/windows/Embedded/ce.NET/evaluation/hardware/processors.asp>
Last download: February 2003
- [MSCEQOS] Microsoft Development Network (MSDN):
"How Windows CE .NET is Designed for Quality of Service"
February 2003

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncenet/html/cequalityservice.asp>
Last downloaded: April 2003
- [MSCERDP] Microsoft Development Network (MSDN):
"Remote Desktop Protocol in Windows CE"
December 2000

<http://msdn.microsoft.com/library/library/en-us/dnce30/html/rdp.asp>
Last download: June 2003
- [MSCHAPV2] Microsoft:
"MS-CHAP version 2"
February 28, 2000

http://www.microsoft.com/windows2000/en/server/help/sag_RASS_MSCHAPv2.htm
Last downloaded: May 2003
- [MSCSP] Microsoft Development Network (MSDN):
"Microsoft Cryptographic Service Providers"

http://msdn.microsoft.com/library/en-us/security/security/microsoft_cryptographic_service_providers.asp
Last download: May 2003
- [MSDEVCON01] Marcus Perryman:
"Programming for Native Applications"
February 27, 2002
Presentation at The Microsoft Mobility Developer Conference

<http://jungla.dit.upm.es/~hvelayos/mdc/Clien8.ppt>
Last download: March 2003
- [MSDISCONN] Microsoft:
"Windows CE Applications in a Disconnected Environment"

http://www.microsoft.com/mobile/developer/training/Training_modules/Windows_CE_Applications_Disconnected_Environment.doc
Last download: May 2003
- [MSDNCHAT2002] Microsoft Development Network (MSDN):
"Chat with the Pocket PC Development Team"
September 27, 2002

http://msdn.microsoft.com/chats/embedded/embedded_092602.asp
Last download: April 2003

- [MSDNCHAT2003] Microsoft Development Network (MSDN):
"Chat on the Topic Windows CE.NET"
March 18, 2003

http://msdn.microsoft.com/chats/embedded/embedded_031803.asp
Last download: August 2003
- [MSDNCOM] Microsoft Development Network (MSDN):
"MSDN: Platform SDK: COM: Schannel"
Release: February 2003

http://msdn.microsoft.com/library/en-us/com/html/security_7bcc.asp
Last download: March 2003
- [MSDNFAQ] Microsoft Development Network (MSDN):
"Windows CE and Pocket PC: FAQ"
April 2001

<http://msdn.microsoft.com/library/en-us/dnce30/html/ppcfaq.asp>
Last download: November 2002
- [MSDNTAPNHOLD] Microsoft Development Network (MSDN):
"Tap-and-Hold Confirmation in eMbedded Visual Basic"

http://msdn.microsoft.com/library/en-us/dnppc2k2/html/ppc_olevb.asp
Last downloaded: March 2003
- [MSDNWEB] Microsoft Development Network (MSDN):
Website

<http://msdn.microsoft.com>
Last download: September 2002
- [MSFILEFILTER] Microsoft Development Network (MSDN):
"Transferring Files"
January 08, 2003

http://msdn.microsoft.com/library/en-us/wceconct/html/_wcesdk_Transferring_Files.asp
Last download: July 2003
- [MSHHPCFAQ] Microsoft:
"Microsoft Handheld PC. FAQ"
October 31, 2002

<http://www.microsoft.com/mobile/handheldpc/faq/default.asp>
Last download: November 2002
- [MSJSCRIPTVERSION] Microsoft Development Network (MSDN):
"JScript - Version Information"

<http://msdn.microsoft.com/library/en-us/script56/html/js56jsoriVersionInformation.asp>
Last download: April 2003
- [MSPIESESECURITY] Microsoft Development Network (MSDN):
"Creating ActiveX Controls for the Internet Explorer on Pocket PC, via the Active Template Library"

http://msdn.microsoft.com/library/en-us/dnppc2k/html/ppc_pie.asp
Last download: April 2003

- [MSPIETEST] Microsoft:
"PIE Test Page"

<http://www.businessanyplace.net/sample/pie2002>
Last download: April 2003
- [MSPOWERNOTE] Microsoft Development Network (MSDN):
"Power Notification"

http://msdn.microsoft.com/library/en-us/dnppc2k/html/ppc_otification.asp
Last downloaded: June 2003
- [MSPOWERTOYS] Microsoft:
"PowerToys for the Pocket PC"

<http://www.microsoft.com/mobile/pocketpc/downloads/powertoys.asp>
Last download: April 2003
- [MSPPC2003] Microsoft (Presrelease):
"Microsoft Unveils Windows Mobile 2003 for Pocket PCs"
June 23, 2003

<http://www.microsoft.com/presspass/press/2003/Jun03/06-23Mobile2003PPCLaunchPR.asp>
Last download: August 2003
- [MSQ184650] Microsoft Knowledge Base Article # 184650:
"INFO: What Language Features does VB have that VBCE and eVB Do Not"
September 4, 2002

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;184650>
Last download: April 2003
- [MSQ185223] Microsoft Knowledge Base Article # 185223:
"HOW TO: Manually Uninstall Visual Basic CE Programs"
October 27, 2002

<http://support.microsoft.com/support/kb/articles/q185/2/23.asp>
Last download: June 2003
- [MSQ259369] Microsoft Knowledge Base Article # 184650:
"TCP Ports Required by ActiveSync"
June 4, 2003

<http://support.microsoft.com/default.aspx?scid=kb;en-us;259369>
Last downloaded: June 2003
- [MSQ260081] Microsoft Knowledge Base Article # 260081:
"Frequently Asked Questions (FAQ) About eVB"
September 4, 2002

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;260081>
Last download: April 2003
- [MSQ314989] Microsoft Knowledge Base Article # 314989:
"Let Me In: Pocket PC User Interface Password Redirect Sample"
May 20, 2003

<http://support.microsoft.com/default.aspx?scid=kb;en-us;314989>
Last download: June 2003

- [MSQ322956] Microsoft Knowledge Base Article # 322956:
"Sample to Add Root Certificates to Pocket PC 2002"
Juli 18, 2002
- <http://support.microsoft.com/default.aspx?scid=kb;EN-US;322956>
Last download: April 2003
- [MSRAPI] Microsoft Development Network (MSDN):
"RAPI Policy"
- http://msdn.microsoft.com/library/en-us/amo_ppc/htm/rapi_policy_emgb.asp
Last download: July 2003
- [MSSMARTCARD] Microsoft Development Network (MSDN):
"Microsoft Windows CE 3.0 Smart Card Subsystem"
June 2000
- <http://msdn.microsoft.com/library/en-us/dnce30/html/cesmartcard30.asp>
Last download: April 2003
- [MSSYNCLOCKED] Microsoft:
"Reply: ActiveSync can access a locked workstation w/o unlocking"
on BUGTRAQ April 16, 2001
- <http://cert.uni-stuttgart.de/archive/bugtraq/2001/04/msg00282.html>
Last download: May 2003
- [MSTECHNET2002] Microsoft:
"Pocket PC Security"
September 30, 2002
- <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itcommunity/chats/trans/Security/sec0930.asp>
Last download: March 2003
- [MSTELURL] Microsoft Development Network (MSDN):
"SIP and TEL URLs"
- http://msdn.microsoft.com/library/en-us/rtccInt/rtc/sip_and_tel_urls.asp
Last download: May 2003
- [MSVPNFAQ] Microsoft:
"Virtual Private Networking: Frequently Asked Questions"
March 25, 2003
- <http://www.microsoft.com/windowsserver2003/techinfo/overview/vpnfaq.msp>
Last download: May 2003
- [MSVNPPTP] Microsoft Development Network (MSDN):
"Microsoft Windows CE 3.0 Support of PPP and PPTP"
September 2000
- http://msdn.microsoft.com/library/en-us/dnce30/html/ppp_pptp.asp
Last downloaded: May 2003
- [MSWINPOWEREDFAQ] Microsoft:
"Frequently Asked Questions"
October 24, 2001
- <http://www.microsoft.com/windows/powered/mobile/evaluation/faq/default.asp>
Last download: April 2003

- [MSWINPOWEREDLOGO] Microsoft:
"Windows Powered"

<http://www.microsoft.com/windows/powered>
Last download: April 2003
- [NAIVILBO] McAfee Virus Information Library:
"Entry for Back Orifice"
October 16, 2002

http://vil.nai.com/vil/content/v_10002.htm
Last download: June 2003
- [NAIVILELKERN] McAfee Virus Information Library:
"Entry for W32/Elkern.cav"
October 17, 2002

http://vil.nai.com/vil/content/v_99238.htm
Last download: June 2003
- [NAIVILPWSTROY] McAfee Virus Information Library:
"Entry for VBS/PWStroy"
Juli 09, 2002

http://vil.nai.com/vil/content/v_99127.htm
Last download: June 2003
- [NAIVILSKA] McAfee Virus Information Library:
"Entry for W32/Ska@M"
September 27, 2002

http://vil.nai.com/vil/content/v_10144.htm
Last download: May 2003
- [NAIVILVBSWG] McAfee Virus Information Library:
"Entry for VBSWG"
September 18, 2002

http://vil.nai.com/vil/content/v_99011.htm
Last download: June 2003
- [NAIWEBINFO] McAfee:
"VirusScanWireless"

<http://www.mcafee.com/myapps/vsw/default.asp>
Last download: May 2003
- [NEWSKANE02] Margaret Kane:
"Handheld market contracts"
in CNET News.com, October 31, 2002

<http://news.com.com/2100-1040-963511.html>
Last download: January 2003
- [NORDENHOLZ2002] Tiemo Nordenholz:
*"Synchronizing Windows CE with a UN*X box"*
April 27, 2002

<http://www.thiemo.net/projects/cassi>
<http://www.thiemo.net/projects/cassi/connect1.shtml>
Last download: July 2003

- [NTTPR] NTT DoCoMo (Pressrelease):
"NTT DoCoMo to Advice Customers about Malicious E-mails"
June 13, 2001

http://www.nttdocomo.com/current_information/product/pressrelease/article/20010613-57215.html
Last download: August 2003
- [OCHOA1999] Hernan Ochoa:
"ActiveSync 3.0 Vulnerability: Obtaining the Partnership's Password"
1999

<http://www.cegadgets.com/arthernanpass.htm>
Last downloaded: June 2003
- [OMAP710WEB] Texas Instruments:
"2.5G/3G OMAP Processors : OMAP710 Processor"

<http://focus.ti.com/omap/docs/omapgenpage.tsp?navigationId=9303&templatedId=5663&path=templatedata/cm/omapproc/data/omap710>
Last download: February 2003
- [PGPMOBILE] PGP:
"PGP MOBILE PRODUCTS"

<http://www.pgp.com/products/mobile.html>
Last download: April 2003
- [PHMTASK] Philippe Majerus:
"PHM Taskmanager"

<http://www.phm.lu/Products/PocketPC/taskmgr.asp>
Last download: May 2003
- [POINTSEC] PointSec Mobile Device Security:
"Pointsec for Pocket PC 2.0 - Product Overview"
March 2003

http://www.pointsec.com/news/download/Pointsec_PPC_2.0_POP_PA1.pdf
Last download: April 2003
- [POLICYTWEAK] Philippe Majerus:
"Enable policy restrictions (Pocket PC 2002)"
posted on a registry tweak list on July 26, 2002
original author unknown

<http://www.phm.lu/PocketPC/tRegTweaks/Tweak.asp?ref=59>
Last download: August 2003
- [PPC2002SDK] Microsoft:
Pocket PC 2002 Software Development Kit
Version 1/31/2002 on CD ROM, also downloadable from
<http://www.microsoft.com/mobile/developer/downloads/ppcsdk2002.asp>
- [PPCLOGO] Microsoft:
"Designed for Windows for Pocket PC - Handbook for Software Applications"
April 9, 2002

http://www.eu.microsoft.com/mobile/assets/PocketPC_SoftApp_Handbook.doc
Last download: November 2002

- [PPCTEST2003-05] Henrich C. Pöhls
 "*Pocket PC Scanner Test 2003-05*"
 May 2003

 <http://agn-www.informatik.uni-hamburg.de/vtc/>
- [RESCOWEB] RESCO:
 "*Resco Explorer 2003*"

 <http://www.resco-net.com/explorer.asp>
 Last downloaded: April 2003
- [RFC1421-1424] IETF:
 RFC 1421: "*Privacy Enhancement for Internet Electronic Mail - Part I*"
 <http://www.ietf.org/rfc/rfc1421.txt>
 RFC 1422: "*Privacy Enhancement for Internet Electronic Mail - Part II*"
 <http://www.ietf.org/rfc/rfc1422.txt>
 RFC 1423: "*Privacy Enhancement for Internet Electronic Mail - Part III*"
 <http://www.ietf.org/rfc/rfc1423.txt>
 RFC 1424: "*Privacy Enhancement for Internet Electronic Mail - Part IV*"
 <http://www.ietf.org/rfc/rfc1424.txt>
 All last download: April 2003
- [RFC1661] IETF:
 RFC 1661: "*The Point-to-Point Protocol (PPP)*"
 July 1994

 <http://www.ietf.org/rfc/rfc1661.txt>
 Last download: November 2002
- [RFC1847] IETF:
 RFC 1847: "*Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted*"
 October 1995

 <http://www.ietf.org/rfc/rfc1847.txt>
 Last download: April 2003
- [RFC2015] IETF:
 RFC 2015: "*MIME Security with Pretty Good Privacy (PGP)*"
 October 1996

 <http://www.ietf.org/rfc/rfc2015.txt>
 Last download: April 2003
- [RFC2222] IETF:
 RFC 2222: "*Simple Authentication and Security Layer (SASL)*"
 October 1997

 <http://www.ietf.org/rfc/rfc2222.txt>
 Last download: March 2003
- [RFC2246] IETF:
 RFC 2246 DRAFT: "*The TLS Protocol Version 1.1*"
 March 2003

 <http://www.ietf.org/internet-drafts/draft-ietf-tls-rfc2246-bis-02.txt>
 Last download: November 2002
 Latest Draft:
 <http://www.ietf.org/internet-drafts/draft-ietf-tls-rfc2246-bis-03.txt>
 Last download: March 2003

- [RFC2487] IETF:
RFC 2487: *"SMTP Service Extension for Secure SMTP over TLS"*
January 1999

<http://www.ietf.org/rfc/rfc2487.txt>
Last download: March 2003
- [RFC2554] IETF:
RFC 2554: *"SMTP Service Extension for Authentication"*
March 1999

<http://www.ietf.org/rfc/rfc2554.txt>
Last download: March 2003
- [RFC2595] IETF:
RFC 2595: *"Using TLS with IMAP, POP3 and ACAP"*
January 1999

<http://www.ietf.org/rfc/rfc2595.txt>
Last download: March 2003
- [RFC2633] IETF:
RFC 2633: *"S/MIME Version 3 Message Specification"*
June 1999

<http://www.ietf.org/rfc/rfc2633.txt>
Last download: April 2003
- [RFC2637] IETF:
RFC 2637: *"Point-to-Point Tunneling Protocol (PPTP)"*
July 1999

<http://www.ietf.org/rfc/rfc2637.txt>
Last downloaded: May 2003
- [RFC3078] IETF:
RFC 3078: *"Microsoft Point-To-Point Encryption (MPPE) Protocol"*
March 2001

<http://www.ietf.org/rfc/rfc3078.txt>
Last downloaded: May 2003
- [ROCKMAN2002] Simon Rockman:
"Security flaw in Pocket PC Phone Edition"
May 18, 2002 in The Register

<http://www.theregister.co.uk/content/59/25336.html>
Last download: April 2003
- [ROGETS5] Peter Mark Roget, edited by Robert L. Chapman
"Roget's International Thesaurus 5th Edition"

HarperCollins Publishers, New York, 1992
ISBN: 0-00-470388-X
- [ROTNES2003] Christopher Sogge Røtnes:
"JS Bug makes it possible to deliberately crash Pocket PC IE"
posted on BugTrag January 3, 2003

<http://www.securityfocus.com/archive/1/305083>
Last download: April 2003

- [RSASMIME] RSA:
"Products from Vendors in the Get S/MIME Program"

<http://www.rsasecurity.com/standards/smime/products.html>
Last download: April 2003
- [SAMPLES2001] Jeff Samples:
"ActiveSync can access a locked workstation w/o unlocking"
on BUGTRAQ April 16, 2001

<http://cert.uni-stuttgart.de/archive/bugtraq/2001/04/msg00268.html>
Last downloaded: May 2003
- [SANDISK] SanDisk Corporation:
"SanDisk Introduces Four Gigabyte CompactFlash Card, World's Highest Capacity CF Flash Memory Card"
March 13, 2003

http://www.sandisk.com/pressrelease/031303_cf4GB.htm
Last download: August 2003
- [SCHEIDL1999] Gerald Scheidl:
"Virus Naming Convention 1999 (VNC99)"
Revision Beta, July 15, 1999

<http://members.chello.at/erikajo/vnc99b2.txt>
Last download: February 2003
- [SEEDORF2002] Jan F. Seedorf:
"Verfahren zur Qualitätsbestimmung der Erkennung von bössartiger Software"
August, 2002

http://agn-www.informatik.uni-hamburg.de/papers/doc/Diplomarbeit_JanSeedorf_pdf.zip
Last download: September 2002
- [SHARGIN2003] Alexander Shargin:
"QA: What applications are listed in the running program list?"
In Pocket PC Developer Network, May 09, 2003

<http://www.pocketpcdn.com/articles/applist.html>
Last download: June 2003
- [SIERAKOWSKI2002] Richard Sierakowski:
"Support Notes: SSL Support for PDA"
August 1, 2002

https://my.anlx.net/notes/ssl/ssl_pda.html
Last download: April 2003
- [SMARTPHONE2002SDK] Microsoft:
Smartphone 2002 Software Development Kit
Version from CD ROM, also downloadable from:
http://download.microsoft.com/download/Smartphone/SDK/1.0/WCE/EN-US/smartphone_2002_sdk.exe
- [SP2002SDKHELP] Microsoft:
Smartphone 2002 SDK Helpfile

included in [SMARTPHONE2002SDK]

- [SPVDEV] Orange Developer Website:
"SPV Specification"

<http://www.developers.orange.com/spvspec/>
Last download: April 2003
- [SPVPRESS] Orange (Pressrelease):
*"ORANGE AND MICROSOFT LAUNCH THE WORLD'S
FIRST WINDOWS POWERED SMARTPHONE – THE SPV"*
October 22, 2002

<http://orange.dk/spv/presse/uk>
Last downloaded: April 2003
- [SPVUNLOCK] msmobiles.com:
"How to software-unlock Orange SPV cell phone"

<http://msmobiles.com/article.php/20.html>
Last download: May 2003
- [SPVUNLOCKDEV] Orange Developer Website:
"Unlocking the SPV - Full Instructions"

[http://developers.orange.com/tipsandhints/2003/04/
24/Unlocking_full_instructions](http://developers.orange.com/tipsandhints/2003/04/24/Unlocking_full_instructions)
Last download: May 2003
- [SSL3] Alan O. Freier, Philip Karlton, Paul C. Kocher:
"The SSL Protocol - Version 3.0"
November 18, 1996

<http://wp.netscape.com/eng/ssl3/draft302.txt>
Last download: November 2002
- [STRONGARM] intel:
"Intel SA 1110 processor"

http://www.intel.com/design/pca/applicationsprocessors/1110_brf.htm
Last download: November 2002
- [TACKE2001] Christopher Tacke:
"How to use GwesPowerOffSystem from eVB"
September 16, 2001

<http://www.innovatedss.com/Resource/PowerOff.asp>
Last download: June 2003
- [TANENBAUM 1996] Andrew S. Tanenbaum:
"Computer Networks - Third Edition"

Prentice Hall, New Jersey, 1996
ISBN: 0-13-349945-6
- [TCPTUNNEL] Sureshot:
TCP-Tunnel Tool

<http://www.sureshotsoftware.com/tcptunnel/>
Last download: June 2003

- [TLISTKILL] SymbolicTools:
"Tlist/Kill"
February 2, 2003

<http://www.symbolictools.de/public/pocketconsole/applications/tlist/>
Last download: June 2003
- [TRENDFAQ] Trendmicro:
"FAQ for PC-cillin for Wireless - Version 2.0"
June 26, 2001

<http://www.trendmicro.com/ftp/documentation/guides/pccw20faq.pdf>
Last download: April 2003
- [TRENDMANUAL] Trendmicro:
"PC-cillin for Wireless Version 2.0 Users Manual"

<http://www.trendmicro.com/ftp/documentation/guides/pccw20manual.pdf>
Last download: April 2003
- [TRENDPATTERNWEB] Trendmicro:
Virus Pattern Version 345 for PC-Cillin Wireless (Pocket PC)

Direct Pattern#345 Download: http://www.trendmicro.com/ftp/products/wireless_pattern/slm345.zip
Last download: February 18, 2003 - 01:10

Overview:
<http://www.trendmicro.com/download/pattern.asp>
- [TRENDPCCWEB] Trendmicro:
PC-cillin for Wireless (Pocket PC)

http://www.trendmicro.com/ftp/products/pccillin/pc-cillin_for_pocket_pc_arm.exe
Last download: February 17, 2003
- [TRENDREADME] Trendmicro:
"PC-cillin for Wireless Version 2.0 - ReadMe"
June 26, 2001

http://de.trendmicro-europe.com/global/products/collaterals/read_me/pccw_pocket_arm_readme.zip
Last download: February 17, 2003
- [UY2001] Alex Uy:
"Re: ActiveSync can access a locked workstation w/o unlocking"
on BUGTRAQ April 16, 2001

<http://cert.uni-stuttgart.de/archive/bugtraq/2001/04/msg00282.html>
Last download: July 2003
- [VBVENDORLIST] Virus Bulletin:
"VB100-Award vendor list"

<http://www.virusbtn.com/vb100/archives/products.xml>
Last download: January 2003
- [VERISIGN] VeriSign:
"Software Publisher Digital IDs for Microsoft Authenticode Technology"

<http://www.verisign.com.au/developer/msauthenticode.shtml>
Last download: June 2003

- [VGREP] Virus Bulletin:
"Vgrep"

<http://www.virusbtn.com/resources/vgrep/>
Last downloaded: August 2003
- [VTCCODE] aVTC:
"VTC Code of Conduct"
June 2001

<ftp://agn-www.informatik.uni-hamburg.de/pub/CodeConduct/CoC-016.txt>
Last downloaded: June 2003
- [VTCINFO] aVTC:
"Viren und Malware - Eine Einführung"
Hamburger Computer Tage 2002, Information Brochure

<http://agn-www.informatik.uni-hamburg.de/hct/VTC.pdf>
Last downloaded: June 2003
- [VTCTEST2002-12] aVTC:
"aVTC Test 2002-12"
January 29, 2003

<http://agn-www.informatik.uni-hamburg.de/vtc/dt0212.htm>
Last download: April 2003
- [VTCTESTPROTO2002] aVTC:
"AV Product Test Protocol"
from aVTC Test 2002-12

<ftp://agn-www.informatik.uni-hamburg.de/pub/texts/tests/pc-av/2002-12/5protoco.txt>
Last download: April 2003
- [VTCWEBSITE] aVTC:
Website

<http://agn-www.informatik.uni-hamburg.de>
<http://www.avtc.info>
Last download: March 2003
- [WEBTESTBUG] Henrich C. Pöhls:
"Testpage for PIE Jscript BUG"
written with code from [ROTNES2003]

http://www.2000grad.com/security/website/crash_pie.html
Last downloaded: May 2003
- [WEBTESTJS] Henrich C. Pöhls:
"Testpage for JavaScript"
written with code from [CLINICK2000]

http://www.2000grad.com/security/website/script_version.html
Last downloaded: May 2003
- [WEBTESTPOLICY] Henrich C. Pöhls:
"Testpage for Policy Restrictions on Downloads"

<http://www.2000grad.com/security/website/policytest/index.html>
Last downloaded: August 2003

- [WEBTESTTEL] Henrich C. Pöhls:
 "*Testpage for Telephone-URL*"

 <http://www.2000grad.com/security/website/smakecall.html>
 Last downloaded: May 2003
- [WILDLIST] WildList Organization:
 Website

 <http://www.wildlist.org>
 Last download: February 2003
- [WILDLISTDEC02] WildList Organization:
 "*WildList December 2002*"
 January 2002

 <http://www.wildlist.org/WildList/200212.txt>
 Last download: February 2003
- [WILDLISTNOV02] WildList Organization:
 "*WildList November 2002*"

 <http://www.wildlist.org/WildList/200211.txt>
 Last download: December 2002
- [XSCALE] intel:
 "*Intel XSCALE PXA250 Applications Processor*"

 <http://www.intel.com/design/pca/prodbref/298620.htm>
 Last download: January 2003
- [YAO2000] Paul Yao:
 "*Windows CE 3.0: Enhanced Real-Time Features
 Provide Sophisticated Thread Handling*"
 MSDN Magazine, November 2000

 <http://msdn.microsoft.com/msdnmag/issues/1100/RealCE/default.aspx>
 Last download: April 2003
- [ZEHLER2002] Jeff Zehler:
 "*Prevent Data Loss When You Lose Power*"
 2002

 [http://www.intel.com/pca/developernetwork/solutionsjournal/
spring_02/pdf/intel_psm.pdf](http://www.intel.com/pca/developernetwork/solutionsjournal/spring_02/pdf/intel_psm.pdf)
 Last download: April 2003

List of Figures

Figure 1: Number of malware used in the testbeds of recent aVTC tests	1
Figure 2: Computing power versus size of mobile device	4
Figure 3: EMEA mobile device shipments for Q1-2002 and Q1-2001 [CANALYS2002]	5
Figure 4: Worldwide Unit Sales Estimates [ETFORECAST2002]	6
Figure 5: Steps for Prevention of Malware Distribution or Contamination	11
Figure 6: A testbed divided into two test sets, with m and n-m samples in each test set	13
Figure 7: View of the running processes on a Pocket PC 2002	21
Figure 8: Interrupt handling in Windows CE 3.0	23
Figure 9: Screen shot from the memory settings dialog	26
Figure 10: Windows CE memory allocation	27
Figure 11: Remote Process View showing: Memory Access Key and DLL Memory Base Address	28
Figure 12: Closing running programs	29
Figure 13: Trying to overload a ROM file in: File Explorer (l.), Resco Explorer (mid.), ActiveSync (r.)	32
Figure 14: Tap-and-Hold gesture brings up a pop-up menu	35
Figure 15: Overview of different Windows CE networking components and their interactions	37
Figure 16: No valid device ID found (left), Network Logon dialog (right)	40
Figure 17: NDIS: NE2000 miniport driver	41
Figure 18: ActiveSync: Partnership-Dialog, Password-Dialog on the desktop computer	44
Figure 19: ActiveSync Application: main screen (left), options dialog (right)	45
Figure 20: Illustrating the splitting of the ActiveSync application over two PCs	46
Figure 21: ActiveSync: With Pass Through the home system acts as a proxy server	46
Figure 22: ActiveSync: File Filter Configuration Dialogs	48
Figure 23: Connection Manager: Selection dialog (l.) and Work configuration dialogs (m. + r.)	48
Figure 24: Connection Manager: Logon credentials dialog (left), Connecting (right)	49
Figure 25: Power settings dialog	50
Figure 26: Standard PIN screen (l.), password settings (m.) and revealed emergency number (r.)	51
Figure 27: iPAQ5450: Fingerprint-Login (l.), PIN/Fingerprint combinations (m.), max logon attempts (r.)	52
Figure 28: Alarm notification displayed on top of the power-on-protection dialog on wake up	53
Figure 29: Overview of the SSPI [EVT3HELP]	53
Figure 30: CAPI overview [EVT3HELP]	54
Figure 31: Smartcard support through SCSPs and Smartcard Hardware Drivers	55
Figure 32: Application execution under the Smartphone 2002 trust-model	56
Figure 33: Undocumented Policy Restrictions: Icon (l.), dialog (mid.), administrator password (r.)	58
Figure 34: Error message when copying an exe or cab file to the mobile device with policy restriction	58
Figure 35: Downloading exe files through PIE is blocked if the policy restrictions is enabled	59
Figure 36: Save Dialog of Pocket Outlook allows to assign an extension	59
Figure 37: File Explorer does not reveal the last file extension	61
Figure 38: "Beam File..." dialog might reveal the full filename and the extension	62
Figure 39: File Explorer: Activate "Show All Files" to list also hidden files	62
Figure 40: Pocket Word: Saving in different formats (l.), original DOC-File (mid.) in Pocket Word (r.)	63
Figure 41: Pocket Internet Explorer shows its JScript version [WEBTESTJS]	65
Figure 42: Asking to call, after clicking on a TEL-URL	65
Figure 43: Wrong SSL certificate message in PIE (left) and the correct message in IE 6.0 (right)	66
Figure 44: PIE page properties reveals SSL secured pages	67
Figure 45: Standard settings in PIE (left and middle) and additional settings in MS PowerToys (right)	67
Figure 46: Advanced E-mail settings allow to enable authenticated SMTP	69
Figure 47: Pocket Outlook: Contents in Inbox (l.), E-mail with differently named attachment (m. + r.)	71
Figure 48: Embedded Links are made click-able in Pocket Outlook	71
Figure 49: VBScript version shown by little eVB application (see Appendix E)	72
Figure 50: Malware distribution through an e-mail attachment	74
Figure 51: Downloading, saving and by default also executing files with Pocket Internet Explorer	81
Figure 52: A600 Settings allow to disable Auto-run from removable storage media [A600MANUAL]..	83
Figure 53: Example of a Hostile Applet, that misuses a safe-for-scripting ActiveX object to send e-mail	86
Figure 54: Six different steps of the test procedure	108
Figure 55: Kaspersky AV scanner options	112
Figure 56: Kaspersky AV scan report	113

Figure 57: Kaspersky AV report example from the manual [AVPMANUAL] 113
Figure 58: Kaspersky Anti-Virus Updater application on the desktop PC 114
Figure 59: Today screen with F-Secure Anti-Virus 115
Figure 60: F-Secure scanner options 116
Figure 61: F-Secure scan report 117
Figure 62: Update URL and Database Update 117
Figure 63: eTrust Antivirus scanner options 119
Figure 64: eTrust scan report 120
Figure 65: Pattern information and update process for eTrust 120
Figure 66: PC-cillin scan report 122
Figure 67: Trendmicro's pattern download web site 123
Figure 68: Pre-Test: ITW Macro-Virus Detection Rate (sample and malware based) 125
Figure 69: ITW-Macro Virus Detection Rate (malware) on all TestBeds 127
Figure 70: ITW-Macro Virus Detection Rate (samples) on all TestBeds 127
Figure 71: Pre-test: ITW Script-Virus Detection Rate (sample and malware based) 129
Figure 72: ITW-Script Virus Detection Rate (malware) on all TestBeds 131
Figure 73: ITW-Script Virus Detection Rate (samples) on all TestBeds 131
Figure 74: Head-to-Head Comparison of detection rates from INO & PCC on latest testbeds 133
Figure 75 Comparison of the detection rate of Desktop and Pocket PC products 134
Figure 76: INO gives an error, while deleting the "infected" autorun.exe 136

List of Tables

Table 1: Grouping of mobile devices by processor speed and RAM size.....	18
Table 2: Difference between the three Pocket PC 2002 editions [EVT3HELP].....	20
Table 3: Windows CE thread priority levels.....	22
Table 4: Algorithms and key length of the default MS Base CSP	54
Table 5: Entry Points for Viruses	89
Table 6: All Virus Contamination Requirements are met	89
Table 7: All Virus Contamination Options are possible	90
Table 8: Entry Points for worms	90
Table 9: Not all Worm Contamination Requirements are fully met	91
Table 10: All Worm Contamination Options are possible.....	92
Table 11: Entry Points for Trojan Horses	93
Table 12: Not all Trojan horse Contamination Requirements are fully met	94
Table 13: All Trojan horse Options are possible	95
Table 14: Entry Points for Hostile Applets.....	95
Table 15: None of the Hostile Applet requirements are fully met.....	96
Table 16: All Hostile Applet Options are only possible with additional user interactions	96
Table 17: Comparison of the Testbed structure and the Wildlist	106
Table 18: List of anti-malware products vendors and their support of PDA operating systems.	111
Table 19: Kaspersky version information	111
Table 20: Kaspersky system requirements	112
Table 21: Kaspersky AV scanner options	112
Table 22: F-Secure version information	114
Table 23: F-Secure system requirements	115
Table 24: F-Secure scan options	116
Table 25: eTrust version information.....	118
Table 26: eTrust system requirements.....	118
Table 27: eTrust scan options	119
Table 28: PC-cillin version information.....	121
Table 29: PC-cillin system requirements.....	121
Table 30: Scanner Results for Macro ITW Viruses (Testbed: mac_itw.304, Nov. 2001-Dec. 2002)..	126
Table 31: Scanner Results for Macro ITW Viruses (Testbed: mac_itw.dec02, December 2002).....	126
Table 32: Scanner Results for Script ITW Viruses (Testbed: scr_itw.304, Nov. 2001-Dec. 2002).....	130
Table 33: Scanner Results for Script ITW Viruses (Testbed: scr_itw.dec02, December 2002)	130
Table 34: Auto-Insertion Test Set #1.....	135
Table 35: Auto-Insertion Test Set #2.....	135

Acknowledgement

I would like to express my thanks to all those who helped to make this work possible with their support and valuable hints.

Firstly, I would like to express my gratitude to my supervisors Prof. Dr. Klaus Brunnstein and Prof. Dr. Norbert Ritter for their interest in my work, their ongoing and friendly help and for the stimulating criticism.

Special thanks go to Frank Pollmann and Andreas Koch from T-Systems for making a T-Mobile MDA Pocket PC 2002 device available for the duration of this work. It was a good testing ground for the Pocket PC operating system's features and was the test device on which the anti-malware product test was performed.

Then my thanks go to the aVTC team, which always provided me with answers or hints on questions regarding the desktop anti-malware product tests. Especially Prof. Dr. Klaus Brunnstein for allowing access to the malware testbeds, Jan Seedorf for his continuous help and availability when I had questions, needed CDs, SD-Cards, Books or other assistance. Thanks also go to Michel Messerschmidt for explaining details on the aVTC reporting, and Stefan Heimann, for fast re-organising the latest testbeds.

Also to all the other people at the Arbeitsbereich AGN, to which I talked to in hallways and during seminars: You provided me with good questions, valuable sources and helpful hints, thank you.

I would also like to thank the people who proof read the numerous drafts that I produced (I think I had over 80 versions). Special thanks here go to Sven Poggensee and my brother Lennart, who also was a great help compiling the final printouts. Moreover, I could not help but to thank Word for not corrupting the document during the final phases, were it gained a size of over 90 Mbytes.

Finally, yet importantly, I would like to thank my whole family for all their support of any kind. They made my successful studies possible and a funny and interesting time of my life. In addition, I have to thank my girlfriend Yvonne Beck for her support and understanding for my long working nights.

This page is intentionally left blank

Appendix A – Abbreviations

ALU	Arithmetic Logic Unit
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
aVTC	anti-Virus Test Center
BT	Bluetooth
CAPI	Common ISDN Application Interface
CEGUID	Windows CE Globally Unique Identifier
CF	Compact Flash
CHAP	Challenge-Handshake Authentication Protocol
CIFS	Common Internet File System
COM	Component Object Model
CPU	Central Processing Unit
DAC	Discretionary Access Control
DCOM	Distributed Component Object Model
DLL	Dynamic Link Library
DoS	Denial of Service
DSP	Digital Signal Processing
DTMF	Dial Tone Multi Frequency
EAP	Enhanced Authentication Protocol
EMEA	European, Middle East and African region
FAQ	Frequently Asked Questions
FAT	File Allocation Table
FTP	File Transfer Protocol
GSM	Groupe Spéciale Mobile
GUI	Graphical User Interface
HKCU	HKEY_CURRENT_USER
HKLM	HKEY_LOCAL_MACHINE
HPC	Handheld PC
HTTP	Hyper-Text Transfer Protocol
ICMP	Internet Control Message Protocol
ID	Identifier
IP	Internet Protocol
IR	Infrared
IRDA	Infrared Data Association
IRQ	Interrupt Request
ISR	Interrupt Service Routine
IST	Interrupt Service Thread
IT	Information Technolog
LAN	Local Area Network
LCD	Liquid Crystal Display

MD5	Message Digest Algorithm Version 5
MIME	Multi-purpose Internet Mail Extension
MMC	Multi-Media Card
MPPC	Microsoft Point-to-Point compression
MPPD	Microsoft Point-to-Point data encryption
MS	Microsoft
MS CHAP	Microsoft Challenge-Handshake Authentication Protocol
MSMQ	Microsoft Message Queuing Service
NDIS	Network Driver Interface Specification
NIC	Network Interface Card
NTFS	New Technology File System
OEM	Original Equipment Manufacturer
OID	Object Identifier
OLE	Object Linking and Embedding
OS	Operating System
OSI	Open Systems Interconnection
PAP	Password Authentication Protocol
PC	Personal Computer
PC CARD	newer term for "PCMCIA"
PCMCIA	Personal Computer Memory Card International Association
PCT	Private Communication Technology
PDA	Personal Digital Assistant
PEM	Privacy Enhanced Mail
PGP	Pretty Good Privacy
PIE	Pocket Internet Explorer
PIM	Personal Information Manager
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
QoS	Quality of Service
RAM	Random Access Memory
RAS	Remote Access Service
RISC	Reduced Instruction Set Computer
ROM	Read-Only Memory
RPC	Remote Procedure Call
SASL	Simple Authentication and Security Layer
SD	Secure Digital
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UNC	Universal Naming Convention
URL	Uniform Resource Locator

VM	Virtual Machine
VPN	Virtual Private Network
WAP	Wireless Application Protocol
WAV	Waveform Audio
WLAN	Wireless Local Area Network
XiP	Execute in Place

Appendix B – Details of Pocket PC 2002 compatible devices



ASUS MYPAL A600

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel PXA250 XScale @ 400 MHz
Size: 125 x 75 x 13
Weight: 138 g
Price: 600 EUR
Battery: 1200 mAh Lithium Ion
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB
Interfaces: IRDA, SD, MMC, CF via optional expansion pack, Microphone, Headphone
Software:
Accessories:
Manufacturer's Link: usa.asus.com/PDA/a600/specification.htm



Casio Cassiopeia E-200

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel StrongARM SA-1110 32-bit processor @ 206 MHz
Size: 130 x 82 x 18
Weight: 190 g
Price: 600 EUR
Battery: Lithium Ion
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB
Interfaces: IRDA, SD, MMC, CF-II, Microphone, Headphone, Typ II via optional Jacket
Software: Office, e-Mail, Video, Internet, Sound, MP3, Images, SMS, Chat
Accessories: USB-Cradle, Power-Supply, Schutzhülle
Manufacturer's Link: www.casio.de/cassiopeia/e200g/



Compaq iPAQ H1910

OS: MS Pocket PC 2002
Display: Transflective TFT LCD Touchscreen [240 x 320] with 65536 Colors
CPU: Intel PXA250 @ 200 MHz
Size: 114 x 70 x 13
Weight: 120 g
Price: 300 EUR
Battery: 900 mAh Lithium Ion
FLASH-ROM: 16 MB
RAM: 64 MB
Sync-Port: USB
Interfaces: IRDA, BlueTooth, SD, MMC, Microphone, Headphone
Software: Office, Voice Recorder, Internet, MP3, Video, eMail, Microsoft Reader (eBooks), MSN Messenger, VPN Client
Accessories: slim removable battery, USB synchronization cable, Power-Supply, hp iPAQ PC Companion CD
Manufacturer's Link: www.compaq.com/products/quickspecs/11491_na/11491_na.HTML



Compaq iPAQ H3765

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 4096 Colors
CPU: Intel StrongARM SA-1110 32-bit RISC processor @ 206 MHz
Size: 130 x 84 x 16
Weight: 178 g
Price: 500 EUR
Battery: 950 mAh Lithium Polymer
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB, optional RS-232(Serial-Port)
Interfaces: IRDA[115 Kbps], SD, Microphone, Headphone, PCMCIA via expansion pack
Software: Office, e-Mail, Internet, Video, Imageser, MP3, Terminal Services Client, VPN Client, Infrared Beaming
Accessories: Power-Supply, USB-Cradle
Manufacturer's Link: www.compaq.com/products/quickspecs/10973_na/10973_na.HTML



Compaq iPAQ H3830

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel StrongARM SA-1110 32-bit RISC Processor @ 206 MHz
Size: 135 x 84 x 15
Weight: 184 g
Price: 700 EUR
Battery: 1400 mAh Lithium Polymer
FLASH-ROM: 32 MB
RAM: 32 MB
Sync-Port: USB, RS-232(Serial-Port)
Interfaces: IrDA[115 Kbps], SD, MMC, CF II via external Jacket, Microphone, Headphone, PCMCIA via external Jacket
Software: Browser, WAP, E-Mail, SMS, Office, Images, Video, MP3
Accessories: Docking-Station, Power-Supply, Schutzhuelle
Manufacturer's Link: www.compaq.com/products/handhelds/pocketpc/H3830.html



Compaq iPAQ H3850

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel StrongARM SA-1110 32-bit RISC Processor @ 206 MHz
Size: 135 x 84 x 16
Weight: 184 g
Price: 700 EUR
Battery: 1400 mAh Lithium Polymer
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB, RS-232(Serial-Port)
Interfaces: IrDA[115 Kbps], SD, MMC, CF II via external Jacket, Microphone, Headphone, PCMCIA via external Jacket
Software: Browser, WAP, E-Mail, SMS, Office, Images, Video, MP3
Accessories: Docking-Station, Power-Supply, Schutzhuelle
Manufacturer's Link: www.compaq.com/products/handhelds/pocketpc/H3850.html



Compaq iPAQ H3870

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel StrongARM SA-1110 32-bit RISC Processor @ 206 MHz
Size: 135 x 84 x 16
Weight: 184 g
Price: 850 EUR
Battery: 1400 mAh Lithium Polymer
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB, RS-232(Serial-Port)
Interfaces: IrDA[115 Kbps], BlueTooth, SD, MMC, CF II via external Jacket, Microphone, Headphone, PCMCIA via external Jacket
Software: Browser, WAP, E-Mail, SMS, Office, Images, Video, MP3
Accessories: Docking-Station, Power-Supply, Schutzhuelle
Manufacturer's Link: www.compaq.com/products/handhelds/pocketpc/H3870.html



Compaq iPAQ H3950

OS: MS Pocket PC 2002
Display: Transflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel PXA250 Application Processor @ 400 MHz
Size: 134 x 84 x 16
Weight: 184 g
Price: 700 EUR
Battery: 1400 mAh Lithium Polymer
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB, RS-232(Serial-Port)
Interfaces: IrDA[115 Kbps], SD, CF via optional extension, Microphone, Headphone, PCMCIA via optional external Jacket
Software: Office, e-Mail, Internet, Video, Images, MP3, VPN Client, Infrared Beaming
Accessories: USB-Cradle, Power-Supply
Manufacturer's Link: www.compaq.com/products/handhelds/pocketpc/H3950.html



Compaq iPAQ H3970

OS: MS Pocket PC 2002
Display: Transflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel PXA250 Application Processor @ 400 MHz
Size: 134 x 84 x 16
Weight: 184 g
Price: 750 EUR
Battery: 1400 mAh Lithium Polymer
FLASH-ROM: 48 MB
RAM: 64 MB
Sync-Port: USB, RS-232(Serial-Port)
Interfaces: IrDA[115 Kbps], BlueTooth, SD, CF via optional extension, Microphone, Headphone, PCMCIA via optional external Jacket
Software: Office, e-Mail, Internet, Video, Images, MP3, VPN Client, Infrared Beaming
Accessories: USB-Cradle, Power-Supply
Manufacturer's Link: www.compaq.com/products/handhelds/pocketpc/H3970.html



Compaq iPAQ H5450

OS: MS Pocket PC 2002
Display: Transflective TFT LCD Touchscreen [240 x 320] with 65536 Colors
CPU: Intel XSCALE @ 400 MHz
Size: 138 x 84 x 16
Weight: 206 g
Price: 700 EUR
Battery: 1250 mAh Lithium Ion
FLASH-ROM: 48 MB
RAM: 64 MB
Sync-Port: USB, RS-232 (Serial)
Interfaces: IrDA, BlueTooth, SD, WLAN (802.11b), Microphone, Headphone
Software: Office, Voice Recorder, Internet, MP3, Video, eMail, Microsoft Reader (eBooks), MSN Messenger, VPN Client, iPAQ Fingerprint-Extension, iPAQ Dateispeicher, iPAQ Task Manager, iPAQ Backup, iPAQ Image
Accessories: Internal Fingerprint-Scanner, slim removable battery, protective cover pack, universal cradle (USB or RS-232), Replacement-Pen, Power-Supply, hp iPAQ PC Companion CD
Manufacturer's Link: www.compaq.com/products/quickspecs/11427_na/11427_na.HTML



Dell AXIM X5 Advanced

OS: MS Pocket PC 2002
Display: TFT Touch Sensitive Transflective LCD [240 x 320] with 65536 Colors
CPU: Intel XScale @ 400 MHz
Size: 128 x 82 x 18
Weight: 196 g
Price: 290 EUR
Battery: 1440 mAh Lithium-Ion
FLASH-ROM: 48 MB
RAM: 64 MB
Sync-Port: USB
Interfaces: IrDA, SD, MMC, CF-II, Microphone, Headphone
Software: Office, Email, Internet, Video, Picture
Accessories:
Manufacturer's Link: ww.dell.com/us/en/hied/products/model_pda_3_axim5_advanced.htm



Dell Axim X5 Standard

OS: MS Pocket PC 2002
Display: TFT Touch Sensitive Transflective LCD [240 x 320] with 65536 Colors
CPU: Intel XScale @ 300 MHz
Size: 128 x 82 x 18
Weight: 196 g
Price: 400 EUR
Battery: 1440 mAh Lithium-Ion
FLASH-ROM: 32 MB
RAM: 32 MB
Sync-Port: USB
Interfaces: IrDA, SD, MMC, CF-II, Microphone, Headphone
Software: Office, Internet, Email, Video, Images
Accessories:
Manufacturer's Link: www.dell.com/us/en/hied/products/model_pda_2_axim5_entry.htm



Fujitsu Siemens Pocket LOOX 600

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel PXA250 Applications Processor @ 400 MHz
Size: 132 x 82 x 17
Weight: 175 g
Price: 600 EUR
Battery: 1520 mAh Lithium Polymer
ROM: 32 MB
RAM: 64 MB
Sync-Port: USB, RS-232(Serial-Port)
Interfaces: IrDA[115 Kbps], BlueTooth, SD, MMC, CF-II, Microphone, Headphone
Software: Office, e-Mail, Internet, Video, Bluetooth-Software
Accessories: USB Docking Station, Protection Bag, Power-Supply
Manufacturer's Link: www.fujitsu-siemens.com/rl/products/handhelds/pocketloox.html



HP Jornada 565

OS: MS Pocket PC 2002
Display: Reflective TFT-Imagescreen [240 x 320] with 65536 Colors
CPU: Intel StrongARM SA-1110 32-bit RISC Processor @ 206 MHz
Size: 132 x 78 x 18
Weight: 176 g
Price: 650 EUR
Battery:
FLASH-ROM: 32 MB
RAM: 32 MB
Sync-Port: USB
Interfaces: CF I, Microphone
Software: SMS, Office, Images, Video, MP3
Accessories: Docking-Station, Power-Supply
Manufacturer's Link: www.hp-expo.com/de/ger/products/jornada/565.html



HP Jornada 568

OS: MS Pocket PC 2002
Display: Reflective TFT-Imagescreen [240 x 320] with 65536 Colors
CPU: Intel StrongARM SA-1110 32-bit RISC Processor @ 206 MHz
Size: 132 x 78 x 18
Weight: 176 g
Price: 750 EUR
Battery:
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB
Interfaces: CF I, Microphone
Software: SMS, Office, Images, Video, MP3
Accessories: Docking-Station, Power-Supply
Manufacturer's Link: www.hp-expo.com/de/ger/products/jornada/568.html



HP Jornada 928 WDA

OS: MS Pocket PC Phone Edition 2002
Display: Reflective TFT LCD Touchscreen [240 x 320] with 65536 Colors
CPU: Texas Instruments OMAP 710 @ 133 MHz
Size: 137 x 78 x 17
Weight: 184 g
Price: 999 EUR
Battery: Lithium Polymer
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB
Interfaces: IrDA, CF-I, 900+1800MHz EGSM/GPRS, Microphone,Headphone
Software: eMail, Web, Office, Phone, Internet, SMS, Video,
Accessories: Power-Supply, USB Cradle, removable battery, Stereo Headset with Microphone
Manufacturer's Link: h40054.www4.hp.com/pressrel/pc/jornada928.htm



NEC MobilePro P300 (MC/PG5000A)

OS: MS Pocket PC 2002
Display: Reflective QVGA-TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel StrongARM 32-bit processor @ 206 MHz
Size: 128 x 78 x 18
Weight: 190 g
Price: 600 EUR
Battery: Lithium Ion
FLASH-ROM: 32 MB
RAM: 32 MB
Sync-Port: USB, RS-232(Serial-Port)
Interfaces: IrDA, SD, CF-II, Microphone, Headphone
Software:
Accessories: Power-Supply, 32MB SD Card, USB Cradle, Serial connection
Manufacturer's Link: www.neccomp.com/products/MobilePro/P300/specifications.htm



O2 XDA

OS: MS Pocket PC Phone Edition 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 4096 Colors
CPU: Intel StrongARM 32-bit Processor @ 206 MHz
Size: 130 x 74 x 19
Weight: 201 g
Price: 500 EUR
Battery: Lithium Polymer
ROM: 32 MB
RAM: 32 MB
Sync-Port: USB
Interfaces: IrDA, SD, MMC, GPRS Class B, DGSM 900 / GSM 1800 MHz, Microphone, Headphone
Software: Office, Internet, Images, Video, Email, Voice recorder/ Dictaphone, Speakerphone
Accessories: Power-Supply, USB-Cradle
Manufacturer's Link: www.mmo2.com/docs/services/xda_how.html



T-Mobile MDA

OS: MS Pocket PC Phone Edition 2002
Display: Reflective TFT Touchscreen [240 x 320] with 4096 Colors
CPU: Intel StrongARM @ 206 MHz
Size: 130 x 73 x 19
Weight: 202 g
Price: 550 EUR
Battery: Lithium Polymer
ROM: 32 MB
RAM: 32 MB
Sync-Port: USB
Interfaces: IrDA, SD, MMC, GSM 900 + 1800 GPRS, Microphone, Headphone
Software: Office, Internet, Video, Image, SIM Manager, T-Online Messenger (TOM)
Accessories: Power-Supply, MDA-Case, 2 Touch-Pens, Stereo-Headset with Microphone
Manufacturer's Link: www.t-mobile.de/mda/



Toshiba e310

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel StrongARM 32-bit Processor @ 206 MHz
Size: 125 x 80 x 13
Weight: 140 g
Price: 400 EUR
Battery: 1000 mAh Advanced Lithium Ion
FLASH-ROM: 32 MB
RAM: 32 MB
Sync-Port: USB
Interfaces: IrDA[115 Kbps], SD, MMC, Microphone, Headphone
Software: Office, E-Mail, Internet, Video, Back up, MP3, Images
Accessories: Power-Supply, USB Cradle, Soft case
Manufacturer's Link: computers.toshiba-europe.com/cgi-bin/ToshibaCSG/selected_product_option.jsp?PRODUCT_ID=16250



Toshiba e570

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel StrongARM 32-bit Processor @ 206 MHz
Size: 78 x 125 x 18
Weight: 180 g
Price: 650 EUR
Battery: Advanced Lithium Ion
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB, RS-232 (Serial-Port)
Interfaces: IrDA[115 Kbps], SD, CF-II, Microphone, Headphone
Software: Office, E-Mail, Internet, Video, Back up, MP3, MPEG4, Images
Accessories: Power-Supply, USB Cradle, Soft case
Manufacturer's Link: computers.toshiba-europe.com/cgi-bin/ToshibaCSG/product_page.jsp?PRODUCT_ID=17783



Toshiba e740

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel® PXA250 @ 400 MHz
Size: 125 x 80 x 16
Weight: 175 g
Price: 650 EUR
Battery: 1000 mAh Lithium Ion
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB
Interfaces: IrDA[115 Kbps], SD, MMC, CF-II, Microphone, Headphone
Software: Office, E-Mail, Internet, Video, Back up, MP3, MPEG4, Image
Accessories: Power-Supply, USB Cradle, Soft case
Manufacturer's Link: computers.toshiba-europe.com/cgi-bin/ToshibaCSG/selected_product_option.jsp?PRODUCT_ID=18198



Toshiba e740 BT

OS: MS Pocket PC 2002
Display: Reflective TFT-Touchscreen [240 x 320] with 65536 Colors
CPU: Intel® PXA250 @ 400 MHz
Size: 125 x 80 x 16
Weight: 175 g
Price: 700 EUR
Battery: 1000 mAh Lithium Ion
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB
Interfaces: IrDA[115 Kbps], BlueTooth, SD, MMC, CF-II, Microphone, Headphone
Software: Office, E-Mail, Internet, Video, Back up, MP3, MPEG4, Image
Accessories: Power-Supply, USB Cradle, Soft case
Manufacturer's Link: computers.toshiba-europe.com/cgi-bin/ToshibaCSG/selected_product_option.jsp?PRODUCT_ID=18198

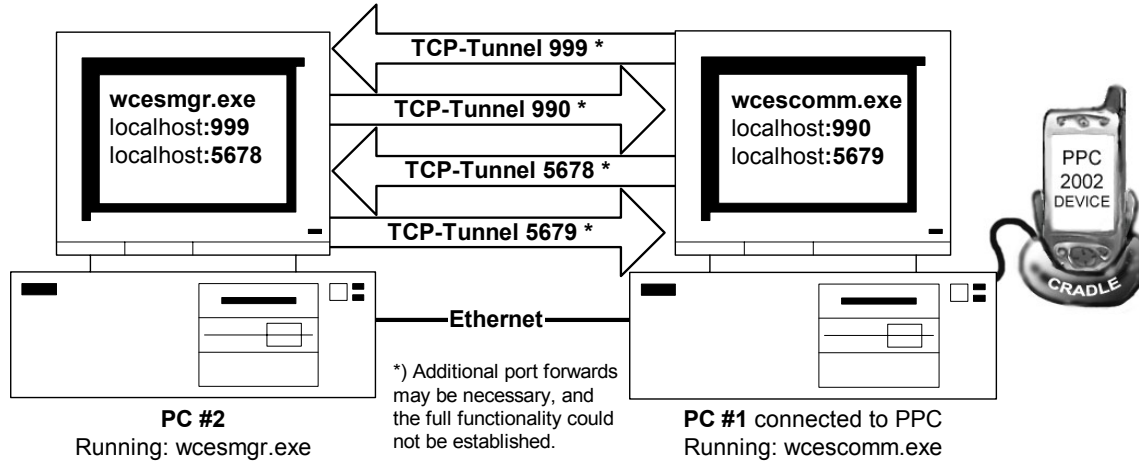


ViewSonic Pocket PC V35

OS: MS Pocket PC 2002
Display: Transflective TFT Touchscreen [320 x 240] with 65536 Colors
CPU: Intel XScale @ 300 MHz
Size: 123 x 77 x 13
Weight: 100 g
Price: 400 EUR
Battery: 900 mAh Lio-Ion
FLASH-ROM: 32 MB
RAM: 64 MB
Sync-Port: USB
Interfaces: IrDA, SD, MMC, Microphone, Headphone
Software:
Accessories: Power-Supply, USB Cradle
Manufacturer's Link: www.viewsonic.com/products/pocket_pc_pocketpcv35.htm

Appendix C – Details of ActiveSync Packets

To examine more details on the undocumented ActiveSync some packets exchanged between the two ActiveSync application parts `wcescomm.exe` and `wcesmgr.exe` were sniffed. The following figure illustrates the setup used to sniff on the connection. It shall be noted that additional TCP tunnels were necessary and that a full ActiveSync connection could not be established using this setup.



C.1. Sniffed Password Packet Details

The details of the packets sniffed after wrong passwords are entered three times at the desktop side to establish an ActiveSync connection with a mobile device with power-on password:

Packet #1, Direction: In, Time:15:31:08,448

TCP

Source port: 5679

Destination port: 1283

Raw Data:

```

0x0000  00 01 02 50 15 F5 00 00-1C D1 68 92 08 00 45 00  ...P.ö...Ñh'..E.
0x0010  00 40 4C BF 40 00 80 06-2C 12 C0 A8 00 64 C0 A8  .@Lç@.€. ,.À".dÀ"
0x0020  00 32 16 2F 05 03 AD 2A-10 92 51 77 D4 37 50 18  .2./...-*.'QwÔ7P.
0x0030  44 0C EA 77 00 00 16 00-83 B2 80 B2 81 B2 86 B2  D.êw....f²€²□²†²
0x0040  87 B2 84 B2 85 B2 8A B2-8B B2 82 B2 B2 B2      ‡²,,²...²Š²<²,²²²
    
```

Password: "1234567890"

Packet #2, Direction: Out, Time:15:31:08,618

TCP

Source port: 1283

Destination port: 5679

Raw Data:

```

0x0000  00 00 1C D1 68 92 00 01-02 50 15 F5 08 00 45 00  ...Ñh'...P.ö..E.
0x0010  00 28 15 8F 40 00 80 06-63 5A C0 A8 00 32 C0 A8  .(.□@.€.cZÀ".2À"
0x0020  00 64 05 03 16 2F 51 77-D4 37 AD 2A 10 AA 50 10  .d.../QwÔ7-*..ªP.
0x0030  62 58 CC DF 00 00                                     bXİß..
    
```

Packet #3, Direction: Out, Time:15:31:08,668

TCP

Source port: 1283

Destination port: 5679

Raw Data:

```

0x0000  00 00 1C D1 68 92 00 01-02 50 15 F5 08 00 45 00  ...Ñh'...P.ö..E.
0x0010  00 2A 15 92 40 00 80 06-63 55 C0 A8 00 32 C0 A8  .*.'@.€.cUÀ".2À"
0x0020  00 64 05 03 16 2F 51 77-D4 37 AD 2A 10 AA 50 18  .d.../QwÔ7-*..ªP.
0x0030  62 58 CC D5 00 00 00 00-                                bXİÖ....
    
```

Packet #4, Direction: In, Time:15:31:08,768
TCP

Source port: 5679
Destination port: 1283

Raw Data:
0x0000 00 01 02 50 15 F5 00 00-1C D1 68 92 08 00 45 00 ...P.ö...Ñh'..E.
0x0010 00 28 4C C0 40 00 80 06-2C 29 C0 A8 00 64 C0 A8 .(LÄ@.€.,)À".dÄ"
0x0020 00 32 16 2F 05 03 AD 2A-10 AA 51 77 D4 39 50 10 .2./...-*.ªQwÔ9P.
0x0030 44 0A EB 2B 00 00 20 20-20 20 20 20 D.ë+..

=====
Packet #5, Direction: In, Time:15:31:15,798
TCP

Source port: 5679
Destination port: 1283

Raw Data:
0x0000 00 01 02 50 15 F5 00 00-1C D1 68 92 08 00 45 00 ...P.ö...Ñh'..E.
0x0010 00 40 4C C3 40 00 80 06-2C 0E C0 A8 00 64 C0 A8 .@LÄ@.€.,.Ä".dÄ"
0x0020 00 32 16 2F 05 03 AD 2A-10 AA 51 77 D4 39 50 18 .2./...-*.ªQwÔ9P.
0x0030 44 0A EA 5F 00 00 16 00-83 B2 80 B2 81 B2 86 B2 D.ê_....f²€²□²†²
0x0040 87 B2 84 B2 85 B2 8A B2-8B B2 82 B2 B2 B2 †²„²...²Š²<²,²²²

=====
Password: "1234567890"

=====
Packet #6, Direction: Out, Time:15:31:15,929
TCP

Source port: 1283
Destination port: 5679

Raw Data:
0x0000 00 00 1C D1 68 92 00 01-02 50 15 F5 08 00 45 00 ...Ñh'...P.ö..E.
0x0010 00 28 15 98 40 00 80 06-63 51 C0 A8 00 32 C0 A8 .(~@.€..cQÄ".2Ä"
0x0020 00 64 05 03 16 2F 51 77-D4 39 AD 2A 10 C2 50 10 .d.../QwÔ9-*.ÂP.
0x0030 62 40 CC DD 00 00 b@iÿ..

=====
Packet #7, Direction: Out, Time:15:31:15,939
TCP

Source port: 1283
Destination port: 5679

Raw Data:
0x0000 00 00 1C D1 68 92 00 01-02 50 15 F5 08 00 45 00 ...Ñh'...P.ö..E.
0x0010 00 2A 15 9B 40 00 80 06-63 4C C0 A8 00 32 C0 A8 .*.>@.€..cLÄ".2Ä"
0x0020 00 64 05 03 16 2F 51 77-D4 39 AD 2A 10 C2 50 18 .d.../QwÔ9-*.ÂP.
0x0030 62 40 CC D3 00 00 00- b@iÓ....

=====
Packet #8, Direction: In, Time:15:31:16,079
TCP

Source port: 5679
Destination port: 1283

Raw Data:
0x0000 00 01 02 50 15 F5 00 00-1C D1 68 92 08 00 45 00 ...P.ö...Ñh'..E.
0x0010 00 28 4C C4 40 00 80 06-2C 25 C0 A8 00 64 C0 A8 .(LÄ@.€.,%Ä".dÄ"
0x0020 00 32 16 2F 05 03 AD 2A-10 C2 51 77 D4 3B 50 10 .2./...-*.ÂQwÔ;P.
0x0030 44 08 EB 13 00 00 20 20-20 20 20 20 D.ë...

Packet #9, Direction: In, Time:15:32:19,921

TCP

Source port: 5679

Destination port: 1283

Raw Data:

```
0x0000 00 01 02 50 15 F5 00 00-1C D1 68 92 08 00 45 00 ...P.ö...Ñh'..E.
0x0010 00 2E 4C CE 40 00 80 06-2C 15 C0 A8 00 64 C0 A8 ..LÎ@.€. ,.À".dÀ"
0x0020 00 32 16 2F 05 03 AD 2A-10 C2 51 77 D4 3B 50 18 .2./...-*.ÂQwÔ;P.
0x0030 44 08 B0 A0 00 00 04 00-83 B2 B2 B2 D.° ....f2 2 2
```

Password: "1"

Packet #10, Direction: Out, Time:15:32:20,101

TCP

Source port: 1283

Destination port: 5679

Raw Data:

```
0x0000 00 00 1C D1 68 92 00 01-02 50 15 F5 08 00 45 00 ...Ñh'...P.ö..E.
0x0010 00 2A 15 9F 40 00 80 06-63 48 C0 A8 00 32 C0 A8 .* .ÿ@.€.cHÄ".2Ä"
0x0020 00 64 05 03 16 2F 51 77-D4 3B AD 2A 10 C8 50 18 .d.../QwÔ;-*.ËP.
0x0030 62 3A CC D1 00 00 00 00- b:îÑ....
```

Packet #11, Direction: In, Time:15:32:20,271

TCP

Source port: 5679

Destination port: 1283

Raw Data:

```
0x0000 00 01 02 50 15 F5 00 00-1C D1 68 92 08 00 45 00 ...P.ö...Ñh'..E.
0x0010 00 28 4C CF 40 00 80 06-2C 1A C0 A8 00 64 C0 A8 .(LÎ@.€. ,.À".dÀ"
0x0020 00 32 16 2F 05 03 AD 2A-10 C8 51 77 D4 3D 50 10 .2./...-*.ÈQwÔ=P.
0x0030 44 06 EB 0D 00 00 20 20-20 20 20 D.ë...
```

C.2. Password Packet Deconstructed

The password packet has the following fields:

Length: 2 Bytes		Password Characters: variable, multiple of 2 Bytes						Trailer: 2 Bytes		
low	hi	Char#1	B2 hex	Char#2	B2 hex	...	Char#n	B2 hex	B2 hex	B2 hex

C.3. Password Decoded

The packets show that the password is exchanged in clear text between the two programs and further probing with longer passwords allowed to compile the following table, to decode the hex values from the character fields of the password packet into ASCII characters.

0=82h	1=83h	2=80h	3=81h	4=86h	5=87h	6=84h	7=85h	8= 8Ah	9=8Bh
@=F2h	A=F3h	B=F0h	C=F1h	D=F6h	E=F7h	F=F4h	G=F5h	H=FAh	I=FBh
J=F8h	K=F9h	L=FEh	M=FFh	N=FCh	O=FDh	P=E2h	Q=E3h	R=E0h	S=E1h
T=E6h	U=E7h	V=E4h	W=E5h	X=EAh	Y=EBh	Z=E8h	[=E9h		
	a=D3h	b=D0h	c=D1h	d=D6h	e=D7h	f=D4h	g=D5h	h=DAh	i=DBh
j=D8h	k=D9h	l=DEh	m=DFh	n=DCh	o=DDh	p=C2h	q=C3h	r=C0h	s=C1h
t=C6h	u=C7h	v=C4h	w=C5h	x=CAh	y=CBh	z=C8h	{=C9h		

So there are groups of two characters that are mixed by always skipping two chars in the count:

0=82h	1=83h	2=80h	3=81h	4=86h	5=87h	6=84h	7=85h	8= 8Ah	9=8Bh
-------	-------	-------	-------	-------	-------	-------	-------	--------	-------

This work has deferred from looking further into this, as this shall only demonstrate that the password is not secured by any form of encryption when exchanged between the two ActiveSync application parts `wcescomm.exe` and `wcesmgr.exe`.

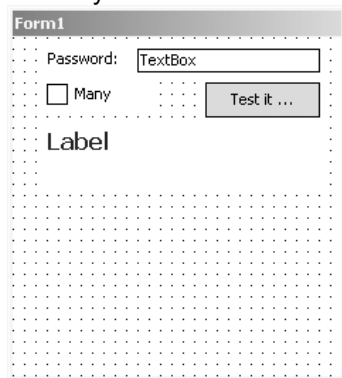
Appendix D – Source Code of Tools and other Code

During the course of this work I have used some trial applications, quickly written prototype programs written in eVB. The source code for the most interesting ones is provided in this Appendix.

D.1 CheckPassword

This program makes successive calls to the `CORE.DLL` API function `CheckPassword`, which allows an application to check if the supplied string matches the power-on password. This allows an application to use the power-on password for their own security aware parts, meaning that the user only needs one password, and it is centrally managed by the operating system. When the many-checkbox is checked, all passwords from “1000” to “9999” are tried. Recovering a lost PIN while the device is not yet locked.

GUI Layout:



eVB source code:

```
Public Declare Function CheckPassword Lib "Coredll" (ByVal _
wPassword As String) As Boolean

Private Sub Button_Click()
    Dim wpass As String
    If CheckBox.Value = 1 Then
        i = 1000
        Do While i < 10000
            wpass = i
            If (CheckPassword(wpass)) Then
                Label.Caption = wpass + " is correct!"
                TextBox.Text = wpass
                i = 99999
            Else
                Label.Caption = i
                i = i + 1
            End If
        Loop
    Else
        wpass = TextBox.Text
        If (CheckPassword(wpass)) Then
            Label.Caption = wpass + " is Correct!"
        Else
            Label.Caption = wpass + " not correct!"
        End If
    End If
End Sub

Private Sub Form_OKClick()
    App.End
End Sub
```

D.2 Show Script Engine Version

This program was used to display the version of the script engine supported by the PocketPC 2002 mobile device. Code comes from.

GUI Layout:



eVB source code [CLINICK2003]:

```
Option Explicit
```

```
Private Sub Button_Click()  
    TextBox.Text = scriptengine & " Version: " _  
        & scriptenginemajorversion & "." _  
        & scriptengineminorversion & " Build: " _  
        & scriptenginebuildversion
```

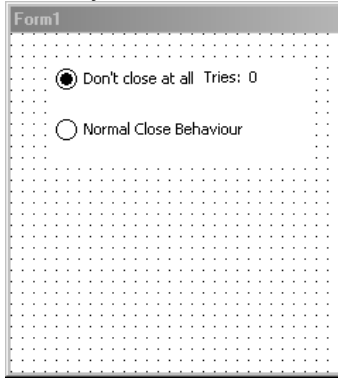
```
End Sub
```

```
Private Sub Form_OKClick()  
    App.End  
End Sub
```

D.3 Change OK / Close Button behaviour

This program was used to demonstrate that the application can control what happens when the user clicks on the Ok button. Even though the PocketPC application development guide [PPCLOGO] would like to leave the decision when to completely close an application to the operating system, the application can take control, it can also prohibit to be closed by the OK button. The latter means that the user must close it using the memory setting.

GUI Layout:



eVB source:

```
Option Explicit
```

```
Dim tries As Integer
```

```
Private Sub Form_OKClick()
    If NoClose.Value Then
        tries = tries + 1
        TryLabel.Caption = tries
    End If

    If NormalClose.Value Then
        App.End
    End If
End Sub
```

```
Private Sub NoClose_Click()
    NoClose.Value = True
End Sub
```

```
Private Sub NormalClose_Click()
    NormalClose.Value = True
End Sub
```


Appendix E – Details of the Testbeds

This appendix contains the list of the malicious samples and shows the structure of all testbeds and their test sets used in the test carried out. The six testbeds for the pre-test are listed in E.1, the four most recent testbeds used for the Pocket PC Scanner Test 2003-07 are listed in E.2.

E.1. Testbeds for the pre-Test

This gives an overview and lists the contents of aVTC's testbeds used in "Pocket PC Scanner Test 2003-05":

Script In-The-Wild testbeds:

- E.1.1) itwskri.002 (From Test 2002-12 frozen October 31,2001)
- E.1.2) scr_itw.021 (From Heureka III frozen January 31,2002)
- E.1.3) scr_itw.024 (From Heureka III frozen April 30,2002)

Macro In-The-Wild testbeds:

- E.1.4) itwmac.002 (From Test 2002-12 frozen October 31,2001)
- E.1.5) mac_itw.021 (From Heureka III frozen January 31,2002)
- E.1.6) mac_itw.024 (From Heureka III frozen April 30,2002)

The testbeds for the pre-test of the products on Pocket PC 2002 were taken from older and at that time (February 2003) currently running aVTC Tests.

E.1.1. itwskri.002

Testbed: SCRIPT-ITW from Test 2002-12
Internal name: itwskri.002

```
-----
malicious samples:      122
# of samples:           122
Size in bytes:          1,419,055
# of malware:           20
```

Detailed List (Path representing Name + # of samples):

```
-----
\itwskri.002\SET_01\jvs\k\kak\A 9
\itwskri.002\SET_01\vbs\H\HAPTIME\A 19
\itwskri.002\SET_01\vbs\f\freelink\A 4
\itwskri.002\SET_01\vbs\f\funny\A 2
\itwskri.002\SET_01\vbs\l\lovelett.er\A 55
\itwskri.002\SET_01\vbs\l\lovelett.er\bg 1
\itwskri.002\SET_01\vbs\l\lovelett.er\bj 2
\itwskri.002\SET_01\vbs\l\lovelett.er\c 2
\itwskri.002\SET_01\vbs\n\netlog\A 2
\itwskri.002\SET_01\vbs\s\SORRY\C 1
\itwskri.002\SET_01\vbs\s\SSIWG\U 1
\itwskri.002\SET_01\vbs\s\san\A 4
\itwskri.002\SET_01\vbs\s\san\b 2
\itwskri.002\SET_01\vbs\t\tam\A 3
\itwskri.002\SET_01\vbs\v\VBSWG\J 1
\itwskri.002\SET_01\vbs\v\VBSWG\K 1
\itwskri.002\SET_01\vbs\v\VBSWG\X 4
\itwskri.002\SET_01\vbs\v\VBSWG\Y 2
\itwskri.002\SET_01\vbs\v\VBSWG\Z 3
\itwskri.002\SET_01\vbs\v\valentin\A 4
```

E.1.2. scr_itw.021

Testbed: SCRIPT-ITW Heureka III January 2002
 Internal name: scr_itw.021

 malicious samples: 34
 # of samples: 34
 Size in bytes: 332,838
 # of malware: 25

Detailed List (Path representing Name + # of samples):

 \scr_itw.021\SET_01\jvs\Kak\gen 2
 \scr_itw.021\SET_01\vbs\Cuerpo\A 1
 \scr_itw.021\SET_01\vbs\FUNNY\A 1
 \scr_itw.021\SET_01\vbs\HAPTITUDE\A 3
 \scr_itw.021\SET_01\vbs\HAPTITUDE\D 3
 \scr_itw.021\SET_01\vbs\LOVELETT\A 1
 \scr_itw.021\SET_01\vbs\LOVELETT\AS 1
 \scr_itw.021\SET_01\vbs\LOVELETT\BG 1
 \scr_itw.021\SET_01\vbs\LOVELETT\BJ 1
 \scr_itw.021\SET_01\vbs\LOVELETT\C 1
 \scr_itw.021\SET_01\vbs\NETLOG\A 1
 \scr_itw.021\SET_01\vbs\SAN\A 2
 \scr_itw.021\SET_01\vbs\SAN\B 2
 \scr_itw.021\SET_01\vbs\SORRY\C 1
 \scr_itw.021\SET_01\vbs\SSIWG\U 1
 \scr_itw.021\SET_01\vbs\STAGES\A 1
 \scr_itw.021\SET_01\vbs\TAM\A 2
 \scr_itw.021\SET_01\vbs\TIMOFONI\A 1
 \scr_itw.021\SET_01\vbs\VALENTIN\A 2
 \scr_itw.021\SET_01\vbs\VBSWG\J 1
 \scr_itw.021\SET_01\vbs\VBSWG\K 1
 \scr_itw.021\SET_01\vbs\VBSWG\X 1
 \scr_itw.021\SET_01\vbs\VBSWG\Y 1
 \scr_itw.021\SET_01\vbs\VBSWG\Z 1
 \scr_itw.021\SET_01\vbs\VBSWG2\gen 1

E.1.3. scr_itw.024

Testbed: SCRIPT-ITW Heureka III April 2002
 Internal name: scr_itw.024

 malicious samples: 33
 # of samples: 33
 Size in bytes: 320,667
 # of malware: 24

Detailed List (Path representing Name + # of samples):

 \scr_itw.024\SET_01\jvs\KAK\gen 2


```

\scr_itw.024\SET_01\vbs\CUERPO\A 1
\scr_itw.024\SET_01\vbs\FUNNY\A 1
\scr_itw.024\SET_01\vbs\HAPTIME\A 3
\scr_itw.024\SET_01\vbs\HAPTIME\D 3
\scr_itw.024\SET_01\vbs\LOVELETT\A 1
\scr_itw.024\SET_01\vbs\LOVELETT\AS 1
\scr_itw.024\SET_01\vbs\LOVELETT\BG 1
\scr_itw.024\SET_01\vbs\LOVELETT\C 1
\scr_itw.024\SET_01\vbs\NETLOG\A 1
\scr_itw.024\SET_01\vbs\SAN\A 2
\scr_itw.024\SET_01\vbs\SAN\B 2
\scr_itw.024\SET_01\vbs\SORRY\C 1
\scr_itw.024\SET_01\vbs\SSIWG\U 1
\scr_itw.024\SET_01\vbs\STAGES\A 1
\scr_itw.024\SET_01\vbs\TAM\A 2
\scr_itw.024\SET_01\vbs\TIMOFONI\A 1
\scr_itw.024\SET_01\vbs\VALENTIN\A 2
\scr_itw.024\SET_01\vbs\VBSWG\J 1
\scr_itw.024\SET_01\vbs\VBSWG\K 1
\scr_itw.024\SET_01\vbs\VBSWG\X 1
\scr_itw.024\SET_01\vbs\VBSWG\Y 1
\scr_itw.024\SET_01\vbs\VBSWG\Z 1
\scr_itw.024\SET_01\vbs\VBSWG2\gen 1

```

E.1.4. itwmac.002

```

Testbed:                MACRO-ITW 2001 from Test 2002-12
Internal name:          itwmac.002
-----

```

```

malicious samples:     1337
# of samples:          1337
Size in bytes:         60,335,110
# of malware:          124

```

```

Detailed List (Path representing Name + # of samples):
-----

```

```

\itwmac.002\SET_01\wm\cap\a 57
\itwmac.002\SET_01\wm\concept\a 228
\itwmac.002\SET_01\wm\copycap\a 9
\itwmac.002\SET_01\wm\mental\a 7
\itwmac.002\SET_01\wm\wazzu\a 10
\itwmac.002\SET_02\w97m\assilem\c 5
\itwmac.002\SET_02\w97m\assilem\g 11
\itwmac.002\SET_02\w97m\bablas\a 7
\itwmac.002\SET_02\w97m\bablas\bh 5
\itwmac.002\SET_02\w97m\bleck\a 9
\itwmac.002\SET_02\w97m\bobo\b 7
\itwmac.002\SET_02\w97m\bobo\c 14
\itwmac.002\SET_02\w97m\bottra\a 9
\itwmac.002\SET_02\w97m\chack\b 8
\itwmac.002\SET_02\w97m\chack\h 6
\itwmac.002\SET_02\w97m\class\b 6

```

\itwmac.002\SET_02\w97m\class\bt 5
\itwmac.002\SET_02\w97m\class\d 11
\itwmac.002\SET_02\w97m\class\q 8
\itwmac.002\SET_02\w97m\claud\a 7
\itwmac.002\SET_02\w97m\coldape\a 10
\itwmac.002\SET_02\w97m\coldape\b 12
\itwmac.002\SET_02\w97m\cont\a 11
\itwmac.002\SET_02\w97m\db\a 10
\itwmac.002\SET_02\w97m\ded\b 14
\itwmac.002\SET_02\w97m\eight941\d 12
\itwmac.002\SET_02\w97m\eight941\e 11
\itwmac.002\SET_02\w97m\eight941\f 4
\itwmac.002\SET_02\w97m\ethan\a 9
\itwmac.002\SET_02\w97m\ethan\aw 6
\itwmac.002\SET_02\w97m\ethan\be 4
\itwmac.002\SET_02\w97m\ethan\d 10
\itwmac.002\SET_02\w97m\ethan\q 7
\itwmac.002\SET_02\w97m\ff\e 5
\itwmac.002\SET_02\w97m\flop\a 8
\itwmac.002\SET_02\w97m\footer\a 7
\itwmac.002\SET_02\w97m\groov\a 11
\itwmac.002\SET_02\w97m\groov\b 13
\itwmac.002\SET_02\wm\npad\a 31
\itwmac.002\SET_02\xm\laroux\a 13
\itwmac.002\SET_02\xm\laroux\ho 7
\itwmac.002\SET_03\w97m\nalp\a 5
\itwmac.002\SET_03\w97m\nono\a 6
\itwmac.002\SET_03\w97m\nottice\au 6
\itwmac.002\SET_03\w97m\onex\a 5
\itwmac.002\SET_03\w97m\onex\e 7
\itwmac.002\SET_03\w97m\opey\a 8
\itwmac.002\SET_03\w97m\opey\af 4
\itwmac.002\SET_03\w97m\opey\m 6
\itwmac.002\SET_03\w97m\ostrich\b 30
\itwmac.002\SET_03\w97m\panther\a 8
\itwmac.002\SET_03\w97m\panther\n 5
\itwmac.002\SET_03\w97m\pecas\b 7
\itwmac.002\SET_03\w97m\pri\a 8
\itwmac.002\SET_03\w97m\pri\b 15
\itwmac.002\SET_03\w97m\pri\q@mm 13
\itwmac.002\SET_03\w97m\proteced\a 7
\itwmac.002\SET_03\w97m\proverb\a 11
\itwmac.002\SET_03\w97m\replug\a 10
\itwmac.002\SET_03\w97m\seliuq\a 9
\itwmac.002\SET_03\w97m\service\a 2
\itwmac.002\SET_03\w97m\shepmah\a 7
\itwmac.002\SET_03\w97m\shore\d 6
\itwmac.002\SET_03\w97m\smac\d 12
\itwmac.002\SET_03\w97m\story\a 11
\itwmac.002\SET_03\w97m\surround\a 8
\itwmac.002\SET_03\w97m\thus\a 10
\itwmac.002\SET_03\w97m\thus\bh 5

\itwmac.002\SET_03\w97m\thus\c 9
\itwmac.002\SET_03\w97m\thus\de 6
\itwmac.002\SET_03\w97m\thus\ep 6
\itwmac.002\SET_03\w97m\thus\i 11
\itwmac.002\SET_03\w97m\thus\x 12
\itwmac.002\SET_03\w97m\titch\d 7
\itwmac.002\SET_03\w97m\titch\h 2
\itwmac.002\SET_03\w97m\title\a 5
\itwmac.002\SET_03\w97m\turn\a 11
\itwmac.002\SET_03\w97m\verlor\a 11
\itwmac.002\SET_03\w97m\vmppck1\by 4
\itwmac.002\SET_03\w97m\vmppck1\dd 5
\itwmac.002\SET_03\w97m\walker\d 6
\itwmac.002\SET_03\w97m>wrench\e 9
\itwmac.002\SET_03\w97m>wrench\i 5
\itwmac.002\SET_04\o97m\tristate\c_doc 9
\itwmac.002\SET_04\o97m\tristate\c_ppt 5
\itwmac.002\SET_04\o97m\tristate\c_xls 7
\itwmac.002\SET_04\w97m\Iis\E 2
\itwmac.002\SET_04\w97m\hope\a 5
\itwmac.002\SET_04\w97m\locale\a 8
\itwmac.002\SET_04\w97m\marker\bn 15
\itwmac.002\SET_04\w97m\marker\c 10
\itwmac.002\SET_04\w97m\marker\cb 6
\itwmac.002\SET_04\w97m\marker\d 11
\itwmac.002\SET_04\w97m\marker\ef 11
\itwmac.002\SET_04\w97m\marker\fi 6
\itwmac.002\SET_04\w97m\marker\o 10
\itwmac.002\SET_04\w97m\marker\q 12
\itwmac.002\SET_04\w97m\melissa\a@mm 13
\itwmac.002\SET_04\w97m\melissa\o@mm 13
\itwmac.002\SET_04\w97m\melissa\u@mm 8
\itwmac.002\SET_04\w97m\myna\b 8
\itwmac.002\SET_04\w97m\myna\c 7
\itwmac.002\SET_04\w97m\nagem\a 8
\itwmac.002\SET_04\x97m\adn\b 3
\itwmac.002\SET_04\x97m\barisada\a 14
\itwmac.002\SET_04\x97m\barisada\b 8
\itwmac.002\SET_04\x97m\barisada\d 6
\itwmac.002\SET_04\x97m\barisada\g 12
\itwmac.002\SET_04\x97m\divi\a 13
\itwmac.002\SET_04\x97m\divi\b 10
\itwmac.002\SET_04\x97m\divi\d 8
\itwmac.002\SET_04\x97m\divi\n 7
\itwmac.002\SET_04\x97m\jini\a 7
\itwmac.002\SET_04\x97m\laroux\a 6
\itwmac.002\SET_04\x97m\laroux\dx 3
\itwmac.002\SET_04\x97m\laroux\e 3
\itwmac.002\SET_04\x97m\laroux\ho 7
\itwmac.002\SET_04\x97m\laroux\ku 12
\itwmac.002\SET_04\x97m\laroux\mv 8
\itwmac.002\SET_04\x97m\squared\a 10

```
\itwmac.002\SET_04\x97m\squared\b 10
\itwmac.002\SET_04\x97m\vcx\a 9
\itwmac.002\SET_04\x97m\yawn\a 15
\itwmac.002\SET_04\xf\sic\a 8
```

E.1.5. mac_itw.021

```
Testbed:                MACRO-ITW Heureka III January 2002
Internal name:          mac_itw.021
```

```
-----
malicious samples:     224
# of samples:          224
Size in bytes:         9,101,824
# of malware:          115
```

Detailed List (Path representing Name + # of samples):

```
-----
\mac_itw.021\SET_01\pp97m\Tristate\C 1
\mac_itw.021\SET_01\w97m\Assilem\G 2
\mac_itw.021\SET_01\w97m\Bablas\A 2
\mac_itw.021\SET_01\w97m\Bablas\BH 2
\mac_itw.021\SET_01\w97m\Bleck\A 2
\mac_itw.021\SET_01\w97m\Bobo\C 2
\mac_itw.021\SET_01\w97m\Bottra\A 2
\mac_itw.021\SET_01\w97m\Chack\B 2
\mac_itw.021\SET_01\w97m\Chack\H 2
\mac_itw.021\SET_01\w97m\Class\B 2
\mac_itw.021\SET_01\w97m\Class\BT 2
\mac_itw.021\SET_01\w97m\Class\D-DB-DC 2
\mac_itw.021\SET_01\w97m\Class\Q 2
\mac_itw.021\SET_01\w97m\Claud\A 2
\mac_itw.021\SET_01\w97m\ColdApe\A 2
\mac_itw.021\SET_01\w97m\ColdApe\B 2
\mac_itw.021\SET_01\w97m\Cont\A 2
\mac_itw.021\SET_01\w97m\DB\A 2
\mac_itw.021\SET_01\w97m\Ded\B 2
\mac_itw.021\SET_01\w97m\Eight941\D 2
\mac_itw.021\SET_01\w97m\Eight941\E 2
\mac_itw.021\SET_01\w97m\Eight941\F 2
\mac_itw.021\SET_01\w97m\Ethan\A 2
\mac_itw.021\SET_01\w97m\Ethan\AK 2
\mac_itw.021\SET_01\w97m\Ethan\AW 2
\mac_itw.021\SET_01\w97m\Ethan\BE 2
\mac_itw.021\SET_01\w97m\Ethan\D 2
\mac_itw.021\SET_01\w97m\Ethan\Q 2
\mac_itw.021\SET_01\w97m\FF\E 2
\mac_itw.021\SET_01\w97m\Flop\A 2
\mac_itw.021\SET_01\w97m\Footer\A 2
\mac_itw.021\SET_01\w97m\Groov\A 2
\mac_itw.021\SET_01\w97m\Groov\B 2
\mac_itw.021\SET_01\w97m\Hope\A 2
\mac_itw.021\SET_01\w97m\IIS\A 2
```

\mac_itw.021\SET_01\w97m\Locale\A 2
\mac_itw.021\SET_01\w97m\Marker\BN 2
\mac_itw.021\SET_01\w97m\Marker\C-AP-DJ 2
\mac_itw.021\SET_01\w97m\Marker\CB 2
\mac_itw.021\SET_01\w97m\Marker\D 2
\mac_itw.021\SET_01\w97m\Marker\O-AK 4
\mac_itw.021\SET_01\w97m\Marker\Q 2
\mac_itw.021\SET_01\w97m\Melissa\A__mm 2
\mac_itw.021\SET_01\w97m\Melissa\O__mm 2
\mac_itw.021\SET_01\w97m\Melissa\U__mm 2
\mac_itw.021\SET_01\w97m\Myna\B 2
\mac_itw.021\SET_01\w97m\Myna\C 2
\mac_itw.021\SET_01\w97m\Nagem\A 2
\mac_itw.021\SET_01\w97m\Nalp\A 2
\mac_itw.021\SET_01\w97m\Nono\A 2
\mac_itw.021\SET_01\w97m\Nottice\AU 2
\mac_itw.021\SET_01\w97m\Onex\A 2
\mac_itw.021\SET_01\w97m\Onex\E 2
\mac_itw.021\SET_01\w97m\Opey\A 2
\mac_itw.021\SET_01\w97m\Opey\M 2
\mac_itw.021\SET_01\w97m\Ostrich\B 2
\mac_itw.021\SET_01\w97m\Panther\A 2
\mac_itw.021\SET_01\w97m\Panther\N 2
\mac_itw.021\SET_01\w97m\Pecas\B 2
\mac_itw.021\SET_01\w97m\Pri\A 2
\mac_itw.021\SET_01\w97m\Pri\Q__mm 2
\mac_itw.021\SET_01\w97m\Proverb\A 2
\mac_itw.021\SET_01\w97m\Replug\A 2
\mac_itw.021\SET_01\w97m\SERVICE\A 2
\mac_itw.021\SET_01\w97m\Seliuq\A 2
\mac_itw.021\SET_01\w97m\Shepmah\A 2
\mac_itw.021\SET_01\w97m\Smac\D 2
\mac_itw.021\SET_01\w97m\Story\A 2
\mac_itw.021\SET_01\w97m\Surround\A 2
\mac_itw.021\SET_01\w97m\Thus\A 2
\mac_itw.021\SET_01\w97m\Thus\BH 2
\mac_itw.021\SET_01\w97m\Thus\C 2
\mac_itw.021\SET_01\w97m\Thus\DE 2
\mac_itw.021\SET_01\w97m\Thus\EP 2
\mac_itw.021\SET_01\w97m\Thus\I 2
\mac_itw.021\SET_01\w97m\Thus\X 2
\mac_itw.021\SET_01\w97m\Titch\D 2
\mac_itw.021\SET_01\w97m\Titch\H 2
\mac_itw.021\SET_01\w97m\Title\A 2
\mac_itw.021\SET_01\w97m\Tristate\C 2
\mac_itw.021\SET_01\w97m\Turn\A 2
\mac_itw.021\SET_01\w97m\VMPCCK1\BY 2
\mac_itw.021\SET_01\w97m\VMPCCK1\DD 2
\mac_itw.021\SET_01\w97m\Verlor\A 2
\mac_itw.021\SET_01\w97m\Walker\D 2
\mac_itw.021\SET_01\w97m\Walker\E 2
\mac_itw.021\SET_01\w97m\Wrench\E 2

```

\mac_itw.021\SET_01\w97m\Wrench\I 2
\mac_itw.021\SET_01\wm\CAP\A 2
\mac_itw.021\SET_01\wm\Concept\A 2
\mac_itw.021\SET_01\wm\CopyCap\A 2
\mac_itw.021\SET_01\wm\Mental\A 2
\mac_itw.021\SET_01\wm\Npad\A 2
\mac_itw.021\SET_01\wm\Wazzu\A 2
\mac_itw.021\SET_01\x97m\Adn\B 1
\mac_itw.021\SET_01\x97m\Barisada\A 2
\mac_itw.021\SET_01\x97m\Barisada\B 2
\mac_itw.021\SET_01\x97m\Barisada\D 2
\mac_itw.021\SET_01\x97m\Barisada\G 2
\mac_itw.021\SET_01\x97m\Divi\A 2
\mac_itw.021\SET_01\x97m\Divi\B 2
\mac_itw.021\SET_01\x97m\Divi\D 2
\mac_itw.021\SET_01\x97m\Laroux\A 1
\mac_itw.021\SET_01\x97m\Laroux\AA 2
\mac_itw.021\SET_01\x97m\Laroux\DX 1
\mac_itw.021\SET_01\x97m\Laroux\E 1
\mac_itw.021\SET_01\x97m\Laroux\HO 1
\mac_itw.021\SET_01\x97m\Laroux\KU 2
\mac_itw.021\SET_01\x97m\PTH\D 2
\mac_itw.021\SET_01\x97m\Squared\A 2
\mac_itw.021\SET_01\x97m\Squared\B 2
\mac_itw.021\SET_01\x97m\Tristate\C 1
\mac_itw.021\SET_01\x97m\Yawn\A 2
\mac_itw.021\SET_01\xm\Laroux\A 1
\mac_itw.021\SET_01\xm\Laroux\HO 2

```

E.1.6. mac_itw.024

```

Testbed:                MACRO-ITW Heureka III April 2002
Internal name:          mac_itw.024

```

```

-----
malicious samples:     189
# of samples:          189
Size in bytes:         7,512,064
# of malware:          99

```

```

Detailed List (Path representing Name + # of samples):

```

```

-----
\mac_itw.024\SET_01\pp97m\Tristate\C 1
\mac_itw.024\SET_01\w97m\Assilem\G 2
\mac_itw.024\SET_01\w97m\Bablas\A 2
\mac_itw.024\SET_01\w97m\Bablas\BH 2
\mac_itw.024\SET_01\w97m\Bleck\A 2
\mac_itw.024\SET_01\w97m\Bottra\A 2
\mac_itw.024\SET_01\w97m\Chack\B 2
\mac_itw.024\SET_01\w97m\Class\B 2
\mac_itw.024\SET_01\w97m\Class\Q 2
\mac_itw.024\SET_01\w97m\Claud\A 2
\mac_itw.024\SET_01\w97m\ColdApe\A 2
\mac_itw.024\SET_01\w97m\ColdApe\B 2

```

\mac_itw.024\SET_01\w97m\Cont\A 2
\mac_itw.024\SET_01\w97m\DB\A 2
\mac_itw.024\SET_01\w97m\Ded\B 2
\mac_itw.024\SET_01\w97m\Eight941\D 2
\mac_itw.024\SET_01\w97m\Eight941\E 2
\mac_itw.024\SET_01\w97m\Eight941\F 2
\mac_itw.024\SET_01\w97m\Ethan\A 2
\mac_itw.024\SET_01\w97m\Ethan\AK 2
\mac_itw.024\SET_01\w97m\Ethan\AW 2
\mac_itw.024\SET_01\w97m\Ethan\BE 2
\mac_itw.024\SET_01\w97m\Ethan\Q 2
\mac_itw.024\SET_01\w97m\FF\E 2
\mac_itw.024\SET_01\w97m\Flop\A 2
\mac_itw.024\SET_01\w97m\Footer\A 2
\mac_itw.024\SET_01\w97m\Groov\A 2
\mac_itw.024\SET_01\w97m\Groov\B 2
\mac_itw.024\SET_01\w97m\Hope\A 2
\mac_itw.024\SET_01\w97m\Locale\A 2
\mac_itw.024\SET_01\w97m\Marker\BN 2
\mac_itw.024\SET_01\w97m\Marker\D 2
\mac_itw.024\SET_01\w97m\Marker\Q 2
\mac_itw.024\SET_01\w97m\Melissa\A__mm 2
\mac_itw.024\SET_01\w97m\Melissa\O__mm 2
\mac_itw.024\SET_01\w97m\Myna\B 2
\mac_itw.024\SET_01\w97m\Myna\C 2
\mac_itw.024\SET_01\w97m\Nagem\A 2
\mac_itw.024\SET_01\w97m\Nono\A 2
\mac_itw.024\SET_01\w97m\Nottice\AU 2
\mac_itw.024\SET_01\w97m\Onex\A 2
\mac_itw.024\SET_01\w97m\Onex\E 2
\mac_itw.024\SET_01\w97m\Opey\A 2
\mac_itw.024\SET_01\w97m\Opey\M 2
\mac_itw.024\SET_01\w97m\Ostrich\B 2
\mac_itw.024\SET_01\w97m\Panther\A 2
\mac_itw.024\SET_01\w97m\Pecas\B 2
\mac_itw.024\SET_01\w97m\Pri\A 2
\mac_itw.024\SET_01\w97m\Pri\Q__mm 2
\mac_itw.024\SET_01\w97m\Proverb\A 2
\mac_itw.024\SET_01\w97m\Replog\A 2
\mac_itw.024\SET_01\w97m\Seliuq\A 2
\mac_itw.024\SET_01\w97m\Shepmah\A 2
\mac_itw.024\SET_01\w97m\Smac\D 2
\mac_itw.024\SET_01\w97m\Story\A 2
\mac_itw.024\SET_01\w97m\Surround\A 2
\mac_itw.024\SET_01\w97m\Thus\A 2
\mac_itw.024\SET_01\w97m\Thus\BH 2
\mac_itw.024\SET_01\w97m\Thus\C 2
\mac_itw.024\SET_01\w97m\Thus\DE 2
\mac_itw.024\SET_01\w97m\Thus\EP 2
\mac_itw.024\SET_01\w97m\Thus\I 2
\mac_itw.024\SET_01\w97m\Thus\X 2
\mac_itw.024\SET_01\w97m\Titch\D 2

\mac_itw.024\SET_01\w97m\Titch\H 2
\mac_itw.024\SET_01\w97m>Title\A 2
\mac_itw.024\SET_01\w97m\Tristate\C 2
\mac_itw.024\SET_01\w97m\Turn\A 2
\mac_itw.024\SET_01\w97m\VMPCCK1\BY 2
\mac_itw.024\SET_01\w97m\VMPCCK1\DD 2
\mac_itw.024\SET_01\w97m\Verlor\A 2
\mac_itw.024\SET_01\w97m\Walker\E 2
\mac_itw.024\SET_01\w97m\Wrench\E 2
\mac_itw.024\SET_01\w97m\Wrench\I 2
\mac_itw.024\SET_01\wm\CAP\A 2
\mac_itw.024\SET_01\wm\Concept\A 2
\mac_itw.024\SET_01\wm\CopyCap\A 2
\mac_itw.024\SET_01\wm\Mental\A 2
\mac_itw.024\SET_01\wm\Npad\A 2
\mac_itw.024\SET_01\wm\Wazzu\A 2
\mac_itw.024\SET_01\x97m\Adn\B 1
\mac_itw.024\SET_01\x97m\Barisada\A 2
\mac_itw.024\SET_01\x97m\Barisada\B 2
\mac_itw.024\SET_01\x97m\Barisada\D 2
\mac_itw.024\SET_01\x97m\Divi\A 2
\mac_itw.024\SET_01\x97m\Divi\B 2
\mac_itw.024\SET_01\x97m\Divi\D 2
\mac_itw.024\SET_01\x97m\Laroux\A 1
\mac_itw.024\SET_01\x97m\Laroux\AA 2
\mac_itw.024\SET_01\x97m\Laroux\DX 1
\mac_itw.024\SET_01\x97m\Laroux\E 1
\mac_itw.024\SET_01\x97m\Laroux\HO 1
\mac_itw.024\SET_01\x97m\PTH\D 2
\mac_itw.024\SET_01\x97m\Squared\A 2
\mac_itw.024\SET_01\x97m\Squared\B 2
\mac_itw.024\SET_01\x97m\Tristate\C 1
\mac_itw.024\SET_01\x97m\Yawn\A 2
\mac_itw.024\SET_01\x97m\Sic\A 1
\mac_itw.024\SET_01\xm\Laroux\A 1

E.2. Testbeds for the Pocket PC Scanner Test 2003-07

This gives an overview and lists the contents of aVTC's testbeds used in "Pocket PC Scanner Test 2003-07":

Script In-The-Wild testbeds:

- E.2.1) scr_itw.304 (Testbed containing viruses that were ITW from 11/2001 till 12/2002)
- E.2.2) scr_itw.dec02 (Testbed containing viruses that were ITW December 2002)

Macro In-The-Wild testbeds:

- E.2.3) mac_itw.304 (Testbed containing viruses that were ITW from 11/2001 till 12/2002)
- E.2.4) mac_itw.dec02 (Testbed containing viruses that were ITW December 2002)

The wildlist organization compiles a monthly list of ITW viruses, this list is used to select the viruses for the aVTC ITW testbeds. The wildlist from December 2002 can be found at the end of Appendix C, in chapter E.3.

The first script (scr_itw.304) and the first macro testbed (mac_itw.304) include all ITW viruses that occurred during a 13 month timeframe, allowing to measure the detection of quite old and quite new viruses.

The second testbeds for script (scr_itw.dec02) and macro (mac_itw.dec02) were taken from the currently compiled aVTC testbeds for the upcoming aVTC Test 2003-09. They were frozen 31st December 2002 and only include samples of viruses that were reported to be ITW in December 2002.

E.2.1. scr_itw.304

```
Testbed:                SCRIPT-ITW from 11/2001 till 12/2002
Internal name:          scr_itw.304
```

```
-----
malicious samples:     201
# of samples:          201
Size in bytes:         2,485,690
# of malware:          30
```

Detailed List (Path representing Name + # of samples):

```
-----
\scr_itw.304\jvs\K\Kak\A 9
\scr_itw.304\vbs\c\Cuerpo\A_MM 5
\scr_itw.304\vbs\d\Daira\A_MM 2
\scr_itw.304\vbs\f\Freelink\A_MM 4
\scr_itw.304\vbs\f\FUNNY\A 2
\scr_itw.304\vbs\h\HAPTITUDE\A_MM 30
\scr_itw.304\vbs\h\HAPTITUDE\D_M 10
\scr_itw.304\vbs\l\LOVELETT\AS 1
\scr_itw.304\vbs\l\LOVELETT\A_MM 73
\scr_itw.304\vbs\l\LOVELETT\BG 1
\scr_itw.304\vbs\l\LOVELETT\BJ 2
\scr_itw.304\vbs\l\LOVELETT\C 3
\scr_itw.304\vbs\n\NETLOG\A 11
\scr_itw.304\vbs\r\REDLOF\A_M 5
\scr_itw.304\vbs\s\SAN\A_M 7
\scr_itw.304\vbs\s\SAN\B 2
\scr_itw.304\vbs\s\Sorry\C 1
\scr_itw.304\vbs\s\SSIWG\U 1
\scr_itw.304\vbs\s\STAGES\A 1
```

```

\scr_itw.304\vbs\s\STAGES\A_14559 1
\scr_itw.304\vbs\s\STAGES\A_2543 1
\scr_itw.304\vbs\t\TAM\A_M 9
\scr_itw.304\vbs\t\TIMOFONI\A 1
\scr_itw.304\vbs\v\VALENTIN\A_MM 5
\scr_itw.304\vbs\v\VBSWG\AQ 1
\scr_itw.304\vbs\v\VBSWG\J_MM 3
\scr_itw.304\vbs\v\VBSWG\K_MM 1
\scr_itw.304\vbs\v\VBSWG\X 4
\scr_itw.304\vbs\v\VBSWG\Y 2
\scr_itw.304\vbs\v\VBSWG\Z 3

```

E.2.2. scr_itw.dec02

```

Test Bed:                SCRIPT-ITW from Test 2003-09
Internal name:           scr_itw.dec02
malicious samples:      178
# of samples:            178
Size in bytes:           2.196.566
# of malware:            22

```

Detailed List (Path representing Name + # of samples):

```

-----
\scr_itw.dec02\vbs\v\VBSWG\Z 3
\scr_itw.dec02\vbs\v\VBSWG\X 4
\scr_itw.dec02\vbs\v\VBSWG\K_MM 1
\scr_itw.dec02\vbs\v\VBSWG\J_MM 3
\scr_itw.dec02\vbs\v\VBSWG\AQ 1
\scr_itw.dec02\vbs\t\TAM\A_M 9
\scr_itw.dec02\vbs\s\STAGES\A_2543 1
\scr_itw.dec02\vbs\s\STAGES\A_14559 1
\scr_itw.dec02\vbs\s\STAGES\A 1
\scr_itw.dec02\vbs\s\SSIWG\A 4
\scr_itw.dec02\vbs\s\SSIWG\U 1
\scr_itw.dec02\vbs\r\REDLOF\A_M 5
\scr_itw.dec02\vbs\n\NETLOG\A 11
\scr_itw.dec02\vbs\l\LOVELETT\C 3
\scr_itw.dec02\vbs\l\LOVELETT\BG 1
\scr_itw.dec02\vbs\l\LOVELETT\A_MM 73
\scr_itw.dec02\vbs\l\LOVELETT\AS 1
\scr_itw.dec02\vbs\h\HAPTIME\D_M 10
\scr_itw.dec02\vbs\h\HAPTIME\A_MM 30
\scr_itw.dec02\vbs\f\Freelink\A_MM 4
\scr_itw.dec02\vbs\d\Daira\A_MM 2
\scr_itw.dec02\jvs\K\Kak\A 9

```

E.2.3. mac_itw.304

Testbed: MACRO-ITW from 11/2001 till 12/2002
 Internal name: mac_itw.304

```
-----
malicious samples:      277
# of samples:           277
Size in bytes:          13,719,985
# of malware:           118
```

Detailed List (Path representing Name + # of samples):

```
-----
\mac_itw.304\PP97M\TRISTATE\C 1
\mac_itw.304\W97M\ASSILEM\A 2
\mac_itw.304\W97M\ASSILEM\G 2
\mac_itw.304\W97M\BABLAS\A 2
\mac_itw.304\W97M\BABLAS\BH 2
\mac_itw.304\W97M\BLECK\A 2
\mac_itw.304\W97M\BOBO\C 2
\mac_itw.304\W97M\BOTTRA\A 2
\mac_itw.304\W97M\BOTTRA\C 5
\mac_itw.304\W97M\CHACK\B 3
\mac_itw.304\W97M\CHACK\H 2
\mac_itw.304\W97M\CLASS\B 2
\mac_itw.304\W97M\CLASS\BT 2
\mac_itw.304\W97M\CLASS\D 2
\mac_itw.304\W97M\CLASS\Q 2
\mac_itw.304\W97M\CLAUD\A 2
\mac_itw.304\W97M\COLDAPE\A 2
\mac_itw.304\W97M\COLDAPE\B 11
\mac_itw.304\W97M\CONT\A 2
\mac_itw.304\W97M\DB\A 2
\mac_itw.304\W97M\DED\B 2
\mac_itw.304\W97M\EIGHT941\D 2
\mac_itw.304\W97M\EIGHT941\E 2
\mac_itw.304\W97M\EIGHT941\F 2
\mac_itw.304\W97M\ETHAN\A 4
\mac_itw.304\W97M\ETHAN\AK 2
\mac_itw.304\W97M\ETHAN\AW 2
\mac_itw.304\W97M\ETHAN\BE 4
\mac_itw.304\W97M\ETHAN\D 2
\mac_itw.304\W97M\ETHAN\Q 2
\mac_itw.304\W97M\FF\E 2
\mac_itw.304\W97M\FLOP\A 2
\mac_itw.304\W97M\FOOTER\A 2
\mac_itw.304\W97M\GROOV\A 2
\mac_itw.304\W97M\GROOV\B 2
\mac_itw.304\W97M\HOPE\A 2
\mac_itw.304\W97M\IIS\A 11
\mac_itw.304\W97M\LOCALE\A 2
\mac_itw.304\W97M\MARKER\BN 2
\mac_itw.304\W97M\MARKER\C 2
```

\mac_itw.304\W97M\MARKER\CB 4
\mac_itw.304\W97M\MARKER\D 2
\mac_itw.304\W97M\MARKER\O 4
\mac_itw.304\W97M\MARKER\Q 4
\mac_itw.304\W97M\MELISSA\A_MM 6
\mac_itw.304\W97M\MELISSA\O_MM 2
\mac_itw.304\W97M\MELISSA\U_MM 2
\mac_itw.304\W97M\MYNA\B 2
\mac_itw.304\W97M\MYNA\C 2
\mac_itw.304\W97M\NAGEM\A 2
\mac_itw.304\W97M\NALP\A 2
\mac_itw.304\W97M\NONO\A 2
\mac_itw.304\W97M\NOTTICE\AU 2
\mac_itw.304\W97M\ONEX\A 2
\mac_itw.304\W97M\ONEX\E 2
\mac_itw.304\W97M\OPEY\A 2
\mac_itw.304\W97M\OPEY\M 2
\mac_itw.304\W97M\OSTRICH\B 2
\mac_itw.304\W97M\PANTHER\A 2
\mac_itw.304\W97M\PANTHER\N 2
\mac_itw.304\W97M\PECAS\B 2
\mac_itw.304\W97M\PRI\A 2
\mac_itw.304\W97M\PRI\Q_MM 2
\mac_itw.304\W97M\PROVERB\A 2
\mac_itw.304\W97M\REPLOG\A 2
\mac_itw.304\W97M\SELIUQ\A 2
\mac_itw.304\W97M\SERVICE\A 4
\mac_itw.304\W97M\SHEPMAH\A 2
\mac_itw.304\W97M\SMAC\D 2
\mac_itw.304\W97M\STORY\A 2
\mac_itw.304\W97M\SURROUND\A 2
\mac_itw.304\W97M\THUS\A 2
\mac_itw.304\W97M\THUS\BH 2
\mac_itw.304\W97M\THUS\C 2
\mac_itw.304\W97M\THUS\DE 2
\mac_itw.304\W97M\THUS\EP 2
\mac_itw.304\W97M\THUS\I 3
\mac_itw.304\W97M\THUS\X 2
\mac_itw.304\W97M\TITCH\D 2
\mac_itw.304\W97M\TITCH\H 2
\mac_itw.304\W97M\TITLE\A 2
\mac_itw.304\W97M\TRISTATE\C 2
\mac_itw.304\W97M\TURN\A 2
\mac_itw.304\W97M\VERLOR\A 7
\mac_itw.304\W97M\VMPCCK1\BY 2
\mac_itw.304\W97M\VMPCCK1\DD 2
\mac_itw.304\W97M\WALKER\D 2
\mac_itw.304\W97M\WALKER\E 2
\mac_itw.304\W97M\WRENCH\E 2
\mac_itw.304\W97M\WRENCH\I 4
\mac_itw.304\WM\CAP\A 2
\mac_itw.304\WM\CONCEPT\A 2

```

\mac_itw.304\WM\COPYCAP\A 2
\mac_itw.304\WM\MENTAL\A 2
\mac_itw.304\WM\NPAD\A 2
\mac_itw.304\WM\WAZZU\A 2
\mac_itw.304\X97M\ADN\B 1
\mac_itw.304\X97M\BARISADA\A 2
\mac_itw.304\X97M\BARISADA\B 2
\mac_itw.304\X97M\BARISADA\D 2
\mac_itw.304\X97M\BARISADA\G 2
\mac_itw.304\X97M\DIVI\A 2
\mac_itw.304\X97M\DIVI\B 2
\mac_itw.304\X97M\DIVI\D 2
\mac_itw.304\X97M\LAROUX\A 1
\mac_itw.304\X97M\LAROUX\AA 2
\mac_itw.304\X97M\LAROUX\DX 3
\mac_itw.304\X97M\LAROUX\E 1
\mac_itw.304\X97M\LAROUX\HO 2
\mac_itw.304\X97M\LAROUX\KU 2
\mac_itw.304\X97M\PTH\D 2
\mac_itw.304\X97M\SQUARED\A 2
\mac_itw.304\X97M\SQUARED\B 2
\mac_itw.304\X97M\TRISTATE\C 2
\mac_itw.304\X97M\YAWN\A 2
\mac_itw.304\XF\SIC\A 1
\mac_itw.304\XM\LAROUX\A 1
\mac_itw.304\XM\LAROUX\HO 2

```

E.2.4. mac_itw.dec02

```

Test Bed:          MACRO-ITW from Test 2003-09
Internal name:     mac_itw.dec02

```

```

-----
Internal name:     mac_itw.dec02
-----

```

```

malicious samples: 976
# of samples:      976
Size in bytes:     47.731.639
# of malware:      76

```

```

Detailed List (Path representing Name + # of samples):
-----

```

```

\mac_itw.dec02\set07\W97M\WRENCH\E 9
\mac_itw.dec02\set07\W97M\WALKER\E 8
\mac_itw.dec02\set07\W97M\VMPCk1\DD 5
\mac_itw.dec02\set07\W97M\VMPCk1\BY 4
\mac_itw.dec02\set07\W97M\VERLOR\A 16
\mac_itw.dec02\set07\W97M\TITLE\A 5
\mac_itw.dec02\set07\W97M\TITCH\H 2
\mac_itw.dec02\set07\W97M\TITCH\D 7
\mac_itw.dec02\set07\W97M\THUS\X 12
\mac_itw.dec02\set07\W97M\THUS\I 12
\mac_itw.dec02\set07\W97M\THUS\C 9
\mac_itw.dec02\set07\W97M\THUS\A 10

```

\mac_itw.dec02\set06\W97M\SURROUND\A 8
\mac_itw.dec02\set06\W97M\STORY\A 11
\mac_itw.dec02\set06\W97M\SMAC\D 12
\mac_itw.dec02\set06\W97M\SERVICE\A 4
\mac_itw.dec02\set06\W97M\PROVERB\A 11
\mac_itw.dec02\set06\W97M\PRI\Q_MM 15
\mac_itw.dec02\set06\W97M\PRI\A 8
\mac_itw.dec02\set06\W97M\PANTHER\A 8
\mac_itw.dec02\set06\W97M\OSTRICH\B 30
\mac_itw.dec02\set06\W97M\OPEY\M 6
\mac_itw.dec02\set06\W97M\OPEY\A 8
\mac_itw.dec02\set06\W97M\ONEX\E 7
\mac_itw.dec02\set06\W97M\NONO\A 6
\mac_itw.dec02\set06\W97M\NAGEM\A 8
\mac_itw.dec02\set06\W97M\MYNA\C 7
\mac_itw.dec02\set06\W97M\MYNA\B 8
\mac_itw.dec02\set05\W97M\MELISSA\O_MM 15
\mac_itw.dec02\set05\W97M\MELISSA\A_MM 19
\mac_itw.dec02\set05\W97M\MARKER\Q 14
\mac_itw.dec02\set05\W97M\MARKER\O 10
\mac_itw.dec02\set05\W97M\MARKER\D 11
\mac_itw.dec02\set05\W97M\MARKER\C 10
\mac_itw.dec02\set05\W97M\MARKER\BN 15
\mac_itw.dec02\set05\W97M\LOCALE\A 8
\mac_itw.dec02\set05\W97M\GROOV\B 13
\mac_itw.dec02\set05\W97M\GROOV\A 11
\mac_itw.dec02\set05\W97M\FLOP\A 8
\mac_itw.dec02\set05\W97M\ETHAN\Q 7
\mac_itw.dec02\set05\W97M\ETHAN\BE 6
\mac_itw.dec02\set05\W97M\ETHAN\AK 8
\mac_itw.dec02\set05\W97M\ETHAN\A 11
\mac_itw.dec02\set05\W97M\EIGHT941\E 11
\mac_itw.dec02\set05\W97M\EIGHT941\D 12
\mac_itw.dec02\set04\WM\CONCEPT\A 47
\mac_itw.dec02\set04\W97M\COLDAPE\B 21
\mac_itw.dec02\set04\W97M\COLDAPE\A 10
\mac_itw.dec02\set04\W97M\CLASS\Q 8
\mac_itw.dec02\set04\W97M\CLASS\D 11
\mac_itw.dec02\set04\W97M\CLASS\B 6
\mac_itw.dec02\set04\W97M\CHACK\B 9
\mac_itw.dec02\set04\W97M\BOTTRA\C 14
\mac_itw.dec02\set04\W97M\BOTTRA\A 9
\mac_itw.dec02\set04\W97M\BLECK\A 9
\mac_itw.dec02\set04\W97M\BABLAS\A 7
\mac_itw.dec02\set04\W97M\ASSILEM\A 6
\mac_itw.dec02\set03\WM\CONCEPT\A 128
\mac_itw.dec02\set02\WM\WAZZU\A 10
\mac_itw.dec02\set02\WM\NPAD\A 31
\mac_itw.dec02\set02\WM\COPYCAP\A 9
\mac_itw.dec02\set02\WM\CONCEPT\A 53
\mac_itw.dec02\set02\WM\CAP\A 57
\mac_itw.dec02\set01\XM\LAROUX\A 11

```
\mac_itw.dec02\set01\X97M\YAWN\A 11  
\mac_itw.dec02\set01\X97M\LAROUX\HO 7  
\mac_itw.dec02\set01\X97M\LAROUX\E 2  
\mac_itw.dec02\set01\X97M\LAROUX\DX 4  
\mac_itw.dec02\set01\X97M\LAROUX\A 5  
\mac_itw.dec02\set01\X97M\JINI\A 7  
\mac_itw.dec02\set01\X97M\DIVI\D 4  
\mac_itw.dec02\set01\X97M\DIVI\A 9  
\mac_itw.dec02\set01\X97M\BARISADA\B 4  
\mac_itw.dec02\set01\O97M\TRISTATE\C_XLS 8  
\mac_itw.dec02\set01\O97M\TRISTATE\C_PPT 5  
\mac_itw.dec02\set01\O97M\TRISTATE\C_DOC 9
```

E.3. Wildlist December 2002

Remark: The appended list is maintained by Wildlist Organisation. The aVTC uses this list, with permission of this organisation, to select broadly available viruses for "In-the-Wild" (file, boot and macro) virus detection testing. We wish to thank Wildlist Org and esp. Joe Wells and Ian Whalley for their support.

=====

PC Viruses In-the-Wild - December, 2002

=====

This is a cooperative listing of viruses reported as being in the wild by 73 virus information professionals. The basis for these reports are virus incidents where a sample was received, and positively identified by the participant. Rumors and unverified reports have been excluded.

Some programs included in this list may fall outside the traditional definition of a computer virus. However, such programs are spreading throughout diverse user populations, are a threat to users and are therefore included in this list.

This report is cumulative. That is, this is not just a report of which viruses were seen last month. Monthly data is received from most participants, but the new data is added to the old. Participants are expected to let us know when to remove their name from a virus. The list should not be considered a list of "the most common viruses", however, since no specific provision is made for a commonness factor.

This data indicates only "which" viruses are In-the-Wild, but viruses reported by many (or most) participants are obviously widespread. The WildList is currently being used as the basis for in-the-wild virus testing and certification of anti-virus products by the ICSA, Virus Bulletin and Secure Computing. Additionally, a virus collection based upon The WildList is being used in an effort to standardize the naming of common viruses.

The WildList - (c)1993-2002 by Joe Wells - <http://www.wildlist.org>

=====

Key	Participant	Region	Organization	Product
Ac	Alan Candy	New Zealand	Applied Insight	-
Ad	Allan Dyer	Hong Kong	Yui Kee Co. Ltd.	F-Prot
Ae	Amir Elbaz	Israel	eAladdin	eSafe Protect
Ak	Ahmad Y. Kashoor	Syria	CompuKashoor	Dr Solomon's
Al	Andrew J. Lee	UK	Team Anti-Virus	-
Am	Andreas Marx	Germany	Univ. of Magdeburg	-
Ao	Andy Cianciotto	USA	Symantec	-
As	Alex Shipp	UK	MessageLabs	StarScan
Ay	Allysa Myers	USA	McAfee (US)	VirusScan
Bd	Bogdan Dumitru	Romania	Softwin SRL	-
Cb	Carl Bretteville	Norway	Norman ASA	NVC
Cr	Costin RAIU	Romania	Kaspersky Labs	KAV
Cs	Christian Schmid	Austria	DataPROT Linz	F-Prot
Dp	David Phillips	UK	Open University	-
Dr	David Rotenberg	Brazil	Maple Informatica	-
Ei	Eddy Willems	EU	EICAR	-
Ek	Eugene Kaspersky	Russia	Kaspersky Labs	KAV
Ew	Eddy Willems	Belgium/Lux.	Data Alert Int'l	VirusScan
Fp	Francois Paget	France	McAfee (France)	VirusScan
Gb	Gerald Batten	Canada	Independent	-
Gr	Greg Romania	USA	ICSA Labs	-
Jd	Joost de Raeymaeker	Portugal	RSVP	-
Jh	Joe Hartmann	USA	Trend Micro	PC-cillin
Jj	Jong Purisima	-	Trend Micro	PC-cillin
Jk	Jimmy Kuo	USA	McAfee (Independent)	-
Jm	Jose Martinez	Peru	HackSoft S.R.Ltda	TH AV
Jp	Josef Pichlmayr	Austria	Ikarus Software	-
Jy	Jamz Yaneza	-	Trend Micro	PC-cillin
Kb	Kenneth Bechtel	USA	Team Anti-Virus	-
Kd	K. T. Davies	India	Pioneer Micro	Vaxine
Ls	Luca Sambucci	Italy	Min. Comunicazioni	-

Ma	Matthew Ham	UK	Virus Bulletin	-
Mh	Mikko Hypponen	Finland	F-Secure Corp.	F-Secure
Mo	Martin Overton	UK	Independent	-
Ms	Marek Sell	Poland	Marek Sell, Ltd.	MkS_vir
Mt	Miroslav Trnka	Slovakia	ESET Ltd	NOD-ICE
Mx	Michael Xie	Canada/USA	Fortinet, Inc.	FortiGate
Oz	Jakub Kaminski	Australia	Computer Associates	VET
Pb	Pavel Baudis	Czech Republic	ALWIL Software	Avast!
Ph	Per Hellqvist	Sweden	Symantec	NAV
Pm	Paolo Monti	Italy	Future Time S.r.l	NOD32
Pr	Peter Radatti	USA	CyberSoft, Inc.	Vfind
Pt	Peter Theobald	India	IT Secure Software	VirusScan
Ra	Ruben Arias	Argentina	RALP	Integ Master
Rf	Richard Foley	Ireland	Reflex Magnetics	TBAV
Rg	Ray Glath, Sr.	USA	Tavisco Ltd.	Vi-Spy
Rp	Ronnie Pineda	Philippines	Mannasoft Corp.	VirusScan
Rv	Robert Vibert	Canada	Independent	-
Rz	Righard Zwienenberg	Netherlands	Norman Data Defense	NVC
Sa	Siggi Stefnisson	Iceland	FRISK Software	F-Prot
Sh	Sha-Li Hsieh	USA	Computer Associates	Inoculate IT
Sj	Sanjay Katkar	India	Cat Computer Services	Quick Heal
Sk	Seok-Chul Kwon	Korea	HAURI	ViRobot
Sm	Seiji Murakami	Japan	JCSR	-
So	SiHaeng Cho	South Korea	Ahnlab, Inc.	V3.
Sr	Subramanya Rao	India	Proland Software	Protector Plus
Ss	Szilard Stange	Hungary	VirusBuster Ltd.	Virus Buster
St	Stuart Taylor	UK	Sophos Plc.	Sweep
Ta	Tjark Auerbach	Germany	H+BEDV GmbH	AntiVir
Tc	Tzvetan Chaliavski	USA	Command Software	F-Prot Pro
Td	Toralv Dirro	Germany	U of Hamburg	VirusScan
Ti	Torben Immisch	Denmark	Swanholm Distrib.	NAV
Tm	Taras Malivanchuk	Israel	iRiS Software	AntiVirus Plus
Wl	WLO	-	WLO	-
Ws	Wolfgang Stiller	USA	Stiller Research	Integ Master
Xc	Xabier Cazalis	Spain	Panda Software	Panda
Za	Daryl Pecelj	Global	Microsoft Corporation	-
Zb	James Wolfe	USA	Independent	-
Zv	Corporate Group	-	-	-
Zw	Corporate Group	-	-	-
Zx	Corporate Group	-	-	-
Zy	Corporate Group	-	-	-
Zz	Corporate Group	-	-	-

=====
The WildList
=====

This main list includes viruses reported by multiple participants, which appear to be non-regional in nature. Technically, this first list is "the" WildList according to original specification, which required viruses to be verified In-the-Wild by a minimum of two participants. A supplemental list follows that contains viruses reported by single participants. After falling off, viruses sometimes reappear on The WildList. Such viruses are denoted with the symbol "*".

+ Viruses marked with a plus sign (+) are new to the main list this month.

Name of Virus	[Alias(es)	List Reported	Date	by:
AntiCMOS.A.....	[Lenart.....]	1/95	OzSjSkSmSoWsZz	
AntiEXE.A.....	[D3, New Bug, Ne]	9/94	FpSoWsZz	
Bleah.....	[Bleah.D, Bleah.]	2/02	AlFpSkSo	
Dodgy.....	[.....]	4/02	SgSkSo	
Empire.Monkey.B.....	[Monkey 2.....]	6/94	SoWsZz	
Form.A.....	[Form 18.....]	7/94	SmWsZz	
JS/Kak.A-m.....	[.....]	2/00	AlAoAsAyCrDpEiEwFpJdKbKdMo MtOzPbSjSmSrStZbZvZz	
Junkie.mp.1027.A.....	[DrWhite.1027...]	7/94	OzWs	
NYB.A.....	[Bl.....]	7/94	SjSoSrWsZz	
O97M/Tristate.C.....	[Crown.B.....]	4/99	AsCrFpKbKdLsMtSmSoStZbZvZz	
One_Half.mp.3544.A.....	[Dis, Free Love.]	10/95	MtPbSo	

Risk Analysis of Mobile Devices with Special Concern of Malware Contamination – Appendix E

Ripper.....	[Jack Ripper....]	4/02	EwSkSo
VBS/Freelink-mm.....	[.....]	10/99	AoJdPhSmSt
VBS/Haptime.A-mm.....	[Help.....]	6/01	AoAsAyFpJmJyKdMsMtOzPhSaSj SkSmSoStTmXcZbZvZyZz
VBS/Haptime.D-mm.....	[.....]	6/01	MtSo
VBS/LoveLetter.A-mm.....	[BugFix, I-Worm.]	5/00	AoAsAyEiEkEwFpJdJmMsRvSaSm SoStTmZbZvZz
VBS/LoveLetter.AS-mm.....	[Plan.A.....]	10/00	AoAsAyDpFpMsOzPhRvSkSoStTm ZvZyZz
VBS/LoveLetter.C-mm.....	[.....]	10/00	AoAsAySkSoSt
VBS/Netlog.A.....	[Network.....]	3/00	AoOzSmSo
VBS/Redlof.A-m.....	[.....]	10/02	AoDpGrJmMsMtPhSgSjSkSmStTa TcZbZvZz
VBS/SSIWG2.A-mm.....	[Daira.A-mm.....]	5/02	TaTm
VBS/Stages.A-mm.....	[ShellScrap.....]	7/00	AoAsFpJdOzZz
VBS/Tam.A-m.....	[.....]	2/01	AoAsFpZbZvZz
VBS/VBSWG.AQ-mm.....	[.....]	7/02	AsSoSt
VBS/VBSWG.J-mm.....	[Anna K, Kalamar]	2/01	AmAoAsAyEiEwSoStZvZz
VBS/VBSWG.K-mm.....	[NeueTarife SST,]	2/01	EiEw
VBS/VBSWG.X-mm.....	[HomePage, SST..]	5/01	AmAoAsDpEiEwFpJdStZvZyZz
VBS/VBSWG.Z-mm.....	[Mawanella, Mawn]	6/01	AsSmZb
W32/Acebot.....	[Newbiero.....]	6/02	AoMoSgTm
W32/Aliz.A-mm.....	[.....]	11/01	AlAmAoAsAyBdEkFpJjKdLsMsMt OzPbSaSmSoSsStTaZvZyZz
W32/Anset.A-mm.....	[.....]	11/01	AsKdTaTi
W32/Anset.B-mm.....	[.....]	12/01	AmTa
W32/Anset.C-mm.....	[.....]	12/01	AmSsTa
W32/Aplore.A-mm.....	[Aphex.....]	6/02	AoAsZbZvZyZz
W32/Apost.A-mm.....	[.....]	10/01	AoAsStTaZv
W32/BadTrans.A-mm.....	[13312.....]	5/01	AlAoAsFpGrJdKdLsOzPhSoSrSt TmZbZvZz
W32/BadTrans.B-mm.....	[29020.....]	11/01	AlAmAoAsBdCrDpDrEiEkEwFpJd JjJkJmJyKbKdLsMhMoMsMtOzPb PhRzSaShSjSkSmSoSrSsStTaTc TmXcZbZvZy
W32/Benjamin.A-mm.....	[.....]	6/02	AoMsSoStTaTm
W32/BleBla.B-mm.....	[Verona.B.....]	12/00	AoAsAyFpJmKdSjSoZbZz
W32/Bady.C.....	[Code Red II....]	8/01	AoSoZv
W32/Braid.A-mm.....	[.....]	11/02	AlAsAyFpGrJkJmJpMhMsMtPhSg SjSkSmStTcTmZbZvZz
W32/BugBear-mm.....	[.....]	10/02	AlAmAoAsAyDpEiEwFpGrJkJmJp MhMoMsMtOzPbPhRvSaSgShSjSk SmSrSsStTaTcTmXcZbZvZyZz
W32/Cervivec.A.....	[.....]	4/02	AoAsMtPbStTaZbZvZz
W32/Chir.A-mm.....	[.....]	10/02	SjStTa
W32/Choke.A.....	[.....]	7/01	AoAyEiOzSt
W32/Datom.A.....	[.....]	9/02	AmAoAyFpMsMtOzSjSmSoSrStTc Xc
W32/Duni.A.....	[W32/Duni.B, W32]	8/02	AyFpMoOzSaSmStXc
W32/Elkern.A.....	[WQK.A.....]	12/01	AoAsEkFpJjKdOzShSkSoStTaTm
W32/Elkern.B.....	[WQK.B.....]	2/02	EkFpJmKbKdSaSkSoStZv
W32/Elkern.C.....	[WQK.C.....]	5/02	AlAoAyEkFpJdJmKbKdLsMsMtOz PhRvSaShSjSkSoStTaTcTmXcZb ZvZz
W32/ExploreZip-m.....	[Worm.ExploreZip]	6/99	RvZb
W32/FBound.C-mm.....	[.....]	4/02	AmAoAyKdMhMsMtOzShSmSoSrSs StTaXcZvZyZz
W32/Fix2001-m.....	[Admin.12288, Fi]	2/00	AoAsSmZv
W32/Frethem.F-mm.....	[.....]	7/02	AyEkOzSmSoSrStTaTmZz
W32/Frethem.L-mm.....	[.....]	9/02	AlAmAoFpPhPmSkSsStXcZy
W32/Funlove.4099.....	[.....]	12/99	AlAmAoAsAyEiEwFpGrJmJyKdMo MsMtOzPhRvSaSgSjSkSmSoSrSt ZbZvZyZz
W32/Gibe.A-mm.....	[.....]	3/02	AlAmAoAsAyDpFpJdMhMoMsOzSg ShSjSmSoSrSsStTaZbZvZy
W32/Gokar.A-mm.....	[.....]	1/02	AoAsFpJyMoOzPhStTmZvZz
W32/Goner.A-mm.....	[.....]	12/01	AlAmAoAsAyBdCrEiEkFpJdJjJk JmJyKbKdLsMsMtOzRzSaShSmSo SrSsStTmXcZbZvZyZz
W32/Gop.A-mm.....	[Invery.A.....]	2/02	DpPbShSkSoStZaZvZz

W32/Hai.A..... [.....]	10/01	AoAyFpOzSgSkSoXc
W32/HiGuy.A-mm..... [Tettona.....]	7/02	AoAsAyDpFpPhSgSmSoSrStZyZz
W32/Hybris.A-mm..... [Hybris.22528-mm]	1/01	MsSk
W32/Hybris.B-mm..... [Hybris.23040-mm]	12/00	AlAmAoAsAyDpDrEiEwFpGrJdJm JpKbLsMoMtOzPbPhSaShSkSmSo SrSsStTaTcTiTmZbZvZyZz
W32/Hybris.C-mm..... [.....]	12/00	AsDpJpPhStTaZvZy
W32/Hybris.D-mm..... [Hybris.25088-mm]	1/01	AlAmEiEkFpJmJpPhSaSjSkSmSo StTaTcZvZy
W32/Klez.A-mm..... [.....]	12/01	AoAsCrKdSkSoStTm
W32/Klez.B-mm..... [.....]	11/01	AlAsAyJjSkSt
W32/Klez.C-mm..... [.....]	3/02	JjSk
W32/Klez.D-mm..... [.....]	12/01	FpOzShSoSsStTm
W32/Klez.E-mm..... [.....]	2/02	AlAmAoAyDpEiEkEwFpJdJmJpKb KdLsMhMoMsMtOzPbPhRvSaSgSh SkSoSrSsStTaTcTmXcZbZvZyZz
W32/Klez.G-mm..... [.....]	3/02	DpEiSkSoStZv
W32/Klez.H-mm..... [.....]	5/02	AlAmAoAsAyDpEiEkEwFpGrJdJk JmJpKbKdLsMoMsMtOzPbPhPmRv RzSaSgShSjSmSoSrSsStTaTcTm WlXcZbZvZyZz
W32/Kriz.4050..... [.....]	11/00	AoAsJmKdMsOzSjSkSo
W32/Magistr.A-mm..... [Disembowler.....]	4/01	AlAmAoAsAyCrDpDrEiEkEwFpJd JkJmJyKdLsMoMsOzPbPhRvSaSg ShSkSmSoSrSsStTaTcTiTmWlXc ZbZvZyZz
W32/Magistr.B-mm..... [.....]	10/01	AlAmAyDpDrEiEkEwFpGrJdJmJy KbKdMhMoMsMtOzPhSaSgShSjSk SoSrSsStTaTcTiTmWlXcZbZvZy Zz
W32/Maldal.C-mm..... [Keyluc, Reezak.]	3/02	AsFpShSt
W32/Maldal.E-mm..... [Keyluc, Reezak.]	1/02	OzSaSrTa
W32/Maldal.F-mm..... [Keyluc, Reezak.]	3/02	AmAsStTa
W32/MSInit.B..... [Bymer.B.....]	11/00	MtOzSmSo
W32/MTX-m..... [Apology, Matrix]	9/00	AoAsAyCrDpEiEwFpJdJyKbKdMo MtOzPhSaSgSjSmSoSrStTmZvZy Zz
W32/MyLife.A-mm..... [30720.....]	4/02	AmAsAySrStZbZyZz
W32/MyLife.B-mm..... [7680.....]	4/02	AoAsAyMhMtOzShSoSsStZbZyZz
W32/MyLife.F-mm..... [11524.....]	4/02	AmAsAySoStZv
W32/MyLife.G-mm..... [.....]	6/02	StTa
W32/MyLife.J-mm..... [.....]	5/02	AoShSmSrTaZvZz
W32/MyParty.A-mm..... [.....]	2/02	AmAoCrFpJdMoMsMtOzSaShSkSm SoSrStTmXcZbZvZz
W32/Naked.A-mm..... [NakedWife.....]	3/01	AsSo
W32/Navidad.A-m..... [Navidad.....]	11/00	AoAsEiEwOzSmSoZvZz
W32/Navidad.B-m..... [Emanuel.....]	1/01	EiEwSkSoSsZvZz
W32/Nimda.A-mm..... [.....]	10/01	AlAmAoAsAyCrDpDrEiFpGrJdJm JpKbKdMhMoMsMtOzPbPhRvSaSh SkSmSoSrSsStTaTiTmXcZaZbZv ZyZz
W32/Nimda.E-mm..... [.....]	11/01	AlAmAoAsAyCrEiEkFpJmJyKdMh MoMtOzPbPhSaSgShSjSkSoStTa TiTmZaZvZy
W32/Onamu.A-mm..... [.....]	5/02	JmPhSoStZyZz
W32/Opaserv.A..... [.....]	10/02	AyFpMhMoMsPhSjSmStTcXc
W32/Opaserv.B..... [.....]	10/02	AyMoMsPhSjSrSsSt
W32/Opaserv.D..... [.....]	10/02	AyMoPhSjSrSs
W32/Opaserv.E..... [.....]	11/02	AyMhStXc
W32/Opaserv.G..... [.....]	11/02	AySrTa
+W32/Opaserv.H..... [.....]	12/02	MoTa
+W32/Oror.B-mm..... [.....]	12/02	GrTaTm
+W32/Parite.A..... [.....]	12/02	PmTa
W32/PrettyPark.37376-mm..... [.....]	6/99	AsFpZvZz
W32/Prolin.A-mm..... [Creative.A.....]	12/00	AoAsZv
W32/Qaz..... [.....]	9/00	AmAoEwFpJmKbMtOzRvSmSoZbZv Zz
W32/SirCam.A-mm..... [.....]	7/01	AlAmAoAsAyCrDpDrEiEwFpGrJd JmJyKbKdLsMhMoMsMtOzPbPhRv SaSgShSjSkSmSoSrSsStTaTcTi

			TmXcZbZvZyZz
W32/Ska.A-m.....	[HAPPY99.....]	3/99	AmAoAsCrFpOzPhSmSrZbZvZz
W32/Stator.A.....	[.....]	9/02	AoMsMtSt
W32/Surnova.A.....	[Supova.....]	9/02	MoMsSkStTaTm
W32/Surnova.D.....	[Supova.....]	9/02	AlMoSt
+W32/Winevar.A-mm.....	[.....]	12/02	AsAyGrJpMsSjSkSmSrStZv
W32/Yaha.A-mm.....	[Lentin.A, Yaha.]	4/02	AoAsFpJdKdMhStTaZvZyZz
+W32/Yaha.C-mm.....	[.....]	12/02	AsTa
W32/Yaha.D-mm.....	[Lentin.B.....]	5/02	AoAsMsStTaZy
W32/Yaha.E-mm.....	[Lentin.D.....]	7/02	AlAmAoAsAyDpFpMoPbPhShSsSt TaZbZy
W32/Yaha.G-mm.....	[Lentin.F.....]	7/02	AlAmAsAyFpGrJpMoMtOzPhPmRz SaSkSmSoSrStTaTcTmZbZz
W32/Zoek.D-mm.....	[.....]	9/02	AsStZy
W32/Zoher.A-mm.....	[Sheer.A, Sheer.]	1/02	AlAmAoDrJyOzShSmSoSrZv
W95/CIH.1003.....	[CIH.A, Spacefil]	8/98	AmAsAyFpGrJdKdMsMtOzPhSaSk SmSoSrTmZbZvZz
W95/CIH.1019.A.....	[1019, CIH.C....]	7/98	AlAoSjSk
+W95/Dupator.1503.....	[.....]	12/02	AyJmMoMsPmSj
W95/Lovesong.998.....	[.....]	7/02	SkSo
W95/Plage-m.....	[HLLW, W32/Plage]	7/00	AsKd
W95/Spaces.1445.....	[Busm.1445.....]	12/00	AmAoAsAyCrFpJmKdMoMsPhSaSj SkSoSrSsStTmXcZbZvZyZz
W95/Weird.10240.....	[Kuang.GR.....]	7/00	AmAoAsKdPhSkSoZv
W97M/Assilem.A.....	[.....]	7/02	AlAs
W97M/Bablas.A.....	[.....]	1/00	AsSmZbZvZz
W97M/Bleck.A.....	[.....]	3/02	AsTm
W97M/Bottra.A.....	[.....]	7/01	AsZv
+W97M/Bottra.C.....	[.....]	12/02	AsSt
W97M/Chack.B.....	[.....]	3/00	AsDpZb
W97M/Class.B.....	[.....]	2/99	AsKbZbZv
W97M/Class.D.....	[.....]	12/98	AsEiEwFpKdSmZbZvZz
W97M/Class.Q.....	[.....]	12/98	AsZv
W97M/ColdApe.A.....	[.....]	12/98	AoAsFpZb
W97M/ColdApe.B.....	[.....]	3/99	AsZvZz
W97M/Eight941.D.....	[Eight.....]	5/00	AsSmZbZv
W97M/Eight941.E.....	[.....]	3/00	AoAs
W97M/Ethan.A.....	[.....]	2/99	AlAsEwFpJyPhSjSmStZbZvZyZz
W97M/Ethan.AK.....	[.....]	12/01	AlAsSt
W97M/Ethan.BE.....	[.....]	3/00	AoAsZv
W97M/Ethan.Q.....	[.....]	9/99	AoAsZbZv
W97M/Flop.A.....	[.....]	8/01	AsZv
W97M/Groov.A.....	[.....]	7/98	AoAsKdPhZbZvZz
W97M/Groov.B.....	[.....]	10/99	AoAsZbZv
W97M/Locale.A.....	[.....]	9/99	AsFpZbZv
W97M/Marker.BN.....	[HSFX, Spooky...]	5/00	AsZbZz
W97M/Marker.C.....	[Spooky.C.....]	4/99	AlAoAsDpEiEwFpJdKbSjSoStZb ZvZz
W97M/Marker.D.....	[.....]	5/99	AsKdSm
W97M/Marker.O.....	[.....]	8/99	AoAsTaZbZvZz
W97M/Marker.Q.....	[.....]	9/99	AsJdZz
W97M/Melissa.A-mm.....	[Maillissa.....]	4/99	AoAsFpPhSmZbZvZz
W97M/Melissa.O-mm.....	[.....]	2/00	AsZb
W97M/Myna.B.....	[.....]	1/00	AsSmZbZvZz
W97M/Myna.C.....	[.....]	3/00	AsStZbZv
W97M/Nagem.A.....	[.....]	1/01	AsKd
W97M/Nono.A.....	[.....]	2/99	AsZv
W97M/Onex.E.....	[.....]	12/00	AsDp
W97M/Opey.A.....	[.....]	3/00	AsSmZbZv
W97M/Opey.M.....	[.....]	10/00	AsZvZz
W97M/Ostrich.B.....	[.....]	7/01	AoAsRz
W97M/Panther.A.....	[.....]	11/99	AoAsZvZz
W97M/Pri.A.....	[.....]	6/99	AsZbZvZz
W97M/Pri.Q-mm.....	[Prilissa.A.....]	11/99	AsZvZz
W97M/Proverb.A.....	[.....]	5/00	AsCrDpMtZbZv
W97M/Service.A-mm.....	[.....]	6/00	AsFp
W97M/Smac.D.....	[BDOC2X, SMAC-D.]	5/00	AoAsFpZvZz
W97M/Story.A.....	[Jack_Box.....]	8/99	AsFpPhZbZvZz
W97M/Surround.A.....	[.....]	7/00	AoAs
W97M/Thus.A.....	[Thursday.A.....]	9/99	AlAoAsCrJdPhSmStZbZvZyZz

W97M/Thus.C.....	[Thursday.....]	6/00	AsZbZv
W97M/Thus.I.....	[.....]	8/00	AsPhSt
W97M/Thus.X.....	[W97M/Bethlem...]	6/00	AsFpZv
W97M/Titch.D.....	[.....]	9/00	AoAsDpFpPhZbZvZz
W97M/Titch.H.....	[.....]	3/01	AsFp
W97M/Title.A.....	[.....]	12/00	FpZv
W97M/Verlor.A.....	[.....]	1/00	AlAsFpZbZvZyZz
W97M/VMPCk1.BY.....	[.....]	8/99	AsFpJdSt
W97M/VMPCk1.DD.....	[.....]	2/01	AsFpZvZz
W97M/Walker.E.....	[.....]	11/01	AsZz
W97M/Wrench.E.....	[.....]	7/00	AoAsFpZbZv
WM/CAP.A.....	[.....]	5/97	AlAoAsFpJyPhSjSmZbZvZz
WM/Concept.A.....	[Prank Macro....]	12/96	AsSmStWsZbZvZz
WM/CopyCap.A.....	[WM.Pac.A.....]	7/98	AsZbZz
WM/Npad.A.....	[Jakarta.....]	10/96	AsFpZvZz
WM/Wazzu.A.....	[Wazzu.....]	12/96	SmWsZbZvZz
Wyx.....	[Preboot.....]	11/01	FpKdOzSjSkSoZz
X97M/Barisada.B.....	[X97M/HJB.B.....]	9/00	AsSo
X97M/Divi.A.....	[BASE5874, DIVI.]	12/99	AsSkSo
X97M/Divi.D.....	[.....]	7/00	AsSoZb
X97M/Jini.Al.....	[.....]	11/00	AsJd
X97M/Laroux.A.....	[.....]	7/97	AsMtPbSkSoTaZbZvZz
X97M/Laroux.DX.....	[.....]	12/98	AsJdKdStZbZvZz
X97M/Laroux.E.....	[.....]	12/98	AsZbZvZz
X97M/Laroux.HO.....	[.....]	3/02	AsZv
X97M/Yawn.A.....	[.....]	7/00	AsKdSoStZv
XM/Laroux.A.....	[.....]	2/97	PbSkSmSoZz

=====
 Total for top list: 196
 =====

Supplemental List

=====
 As was noted at the start of the main list, this list is not technically part of "The WildList", as originally defined. By design, The WildList is a list of viruses verified as being In-the-Wild by a minimum of two WildList participants. The viruses listed below do not currently meet that criteria.

This additional list includes viruses reported by a single participant and are often either moving onto the main list, or dropping off of it. Please note especially that this list also tends to be more of a regional reporting mechanism. For example, a virus is often reported as very common by one regional participant, but is found nowhere else in the world.

Viruses marked with a minus sign (-) dropped from the main list this month. Viruses marked with a plus sign (+) are new to the supplemental list this month.

After falling off, viruses sometimes reappear on the supplemental section of The WildList. Such viruses are denoted with the symbol "*".

Name of Virus	[Alias(es)]	List Date	Reported by:
Bap.mp.1536.....	[Baphometh.....]	5/02	So
BAT/BWG.B.....	[.....]	10/02	Ta
BAT/Christina.A.....	[.....]	2/02	Tm
+Bat/Eversaw-mm.....	[.....]	12/02	Ta
-BAT/HitOut.A-mm.....	[.....]	6/02	Rz
BAT/Savec.....	[.....]	10/02	Ta
Bolivian.....	[.....]	12/01	Tm
Boot-437.A.....	[Bath.....]	4/02	Ws
DEFO.....	[PeterII.Runtime]	4/02	So
Delwin.....	[.....]	4/02	So
Hare.mp.7786.A.....	[.....]	5/02	So
HLL0.Nmkamil.8383.A.....	[.....]	10/02	Ta
Jerusalem.1808.Standard.	[1808, 1813, Isr]	4/02	Ws
JS/Kak.AB-m.....	[.....]	11/01	As
JS/Kak.AE-m.....	[.....]	3/02	Al
JS/Kak.D-m.....	[.....]	1/02	Al
JS/Messenger.....	[.....]	2/02	Ao
JS/YAMA.B-m.....	[.....]	5/01	Jm

Michelangelo.A.....[.....]	7/93	Ws
Natas.mp.4744.....[.....]	5/02	So
Neuroquila.mp.4544.A...[Havoc, Wedding.]	6/02	Ao
NRLG.700.A.....[.....]	6/02	Ao
O97M/Cybernet.A-mm.....[.....]	2/01	As
O97M/Jerk.B.....[AllNet.....]	7/02	As
O97M/Shiver.C.....[.....]	5/02	So
O97M/Tristate.I.....[.....]	7/00	As
Parity_Boot.B.....[Generic 1.....]	9/93	Ew
Pieck.mp.4444.A.....[Kaczor.mp.GR...]	9/02	Mt
Sampo.....[Turbo, Wllop...]	1/95	Zz
StealthBoot.C.....[.....]	11/93	Zz
Stoned.No_INT.A.....[Stoned.....]	9/94	Ws
Stoned.Spirit.....[.....]	6/96	Fp
Stoned.Standard.B.....[New Zealand....]	3/93	Ws
Stoned.W-Boot.....[Stoned.P, Wonka]	12/93	Ws
TMC_Level-42.....[.....]	2/99	Mt
Tremor.4000.A.....[.....]	7/93	Ws
VBS/Ardin.....[Jadra.B, Madonn]	5/02	Ta
VBS/BWG.D-mm.....[.....]	8/02	Ta
VBS/BWG.E-mm.....[.....]	8/02	Ta
VBS/BWG.F-mm.....[.....]	8/02	Ta
VBS/Chick.A-m.....[Brit.A.....]	5/02	Ta
VBS/Chick.B-m.....[Brit.B.....]	5/02	Ta
VBS/Chick.C-m.....[Brit.C.....]	7/02	Ta
VBS/Chick.D-m.....[.....]	7/02	Ta
VBS/Chick.F-m.....[Brit.F.....]	8/02	Ta
VBS/Chu.A-mm.....[.....]	7/02	Ta
VBS/Doublet.A-mm.....[.....]	5/02	Ta
+VBS/Fasan.....[.....]	12/02	Ta
+VBS/Gorum.....[.....]	12/02	Ta
VBS/Haptime.B-mm.....[.....]	7/01	Tm
VBS/Krazyb.A-mm.....[.....]	10/02	Ta
VBS/LoveLetter.BG-mm....[.....]	12/00	As
VBS/LoveLetter.BJ-mm....[.....]	7/01	Ao
VBS/LoveLetter.CI.....[.....]	6/02	Ao
VBS/Lubus.A-mm.....[.....]	1/02	Jm
+VBS/Mill.....[.....]	12/02	Ta
+VBS/Quocus.....[.....]	12/02	Ta
VBS/Shakira.A.....[.....]	8/02	Ta
VBS/Sorry.C.....[Mcon.B, Pica...]	3/01	Ao
VBS/SSIWG.U-mm.....[.....]	10/01	Am
+VBS/Sucop.A.....[.....]	12/02	Ta
W32/Agobot.....[.....]	11/02	Ay
W32/Alcaul.N-mm.....[.....]	5/02	Ta
W32/Alcaul.O-mm.....[.....]	5/02	Ta
W32/Alcaul.S-mm.....[.....]	8/02	Ta
+W32/Apart.A.....[.....]	12/02	St
W32/Appix.A.....[.....]	11/02	St
+W32/Appix.C-mm.....[.....]	12/02	Ta
+W32/Appix.E-mm.....[.....]	12/02	St
W32/Beast.A.....[W95/HLLP.41472.]	1/02	Ao
W32/Bezilom.A.....[.....]	3/02	St
W32/BleBla.A-mm.....[Verona.A.....]	12/00	Oz
W32/Blinkom.A.....[.....]	9/02	Ay
W32/Carrytone.A-mm.....[Taripox.B.....]	3/02	St
-W32/CTX.....[6886, 6886, 720]	2/02	St
W32/Disque.A.....[.....]	2/02	Tm
W32/DoTor.A-mm.....[{W32,W97M,VBS}/]	8/02	Ta
W32/Duload.B.....[.....]	10/02	Ta
W32/Duni.C-mm.....[.....]	7/02	Ay
+W32/Energy.F-mm.....[.....]	12/02	Ta
W32/Enerkaz.A.....[.....]	10/02	Ta
W32/Enerkaz.B.....[.....]	10/02	Ta
W32/Enerkaz.E.....[.....]	10/02	Ta
W32/Estrella.A.....[.....]	1/02	Jm
W32/FBound.A-mm.....[.....]	5/02	Sk
W32/FBound.B-mm.....[.....]	5/02	Sk
+W32/Fleming.....[.....]	12/02	Ta
W32/Frethem.K-mm.....[.....]	9/02	Al

W32/Frethem.M-mm.....	[.....]	8/02	St
+W32/Gismor-mm.....	[.....]	12/02	Ta
W32/Gnuman.....	[.....]	4/01	Ao
W32/Heidi.A-mm.....	[Critical.A.....]	2/02	Tm
W32/HLLP.46808.A.....	[.....]	2/02	Tm
W32/HLLP.DeTroie.A.....	[.....]	6/02	Ao
W32/Holar.C-mm.....	[.....]	12/02	As
+W32/Indor-mm.....	[.....]	12/02	Ta
W32/Kitro.D.....	[.....]	8/02	St
W32/Klez.F-mm.....	[.....]	4/02	Sk
W32/Kriz.4029.....	[.....]	1/00	Sm
W32/Lastscene.B.....	[.....]	2/02	Jh
W32/Libid.A.....	[.....]	10/01	Jm
W32/Maldal.A-mm.....	[.....]	3/02	Ta
W32/Maldal.K-mm.....	[.....]	6/02	Ta
W32/Mario.45056.A.....	[.....]	7/01	Jm
W32/Mars.A-mm.....	[Gubed.A.....]	8/02	Ta
+W32/Merkur-mm.....	[.....]	12/02	Ta
W32/Moocow.A.....	[Kwbot.....]	7/02	St
W32/MSInit.A.....	[Bymer.A.....]	11/00	Ao
W32/MyLife.D-mm.....	[.....]	6/02	Ta
-W32/MyLife.E-mm.....	[.....]	4/02	Ay
-W32/MyLife.H-mm.....	[.....]	5/02	St
W32/MyLife.I-mm.....	[.....]	6/02	Ta
W32/Mypics-m.....	[.....]	1/02	Ao
W32/Nahata.....	[.....]	7/02	Ta
W32/Nautical.....	[.....]	10/02	Ta
W32/Navidad.E-m.....	[.....]	10/02	As
W32/Nimda.B-mm.....	[.....]	12/01	Ta
W32/Nimda.F-mm.....	[.....]	12/01	Ta
W32/Pam.36352.A.....	[.....]	7/01	Jm
W32/Pepex.A.....	[.....]	10/02	Am
W32/Plexis.A-mm.....	[.....]	5/02	Ta
W32/PrettyPark.51433-mm.....	[.....]	5/00	Ao
W32/PrettyPark.60928-mm.....	[.....]	3/01	As
W32/Q4Like.A-mm.....	[.....]	12/01	Sh
+W32/Ramdile.....	[.....]	12/02	Ta
+W32/Razor-mm.....	[.....]	12/02	Ta
W32/Redesi.C-mm.....	[.....]	12/01	Am
W32/Redesi.D-mm.....	[.....]	12/01	St
W32/SelfCloner.....	[.....]	6/02	Tm
+W32/Sponge-mm.....	[.....]	12/02	Ta
W32/Surnova.B.....	[Supova.....]	8/02	St
W32/Surnova.E.....	[.....]	8/02	St
W32/Surnova.F.....	[.....]	10/02	Mo
W32/Sygate.A-mm.....	[.....]	2/02	Tm
W32/Trilisa.A-mm.....	[Orkiz.....]	7/02	Ta
W32/Ultimax.A.....	[.....]	8/02	St
W32/Vanessa.....	[.....]	6/02	Jm
W32/Vig.B.....	[.....]	10/02	Ta
-W32/Vote.A-mm.....	[.....]	11/01	Zv
W32/Welyah.L-mm.....	[.....]	2/02	St
+W32/Whog.878.....	[.....]	12/02	Mt
W32/Yarner.A-mm.....	[.....]	2/02	Am
W32/Yarner.B-mm.....	[.....]	2/02	Am
W95/Bumble.1736.....	[.....]	11/02	Mo
W95/Caw.1262.....	[.....]	2/01	Tm
W95/Fono.....	[El_Inca.17152..]	4/02	So
W95/Lorez.1766.....	[.....]	3/01	Jm
W95/Spaces.1633.....	[.....]	5/01	Oz
W95/Tecata.A.....	[.....]	9/02	Sm
W97M/Antimarc.A-mm.....	[.....]	11/99	As
W97M/Appder.A.....	[.....]	3/02	As
W97M/Assilem.C.....	[.....]	11/00	As
W97M/Astia.B.....	[.....]	3/01	As
W97M/Astia.C.....	[.....]	1/00	As
W97M/Astia.L.....	[.....]	3/00	As
W97M/Astia.U.....	[.....]	5/00	As
W97M/Bablas.BH.....	[.....]	4/01	As
W97M/Bablas.K.....	[.....]	3/00	As

W97M/Bablas.S.....	[.....]	12/00	As
W97M/Blowup.A.....	[.....]	12/01	St
W97M/BoBo.B.....	[.....]	7/02	As
W97M/Bottra.D.....	[.....]	12/01	Rz
W97M/Caligula.A.....	[.....]	11/00	As
W97M/Chack.CK.....	[.....]	4/02	Rz
W97M/Chack.G.....	[.....]	7/02	As
W97M/Chack.H.....	[.....]	7/99	As
W97M/Chack.K.....	[.....]	5/00	As
W97M/Class.AY.....	[.....]	5/01	As
W97M/Class.BV.....	[.....]	11/99	As
W97M/Class.DC.....	[.....]	4/02	St
W97M/Class.ED.....	[.....]	2/00	As
W97M/Class.EQ.....	[.....]	4/01	As
W97M/Claud.A.....	[.....]	3/00	As
W97M/Coke.22231.A.....	[.....]	12/00	As
W97M/Cont.A.....	[.....]	4/01	As
W97M/DB.A.....	[.....]	5/00	As
W97M/Ded.A.....	[.....]	10/99	As
W97M/Ded.B.....	[.....]	1/00	As
W97M/Ded.C.....	[.....]	7/02	As
W97M/Ded.R.....	[.....]	12/01	St
W97M/Ded.S.....	[.....]	12/01	St
W97M/Dest.E.....	[.....]	8/01	St
W97M/Dest.I.....	[.....]	6/02	St
+W97M/Dest.J.....	[.....]	12/02	As
W97M/Dig.C.....	[.....]	2/02	St
W97M/Doccopy.G.....	[.....]	7/02	St
+W97M/Doccopy.J.....	[.....]	12/02	St
W97M/Eight941.F.....	[.....]	12/00	As
W97M/Eight941.U.....	[.....]	3/02	As
W97M/Eight941.Y.....	[.....]	7/02	As
W97M/Ethan.AT.....	[.....]	10/99	As
W97M/Ethan.AW.....	[.....]	10/01	Zv
W97M/Ethan.BI.....	[Ethan.D.....]	3/00	As
W97M/Ethan.BR.....	[.....]	2/00	As
W97M/Ethan.BW.....	[.....]	2/00	As
W97M/Ethan.CC.....	[.....]	5/00	As
W97M/Ethan.CR.....	[.....]	11/00	As
W97M/Ethan.D.....	[.....]	6/00	As
W97M/Ethan.EN.....	[.....]	12/01	St
W97M/Ethan.EO.....	[.....]	5/02	St
W97M/Ethan.P.....	[.....]	2/00	As
W97M/FF.A.....	[Lys.H.....]	7/02	As
-W97M/FF.E.....	[.....]	11/00	As
W97M/Fifteen.A.....	[.....]	2/02	St
W97M/Footer.A.....	[.....]	12/99	As
W97M/Groov.AJ.....	[.....]	12/00	As
W97M/Groov.C.....	[.....]	11/99	As
W97M/Hope.A.....	[.....]	8/01	As
W97M/Hope.AG.....	[.....]	12/01	St
W97M/Hope.AH.....	[.....]	12/01	St
W97M/IIS.A.....	[.....]	11/00	As
W97M/IIS.E.....	[.....]	6/00	As
W97M/Inadd.D.....	[.....]	12/01	As
W97M/Ipid.G.....	[.....]	10/01	As
W97M/Jedi.P.....	[.....]	2/02	St
W97M/Jerk.A.....	[.....]	7/02	As
W97M/Jim.C-mm.....	[.....]	4/01	Ss
W97M/Kestrel.A.....	[.....]	2/02	St
W97M/Macroble.A.....	[.....]	7/02	As
W97M/Macroble.C.....	[.....]	5/01	As
W97M/Marker.AD.....	[.....]	12/00	As
W97M/Marker.AE.....	[.....]	9/99	As
W97M/Marker.AR.....	[.....]	10/99	As
W97M/Marker.AY.....	[.....]	2/01	As
W97M/Marker.AZ.....	[.....]	2/00	As
W97M/Marker.DG.....	[.....]	3/01	Zb
W97M/Marker.EL.....	[.....]	8/00	As
W97M/Marker.EO.....	[.....]	8/00	As

W97M/Marker.FI.....[.....]	4/01	As
W97M/Marker.FM.....[.....]	6/02	St
W97M/Marker.GO.....[.....]	10/01	As
W97M/Marker.J.....[.....]	3/00	As
W97M/Marker.JP.....[.....]	12/01	St
W97M/Marker.JT.....[.....]	12/01	St
W97M/Marker.JU.....[.....]	12/01	Rz
W97M/Marker.KM.....[.....]	3/02	St
W97M/Marker.KQ.....[.....]	5/02	St
W97M/Marker.N.....[.....]	2/00	As
W97M/Marker.T.....[.....]	3/01	As
W97M/Marker.X.....[.....]	9/99	As
W97M/MDMA.K.....[.....]	7/02	As
W97M/Melissa.I-mm.....[Empirical.....]	7/99	As
W97M/Melissa.M-mm.....[.....]	12/99	As
W97M/Metys.R.....[.....]	4/01	As
W97M/Model.A.....[.....]	2/00	As
W97M/Murke.A.....[.....]	10/00	As
W97M/Myna.AU.....[.....]	10/01	St
W97M/Myna.AY.....[.....]	12/01	St
W97M/Myna.AZ.....[.....]	12/01	St
W97M/Myna.BA.....[.....]	12/01	St
W97M/Myna.BB.....[.....]	12/01	St
W97M/Myna.G.....[.....]	4/01	As
W97M/Nagem.B.....[.....]	7/02	As
W97M/NewHope.C.....[.....]	2/01	As
W97M/Nid.A.....[.....]	3/00	As
W97M/NSI.D.....[.....]	7/02	As
W97M/Ocard.A.....[.....]	7/00	As
W97M/Odious.A.....[.....]	1/02	Ao
W97M/Onex.A.....[.....]	4/01	As
W97M/Onex.G.....[.....]	3/02	St
W97M/Onex.I.....[.....]	9/02	St
W97M/Opey.AF.....[.....]	12/00	As
W97M/Opey.AX.....[.....]	2/02	St
W97M/Opey.E.....[.....]	3/00	As
W97M/Ostrich.A.....[.....]	6/02	Am
W97M/Ozwer.F.....[.....]	1/02	Ao
W97M/Ozwer.Q.....[.....]	8/01	Rz
W97M/Ozwer.R.....[.....]	12/01	Rz
W97M/Panggil.C.....[.....]	3/02	St
W97M/Panther.U.....[.....]	2/02	St
W97M/Pecas.B.....[.....]	7/01	As
W97M/Piece.A-mm.....[.....]	3/01	As
W97M/Pri.B.....[.....]	3/99	As
W97M/Proverb.C.....[.....]	8/00	As
W97M/Replace.A.....[.....]	4/02	Zv
W97M/Replog.A.....[.....]	11/00	As
W97M/Seliuq.A.....[.....]	11/00	As
W97M/Shepmah.A.....[.....]	8/01	Cr
W97M/Sting.A.....[.....]	5/02	As
W97M/Surround.B.....[.....]	10/02	As
W97M/Temple.A.....[.....]	3/02	As
W97M/Thus.AP.....[.....]	12/00	As
W97M/Thus.AW.....[.....]	12/00	As
W97M/Thus.BH.....[.....]	4/01	As
W97M/Thus.BX.....[.....]	3/02	As
W97M/Thus.CQ.....[.....]	5/01	As
W97M/Thus.E.....[.....]	3/00	As
W97M/Thus.FR.....[.....]	4/02	St
W97M/Thus.H.....[.....]	3/00	As
W97M/Thus.K.....[.....]	7/00	As
W97M/Thus.M.....[.....]	7/00	As
W97M/Thus.O.....[.....]	11/00	As
W97M/Thus.P.....[.....]	12/00	As
W97M/Thus.Q.....[.....]	6/00	As
W97M/Thus.R.....[.....]	12/00	As
W97M/Thus.W.....[.....]	3/01	As
W97M/Titch.A.....[.....]	1/00	As
W97M/Turn.A.....[.....]	12/99	As

W97M/Verlor.M.....	[.....]	2/02	St
W97M/Walker.D.....	[.....]	6/99	Fp
W97M/Wazzu.X.....	[.....]	5/01	As
W97M/Wrench.G.....	[.....]	7/02	As
W97M/Wrench.I.....	[Egerton.....]	5/01	As
W97M/Wrench.R.....	[.....]	12/01	St
W97M/Xthree.A.....	[.....]	5/01	As
WM/Demon.A.....	[.....]	1/02	Zv
WM/DZT.A.....	[.....]	11/01	Zz
WM/Inexist.A:Fr.....	[.....]	11/01	Fp
WM/MDMA.A.....	[StickyKeys.....]	12/01	Zz
WM/Mental.A.....	[.....]	12/00	As
WM/Wazzu.DO.....	[.....]	11/01	Fp
X97M/Anis.A.....	[.....]	2/02	St
X97M/Barisada.A.....	[.....]	8/00	Sk
X97M/Barisada.D.....	[.....]	12/00	As
X97M/Brep.A.....	[.....]	12/01	Rz
X97M/Divi.AQ.....	[.....]	3/02	St
X97M/Divi.B.....	[.....]	9/00	Sm
X97M/Divi.F.....	[.....]	8/00	As
X97M/Divi.Q.....	[.....]	12/00	As
X97M/Extras.A.....	[.....]	4/02	So
X97M/Laroux.AA.....	[.....]	12/01	Ta
X97M/Laroux.AE.....	[.....]	5/00	As
X97M/Laroux.AJ.....	[.....]	10/00	As
X97M/Laroux.AL.....	[.....]	2/01	As
X97M/Laroux.CF.....	[.....]	7/99	As
X97M/Laroux.CN.....	[.....]	9/99	Kd
X97M/Laroux.CS.....	[.....]	12/01	St
X97M/Laroux.D.....	[.....]	3/01	As
X97M/Laroux.HM.....	[.....]	8/02	Ta
X97M/Laroux.IH.....	[.....]	3/01	As
X97M/Laroux.OK.....	[.....]	12/01	Rz
X97M/Laroux.OM.....	[.....]	4/02	Rz
X97M/Laroux.OO.....	[.....]	4/02	Rz
X97M/Pathetic.D.....	[.....]	6/02	St
X97M/Slacker.E.....	[.....]	12/01	St
X97M/Squared.A.....	[.....]	4/01	As
XM/Compat.A.....	[.....]	12/98	Zz
XM/Laroux.DX.....	[.....]	6/02	Ao
XM/Laroux.E.....	[.....]	10/97	Zz
XM/Laroux.G.....	[.....]	2/01	As
XM/Laroux.HO.....	[.....]	12/99	Kd
XM/Laroux.KU.....	[.....]	4/01	Kd
XM/VCX.A.....	[.....]	2/02	Sm

=====
 Total for both lists: 539
 =====

WildList Sorted by Frequency

=====
 This is not a prevalence table. It does not show how common each virus is. Rather it is The WildList sorted by the number of participants that report each virus. This section gives the names, types, and aliases of the most frequently reported viruses. These viruses have been reported by at least 15 WildList participants. They are sorted with the most frequently reported first.
 =====

Freq	Name	Aliases
45	W32/Klez.H-mm	-
45	W32/SirCam.A-mm	-
44	W32/BadTrans.B-mm	29020
43	W32/Magistr.A-mm	Disembowler
41	W32/Nimda.A-mm	-
40	W32/Magistr.B-mm	-
39	W32/Klez.E-mm	-
38	W32/BugBear-mm	-
37	W32/Hybris.B-mm	Hybris.23040-mm
35	W32/Goner.A-mm	-
31	W32/Nimda.E-mm	-

```

30 W32/Funlove.4099          -
28 W32/Elkern.C             WQK.C
27 W32/MTX-m                Apology; Matrix
24 W32/Aliz.A-mm           -
24 W32/Gibe.A-mm           -
24 W32/Yaha.G-mm           Lentin.F
24 W95/Spaces.1445         Busm.1445
23 JS/Kak.A-m              -
23 VBS/Haptime.A-mm        Help
22 W32/Braid.A-mm          -
21 W32/MyParty.A-mm        -
20 W95/CIH.1003            CIH.A; Spacefiller
19 VBS/LoveLetter.A-mm     BugFix; I-Worm
19 W32/FBound.C-mm         -
18 W32/Hybris.D-mm        Hybris.25088-mm
17 VBS/Redlof.A-m         -
17 W32/BadTrans.A-mm       13312
16 VBS/LoveLetter.AS-mm    Plan.A
16 W32/Yaha.E-mm           Lentin.D
15 W97M/Marker.C           Spooky.C

```

Other

The WildList is a list of PC Viruses In-the-Wild. Sometimes WLO receives reports of programs which according to the various reporters, may not fit strictly into the 'PC Virus' category, but which have been brought to their attention by concerned users. The following programs fall into that category.

Name of Virus	[Alias(es)]	List Date	Reported by:
Linux/Lion.Worm.....	[.....]	6/02	Ao
Linux/OSF.8759.....	[.....]	5/02	AmStTa
+Linux/Slapper.B.....	[.....]	12/02	Jk
SunOS/BoxPoison.....	[.....]	5/02	SoZb
+W32/Fregit.B.....	[.....]	12/02	St

Total for Other list: 6

Release notes for the December WildList:

The following viruses -- originally on the supplemental list -- fell from the December WildList:

```

O97M/HalfCross.A          VBS/Tryc
W32/Nymph.A-mm            W32/Borzella.A
W97M/Assilem.B            VBS/Urgent.A
W97M/Assilem.G            JS/Kak.H-m
W97M/Hope.P               VBS/Netlog.R
W97M/Lenni.A              VBS/San.A-m
W97M/Macroble.B           VBS/San.B-m
W97M/Marker.BO            VBS/Simona.A
W97M/Nottice.A            VBS/Urgent.A
W97M/Opey.H               VBS/Vanina.C-mm
W97M/Titch.G              VBS/VBSWG.Y-mm
W32/Music.A-m             VBS/WhiteHome.A
HLLP.10000-mm             W97M/Marker.HK
W32/Toal.A-mm             W97M/Marker.HT
W97M/Astia.A              W97M/Marker.HX
VBS/Brazil.A-mm           W97M/Media.B
VBS/Calhob.A-mm           W97M/Melissa.AL-mm
VBS/Sigsys                 W97M/Thus.FF

```

The WildList welcomes our newest reporters, Greg Romania, Paolo Monti and Sanjay Katkar.

The WildList is collated by WildList Organization International. A complete archive of the official release of WildLists is available at The WildList web site: (<http://www.wildlist.org/WildList/>). The WildList and all material contained on this web site is the copyright of WildList Organization International unless otherwise stated in the material itself. WildList Organization International

permits quotation and citation of The WildList either in whole or in part providing WildList Organization is identified as the source of the material. The WildList may not be altered or misrepresented in any way and only fair usage is permitted. Beyond these limited rights all rights are reserved.

WildList Organization International and WildList reporters make a diligent effort to ensure the accuracy of the data presented in the WildList. However, WildList Organization makes no warranty against the accuracy of the information presented herein. WildList Organization and WildList Organization reporters cannot be held liable for any loss, or damages incurred from the use of The WildList information.

```
=====
WildList Vol.D12 - (c) 1993-2002 Joe Wells - http://www.wildlist.org
=====
```

Appendix F –Results of Pocket PC Scanner Test 2003-05

The full results of the pre-test or “Pocket PC Scanner Test 2003-05” will be given in this appendix. The pre-test used three testbeds for macro ITW viruses and three testbeds of script ITW as listed in Appendix E.1. The test is also available from the student’s test section of the aVTC website.

F.1. Results for script viruses on itwskri.002 (October 31, 2001):

Scanner	Viruses detected		This includes ---- unreliably ---- identified detected				Files detected	
	Testbed	20	100.0%					122
AVP	0	0.0%	0	0.0%	0	0.0%	0	0.0%
FSE	0	0.0%	0	0.0%	0	0.0%	0	0.0%
INO	20	100.0%	5	25.0%	1	5.0%	118	96.7%
PCC	15	75.0%	15	75.0%	5	25.0%	89	73.0%

Remark: decimal ~ indicates that result is rounded:
(100.~ up to 100.0%, 0.~ down to 0.0%).

F.2. Results for script viruses on scr_itw.021 (January 31, 2002):

Scanner	Viruses detected		This includes ---- unreliably ---- identified detected				Files detected	
	Testbed	25	100.0%					34
AVP	0	0.0%	0	0.0%	0	0.0%	0	0.0%
FSE	0	0.0%	0	0.0%	0	0.0%	0	0.0%
INO	25	100.0%	1	4.0%	0	0.0%	34	100.0%
PCC	19	76.0%	19	76.0%	0	0.0%	27	79.4%

Remark: decimal ~ indicates that result is rounded:
(100.~ up to 100.0%, 0.~ down to 0.0%).

F.3. Results for script viruses on scr_itw.024 (April 30, 2002):

Scanner	Viruses detected		This includes ---- unreliably ---- identified detected				Files detected	
	Testbed	24	100.0%					33
AVP	0	0.0%	0	0.0%	0	0.0%	0	0.0%
FSE	0	0.0%	0	0.0%	0	0.0%	0	0.0%
INO	24	100.0%	1	4.2%	0	0.0%	33	100.0%
PCC	18	75.0%	18	75.0%	0	0.0%	26	78.8%

Remark: decimal ~ indicates that result is rounded:
(100.~ up to 100.0%, 0.~ down to 0.0%).

F.4. Results for macro viruses on itwmac.002 (October 31, 2001):

Scanner	Viruses		This includes				Files	
	detected		----	unreliably	----	detected	detected	
	detected		identified		detected		detected	
Testbed	124	100.0%					1337	100.0%
AVP	0	0.0%	0	0.0%	0	0.0%	0	0.0%
FSE	0	0.0%	0	0.0%	0	0.0%	0	0.0%
INO	124	100.0%	8	6.5%	1	0.8%	1336	99.9%
PCC	118	95.2%	118	95.2%	6	4.8%	1276	95.4%

Remark: decimal ~ indicates that result is rounded:
(100.~ up to 100.0%, 0.~ down to 0.0%).

F.5. Results for macro viruses on mac_itw.021 (January 31, 2002):

Scanner	Viruses		This includes				Files	
	detected		----	unreliably	----	detected	detected	
	detected		identified		detected		detected	
Testbed	115	100.0%					224	100.0%
AVP	0	0.0%	0	0.0%	0	0.0%	0	0.0%
FSE	0	0.0%	0	0.0%	0	0.0%	0	0.0%
INO	115	100.0%	2	1.7%	0	0.0%	224	100.0%
PCC	111	96.5%	111	96.5%	1	0.9%	215	96.0%

Remark: decimal ~ indicates that result is rounded:
(100.~ up to 100.0%, 0.~ down to 0.0%).

F.6. Results for macro viruses on mac_itw.024 (April 30, 2002):

Scanner	Viruses		This includes				Files	
	detected		----	unreliably	----	detected	detected	
	detected		identified		detected		detected	
Testbed	99	100.0%					189	100.0%
AVP	0	0.0%	0	0.0%	0	0.0%	0	0.0%
FSE	0	0.0%	0	0.0%	0	0.0%	0	0.0%
INO	99	100.0%	1	1.0%	0	0.0%	189	100.0%
PCC	97	98.0%	97	98.0%	1	1.0%	185	97.9%

Remark: decimal ~ indicates that result is rounded:
(100.~ up to 100.0%, 0.~ down to 0.0%).