Malicious Threats to Personal Digital Assistants

Eric Chien

Symantec Ltd.
Schipholweg 103 2316XC Leiden, Netherlands
Tel 31 71 408 3111 • Email echien@symantec.com

ABSTRACT

In the 1980s, no one left home without his or her FiloFax. Today, no one leaves home without his or her Personal Digital Assistant (PDA). However, while FiloFaxes contained important names and numbers, PDAs are more than just an address book. Combined with Internet access, the functionality of the PDA is moving towards a desktop computer combined with a cellular phone, small enough to put in ones pocket.

As corporations begin to adopt PDAs as a standard computing device within their digital infrastructure and applications become more robust and meaningful with the standardization of wireless computing, threats from malicious code become more serious.

This paper will explore the malicious code threats on the three major PDA platforms (PalmOS, EPOC32, and Windows CE) and the associated Virus Bulletin 2000 presentation will include demonstrations of such threats. Also, potential solutions to PDA threats will be presented including demonstrations of prototype applications in detecting malicious PDA code both on the PDA and associated devices.

2. Background

2.1 Palm

The leading platform for handheld computing devices is Palm OS. According to IDC, Palm OS controlled 78.4% of the handheld market share in 1999. Overall, IDC expects personal digital assistants to exceed 18.9 million units by 2003. With more than 4,000 applications for the Palm OS, devices running Palm OS are at greatest risk of malicious code.

Palm OS does not use a traditional file system. The file system is optimized for synchronization with a primary device (the desktop computer) and for the limited storage area available. Data is stored in memory blocks called records. Related records are grouped in databases where every record belongs to one and only one database. For example, a database may be a collection of all address book entries or all calendar entries.

A database is analogous to a file. The difference is that data is broken down into multiple records instead of being stored in one contiguous chunk. When modifying or accessing a

database, the changes only take place in memory, instead of creating it in RAM and then writing it out to storage.

Palm devices consist both of RAM and ROM. The ROM generally holds the operating system and newer versions of Palm devices allow for flashing of the ROM to potentially update system files. Palm clones such as Handspring devices can also utilize flash cards adding additional memory or functionality. Palm devices do not use the traditional x86 architecture. The Palm uses the latest Motorola 68k Dragonball series CPU.

2.2 Psion

Psion originally started in 1981 by David Potter. Originally under the name Potter Scientific Investments (PSI), the company soon changed the name to Psion. Since that time, Psion has produced a range of personal digital assistants. In 1996 however, Psion split into a separate organization known as Psion Software. The goal would be to license the underlying EPOC operating system to other OEM hardware vendors. EPOC was designed to work in many devices and not just the latest personal digital assistant developed by Psion. By 1998, Psion software was transformed into Symbian a joint venture of multiple mobile phone giants and Psion. The joint venture made it clear smartphones and communicators would arrive in the future utilizing the EPOC operating system.

EPOC runs on 32-bit CPUs and latest releases are designed for the ARM or StrongARM CPUs. Applications are generally developed under Windows initially in PE (portable executable) format using the EPOC emulator. Once the program has been debugged, the code is recompiled for the ARM architecture and then transferred to the EPOC device.

For storage, Psion devices consist of ROM (Read Only Memory), RAM (Random Access Memory), and optional CF (Compact Flash) cards.

The ROM not only contains the operating system, but also all the built-in applications and middleware. In contrast, a standard personal desktop computer contains only the bootstrap loader and BIOS in the ROM. Typically, applications on a desktop computer are stored on a storage device and when executed are loaded to RAM.

However, files located on the Psion ROM are generally executed in place rather than being loaded to RAM first. In addition, as suggested by the name, the ROM can only be read from and not written to. Thus, in general programs located on the ROM can not be modified. OEMs, administrators, and technicians can possibly re-flash or change the ROM with the appropriate software, firmware, and hardware to provide updates to the operating system or built-in applications.

The RAM is analogous to both the RAM and hard drive storage of a personal desktop computer. The RAM in Psion devices contains the additional applications stored by the user, active programs, and the active copy of the system kernel. The RAM disk maintains

data when there is power and on warm boots. On cold boots, the data in the RAM is completely wiped.

EPOC is versatile in both its usage and licensing. In general, OEMs can use the core operating system of EPOC, but develop a completely different ROM. For example, a device with no user interface would not require the user interface capabilities (EIKON) of EPOC and thus, could be removed saving valuable memory. OEMs could also potentially load a ROM image from CF card into RAM, and then mark it as read only. Thus, when referring to EPOC features one must often do so in reference to a particular implementation.

Hardware resources such as the system RAM are isolated from applications via a privilege boundary. Running under privileged mode, the kernel controls all of the device's hardware resources. The CPU will only perform privileged instructions for the kernel. All other user-mode applications that need access to hardware resources must access them via the kernel. Kernel APIs allow such applications to remain on the unprivileged side of the boundary, while the kernel performs the necessary privileged commands.

Finally, executables on Psion devices can be divided into two categories, applications and servers.

Applications contain a user interface and when executed, run in a separate process space. Thus, the system creates a virtual address space for each executed application. In this way, traditional applications can not accidentally overwrite data area of another application. The MMU, Memory Management Unit, manages the separate process spaces so writable memory of one application is not accessible by an another process.

A server is an executable without a user interface. Servers generally provide APIs for clients (clients can be applications or other servers). Some servers actually use the same process.

2.3 Windows CE

Microsoft released Windows CE in 1996. It was designed by to be an operating system that contained a subset of the Win32 API. The requirements were the operating system should be for small ROM based systems. Initial devices that utilized Windows CE were handheld PCs and since then many devices take advantage of Windows CE including personal digital assistants. While Windows CE can be found in everything from the Auto PC to PDAs, this section will concentrate on Windows CE relative to PDAs.

While source code is generally the same when developing a Windows CE application (not all devices support all features in Windows CE), there are many hardware devices with different CPUs which can utilize Windows CE. Thus, the resultant executable may actually differ depending on the target CPU. In fact, the majority of PDAs running Windows CE are not running Intel x86 compatible architectures.

Windows CE programming can be considered similar to Windows programming just with a subset of APIs and potentially reduced hardware resources (no hard disk, and available RAM).

Since Windows malware is possible, Windows CE malware is of course also possible. However, current Windows malware is often specific to the particular Windows operating system. For example, many viruses take advantage of specific undocumented kernel features of Windows NT, which do not exist in Windows CE. In addition, trojan horses often make registry key changes, which also do not exist under Windows CE.

Unfortunately, analogous features can all be found in Windows CE. For example, malware often utilizes the Run registry keys of Windows 9x. A similar key (HKLM\Init\…) can be found in Windows CE that allows for programs to similarly start before the Explorer and thus, allowing stealth capabilities.

In summary, Windows CE runs on a variety of hardware and has been implemented for many CPU architectures. This includes standard personal digital assistants such as the new Microsoft Pocket PC to embedded devices, which contain no user interface.

3. Vectors of Delivery

Any method that allows the introduction of executable code onto the personal digital assistant represents a vector of delivering potentially malicious code. While there are many methods of introducing code, syncing currently represents the primary method and in the future, Internet access will actually pose the greatest threat. Following is a brief description of the potential vectors of delivery for each device.

3.1 Syncing

The primary method by which applications are transferred onto personal digital assistants is via syncing functionality. All three operating systems have associated syncing applications. The Palm device uses HotSync, Psions use EPOC Connect, and Windows CE devices can install the ActiveSync application.

Syncing functionality is used primarily to synchronize data stored on the device with data stored on the desktop computer, back up data to the desktop computer, manage files, and install new device applications that are located on the desktop computer.

Currently, this provides the easiest means of introducing malicious code. For example, to install a new program on the Palm, one may download the new program from the Internet and save it to their desktop computer. Then, using the HotSync functionality, the program is transferred from their desktop computer to the Palm. Now saved to the Palm, the user can run the new program. The program could be anything from a new chess game to a malicious program that emails out all ones contact records.

In addition, the syncing functionality allows custom applications on a desktop PC to connect and manipulate data on the personal digital assistants. For example, the EPOC Connect APIs allow one to interface with the Message Center on the device. The Message Center provides a single point of access to e-mail, fax, and SMS functionality. Such programmability provides the ability of malware to potentially send mail and manipulate files on the device from the desktop computer and vice versa. Fortunately, this is not only advantageous for malware, but for potentially anti-virus solutions as well, both of which will be discussed later.

3.2 IrDA

Personal digital assistants usually contain an IR (InfraRed) port allowing IR communication capabilities. Such capabilities are generally compliant with IrDA (Infrared Data Association) specifications and thus, one can easily interface with the IR capabilities of the device.

However, the majority of devices utilize an associated application to allow the transfer of data via the IR port. For example, the Palm OS Exchange Manager provides a simple interface for Palm OS applications to send and receive data from a remote device using standard protocols. In addition, low level access is obtained through the use of document APIs. Psion allows applications to implement IR communication capabilities via the ESOCK API.

With IR capabilities, devices are able to receive and send applications and thus, potentially malicious code. Currently, devices are designed to trigger an incoming data alert message however, this message can be disabled. Disabling such a message requires specific agent code on the receiving device. Via IR, malicious programs could potentially speak to other infected devices exchanging information and code, all unbeknownst to the users.

3.3 Network Access

The most susceptible gateway of transferring malicious code is via network access. By adding optional modem hardware to the device or utilizing newer wireless models, one has access to many standard Internet protocols. In general, email access with attachments is available and so is web browsing. One can easily receive emails with executables attached, save those attachments, and run them. Such applications can easily contain malicious code.

Devices generally contain pre-installed mail clients, which are also programmable. So, malicious software writers may only need to interface with existing mail clients rather than creating their own network capable agent.

Also, all three operating systems provide libraries to applications to easily establish a connection with any other machine on the Internet and transfer data to and from that machine using the standard TCP/IP protocols. Thus, malicious code is not limited to

utilizing the programmability of the device's mail client or web browser, but can open listening server ports allowing remote access, send confidential data, or receive additional malicious code.

4. Architectural Threats

While the vectors of delivery provide the doors to enter the personal digital assistant, architectural design provides the keys for opening or exploiting those doors.

4.1 Programmability

Many applications running on the devices are programmable. A third party program can interact with the programs through a standard application-programming interface.

Psion devices contain a client-server architecture, which allows one applications to speak to another, Windows CE applications generally utilize common APIs such as MAPI, and Palm applications can send launch codes to each other. Using launch codes an application can direct another application to perform some action or modify its data.

For example, in Palm OS, a malicious program could send a launch code to query all the email addresses in the Address List application. Then, the same program could send a launch code instructing the email application to queue and send email messages with itself as an attachment. All of this functionality can be performed without user input, and without the user's knowledge.

Such programmability easily allows for email type worms like W97M.Melissa and VBS.LoveLetter. How far and how fast such threats may spread is discussed later.

4.2 File System Architecture

All three OS's provide file operation functions. The file streaming functions allow one to read, write, seek to a specified offset, truncate, and do everything else you'd expect to do with a desktop-style file. Such functionality is all that is needed for a viral threat to spread.

In all three operating systems, viral threats may find other applications on the device, append themselves to those applications, and change the entry point of the program to ensure future execution and continued replication.

The Palm and some Windows CE and Psion implementations do not employ any inherent access control to applications and data files. System applications are easily modified as regular user applications. Of course, system applications stored in ROM can not be permanently changed (unless the ROM is flashed). This allows malicious code to not only modify some system files, but also destroy some system files. With a single click, one could wipe out all the applications and data on their device.

Luckily, most devices contain sensitive information on the ROM, which is not writable. However, the amount of middleware or built-in applications that exist on the ROM vary with each operating system and further depending on OEMs for each device. Since, sensitive system information is generally stored on the ROM, uses can reset or cold boot their devices losing applications and data, but still containing a functional PDA. Then, potentially the user can restore their applications and data via synchronization.

4.3 Development Libraries and Languages

Programming libraries in all three operating systems are quite robust. All provide communication libraries, which pose the greatest risk. Windows CE provides a subset of the standard Windows API with the inclusion of Berkeley style sockets; Palm OS is distributed with a range of libraries including the net library and IR library allowing Palm OS applications to easily establish a connection with any other machine on the Internet or in IR port range; and the Psion contains the ESOCK API and ETEL API allowing network connectivity.

Such libraries make programming high-level threats very easily. For example, without low-level knowledge of IR communications, one can easily create an agent that monitors incoming IR data requests. Network libraries allows programmers to create Berkeley sockets style network programs. Such programs could range from a small SMTP engine, creating email capabilities on a device that may not even have a mail client, to a server listening for incoming commands acting as a remote access trojan.

In addition, script languages are available on the Psion and some Windows CE devices. Creating programs in script languages is traditionally easier. VBS.LoveLetter and W97M.Melissa.A were both coded in script languages. By default, Psion for example, offers a script language known as OPL and Windows CE, depending on applications installed, can host a range of other potential script languages for creations of malware.

5. Spreadability

While the creation of viruses, worms, and trojans are all possible for popular personal digital operating systems, their potential in-the-wild spread is influenced by a variety of factors. One should not be surprised if a malicious threat is discovered tomorrow; however, one should be surprised if such a threat posed an immediate widespread threat.

First, while Palm holds the largest market share of personal digital assistant users, the number of personal digital assistant users is magnitudes lower than the number of PC users. In addition, at this moment the number of network connected personal digital assistant users is also magnitudes lower than the number of people with access to email.

Secondly, if the number of personal digital assistant users increases, the number of different operating systems and their customizations still decreases the ability for a threat to spread effectively to anything other than a smaller subset of personal digital assistant users.

Thus, a malicious personal digital worm would not spread nearly as fast as a Windows worm.

Finally, the model of data exchange for personal digital assistants is still asymmetric. Users still download application and data from a few primary sources rather than all PDA users exchanging information with all other PDA users. It is this symmetric nature of code exchange that can dramatically increase the threat of viral spread as seen with macro viruses.

In addition, some operating systems contain further hurdles, which may limit spread. For example, Psion devices require every EIKON application utilize a UID (Unique IDentifier). These UIDs must be obtained from Symbian and thus could potentially be traceable to a particular individual or computer. Programmers can utilize a set of test UIDs prior to obtaining Symbian authorized UIDs; however, a Psion with two applications with the same UID, will only recognize one of the two applications.

So, in order for PDA malware to spread, the cost of PDAs will need to continue to decrease and become standard productivity devices issued in the corporate space. Also, vendors will need to standardize protocols, be cannibalized by market competition, or a clear market leader will need to emerge. Only then does the threat increase dramatically.

If we reach a day where we check email via our PDA and trade documents and other executable attachments via our PDA, the chances of malicious code being inadvertently executed rises. Luckily, solutions to detect PDA malware are already in development.

6. Solutions

The inherent difficulty with creating anti-virus solutions for personal digital assistants is resource limitations. Obviously, the current megabyte signature files can not simply be placed on a digital assistant with limited storage space and any CPU intensive activities such as heuristic emulation may not be possible.

However, in addition to possible device solutions, one can also potentially utilize the current infrastructure and create solutions on the associated device, namely the syncing desktop computer. These systems allow for more CPU intensive activity and greater storage space.

6.1 Current Infrastructure

Current infrastructure provides reasonable first level protection for personal digital assistants. While no anti-virus product currently contains engines to specifically parse the major PDA operating system file formats, the current engines should suffice in detecting initial malware creations. Initial malware creations will probably not require any special new technology and can most likely be detected by current scanning

techniques. Thus, these anti-virus products can provide first level protection in scanning potentially infected samples prior to their transfer to the personal digital assistant.

For example, those utilizing e-mail on their PDA can have e-mail scanned at the gateway and those who primarily utilize syncing can scan executables on the desktop computer prior to installation on the PDA.

In addition, as malware threats develop on PDA operating systems (polymorphism, obfuscated entry point, etc.), engines and signatures can be created to specifically deal with these threats and then deployed within the current infrastructure's ability to handle engine upgrades.

Obviously, as e-mail gateway scanning isn't enough today (one also needs a desktop scanner), neither is the current infrastructure in regards to personal digital assistants. While desktop systems still can be infected via network shares and floppy diskettes, personal digital assistants can also be infected via infrared transmission and potentially other non-scanned network access from Internet webpages to other proprietary network data transfer mechanisms.

In addition, new malware may be transferred to the personal digital assistant prior to the update of signatures on top level scanning systems. Once transferred to the PDA, the malware may remain undetected unless the actual device undergoes scanning.

So, while current infrastructure can clearly mitigate the spread of initial malware for the personal digital assistant, the current infrastructure does not provide full-tier protection leaving the PDA with open vectors of delivery.

6.2 Associated Device Solutions

Personal digital assistants are still closely linked to an associated device, generally the desktop PC. Users of PDAs rarely go a week without performing some sort of synchronization act with their desktop PC. This interaction between the desktop PC and the PDA allows one to potentially place a more robust solution on the desktop computer, which can remotely scan the PDA.

All three operating systems provide synchronization APIs. This allows custom programs on the desktop computer to perform tasks on the PDA remotely when the PDA is linked to the desktop computer.

From the desktop computer, one can scan files on the personal digital assistant, perform intensive CPU or memory activity on the files utilizing the desktop computer resources, and write to the personal digital assistant repairing or removing malware and side effects.

While such scanning can be invoked automatically each time the user performs a syncing operation, the method still requires users to initiate the synchronization with the desktop computer. The time gap between synchronization events leaves the personal digital

assistant open to infection. For example, if the user receives a malicious attachment via email, in order to verify the attachment is not infected, the user will need to first sync their personal digital assistant with the desktop computer. This model of course eliminates a distinct advantage of the PDA – a small device that may receive e-mail independent of the desktop computer.

So, by utilizing associated device solutions, one can increase their protection, but only so much as non-real time scanners do in today's infrastructure.

Associated device solutions are analogous to having the user manually scan for viruses on their desktop computer (if not worse) and draws away from the advantages of having a PDA – one might as well carry around their desktop computer. However, some level of protection is of course better than no protection.

6.3 Device Solutions

All devices contain functionality to traverse the file system, open files, and write files from the device itself. Thus, anti-virus solutions can simply be placed on the personal digital assistant itself. For simple signature scanning, these anti-virus solutions can actually be small enough to fit within the guidelines of resource limited devices. However, dynamic heuristics and emulation techniques would be difficult to implement.

In contrast, malware that requires emulation techniques may be difficult to create for personal digital assistants as well. Malware must also fall under the same restrictions as the anti-virus itself. Thus, malware that spreads effectively will need to be simple and non-resource intensive as well. Some simplicity of the malware itself may deem resource intensive anti-virus unnecessary.

In addition, while signature files may be megabytes in size for current anti-virus solutions, equivalent signature files for PDA malware may not be as large. The average one to two megabyte signature files in the current infrastructure represent approximately fifty thousand viruses. Clearly, there will not be fifty thousand PDA viruses in the near future. In addition, most malware will not be able to execute on all personal digital assistants, but only particular OS's and even then potentially only on certain devices due to OEM customizations. Thus, signature files for PDA solutions will only necessitate signatures for their device in contrast to current desktop solutions which scan for multiple operating system files.

Current desktop solutions need to contain signatures for malware that affects not only their system, but potentially other systems as well, for example, anti-virus on a file server that holds both PC and Macintosh native files.

Personal digital assistants obviously will have difficulty storing current desktop operating system applications and thus, may not require scanning for anything but threats to their device.

Technologies on the device can utilize many of the traditional techniques used today including integrity checking, behavior blocking, and signature scanning. All of these methods are relatively non-intensive and can be completed quickly making realtime scanning non-prohibitive.

6.4 Inherent Protection

Although syncing provides a vector of delivery, syncing also provides inherent backups. Users who utilizing syncing, automatically backup data to their desktop computer. Thus, if they are infected by a virus or malicious software wipes out their applications, they can easily restore from backups, if they are there.

Recall the same syncing event can also allow malicious software on the PDA to modify data stored on the desktop computer. Nevertheless, the model built by PDAs allows one to potentially return easily to a known clean state.

6.5 Non-Technology

While technology provides us with adequate protection, there is only a single silver bullet, user education. User education can never be stressed enough. While there are worms that spread without users executing attachments (VBS.Bubbleboy, Wscript.KakWorm), the majority of worms require a user to execute an attachment.

By educating users to scrutinize messages, many widespread worms can be avoided. Clearly, users should not run attachments from unknown sources, and verify attachments, which come from known colleagues. This applies not only to desktop computers, but personal digital assistants as well. While some can not avoid downloading the latest chess game, users should assure they are downloading traceable (usually commercial) software and not anonymous freeware applications unless they have been verified to be non-malicious.

In corporations where personal digital assistants are widely used, information security administrators should ensure they have evaluated the functionality versus security and potential security risks personal digital assistants pose. In addition, mobile telephones often do not fall within information security sights, but as mobile telephones become smart phones, they should require review by information security administrators.

Information security administrators should also begin the standardization of personal digital assistants and smart phones within their organization. By approving a single personal digital assistant in their corporations environment they will not only alleviate helpdesk calls, but potential security risks as well.

7. Summary

Unfortunately, there isn't a digital device that is 100% secure. To be 100% secure, one should revert to the old FiloFax. However, while there is a threat, there are also potential solutions.

Although, threats are possible, I would not predict a widespread threat at the current time. In order for personal digital assistant threats to achieve the infection rates of VBS.LoveLetter, many developments are still required. The current variety of operating systems and customizations and the current lack of networking versus the desktop computer clearly limits the spreadability.

However, once such executable code is run, the possibilities are limitless. PDA devices, as discussed, are open for infection and can aid email worms by their robust programmability and networking.

Information security administrators should be aware that the threats exist and if they have not done so, yet begin review of the requirement of personal digital assistants and smart phones within their corporate environment. By reigning in these devices now, administrators may save potential headaches later.

APPENDIX A.

The Virus Bulletin 2000 presentation associated with this paper will present the following applications on the PalmOS, WindowsCE, or EPOC platforms. These applications will not be available for distribution. Note that no viruses or malicious programs were created in the research of this paper and presentation. Symantec researchers are ethically and contractually against the creation of any such malware.

1. Ability to iterate through the Address book for email addresses.
2. Ability to programmatically send email with attachments.
3. Ability to automatically launch stealth programs on warm boots
4. Ability to write to application and data files.
5. Ability to delete applications, data files, and system files.
6. Perform remote tasks on the PDA via syncing
7. Stealth communication and data transfer between PDAs via IR ports
8. Networking communications via modem or wireless protocols including the ability to perform remote control access.
9. ROM flashing
10. Prototype anti-virus device solutions.
11. Prototype anti-virus associated device solutions.
12. Current infrastructure scanning solutions.

REFERENCES

[1] http://www.idc.com

[2] http:/www.symbiandevnet.com

[3] http://www.palmos.com/dev/

[4] http://www.microsoft.com/windowsce/

[5] Tasker, Martin, 'Professional Symbian Programming', 2000

[6] Boling, Douglas, 'Programming Microsoft Windows CE', 1998

[7] Mykland, Robert, 'Palm OS Programming', 2000

[8] Palm Computing, Palm OS SDK Reference, Document Number 3003-002, March 2000

[9] Palm Computing, Palm OS Programmer's Companion, Document Number 3004-002, March 2000

[10] Palm Computing, Conduit Programmer's Reference for Windows, Document Number 3012-001, February 2000

[11] Palm Computing, Conduit Programmer's Companion for Windows, Document Number 3013-001, February 2000

[12] Symbian, EPOC Release 5 C++ SDK, Release 004, 1999

[13] http://www.epocworld.com/

[14] ARM Ltd., ARM Instruction Set, ARM Document Number QRC 0001D, June 1995

[15] ARM Ltd., ARM ELF, Document Number SWS ESPC 0003 A-08, September 1999