FEATURE

Palm Breach

Eric Chien
Symantec, Netherlands

In the 1980s, no one left home without their *Filofax*. Today, no one leaves home without their 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, all small enough to put in your pocket.

This article will touch on malicious threats to the *Palm* Personal Digital Assistant as a preview to a presentation at the *Virus Bulletin* conference in Florida in September. The VB2000 presentation will not only provide a more in-depth look at the *Palm*, but also *EPOC32* and *Windows CE* devices. In addition, 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.

The leading platform for handheld computing devices is *Palm* operating system. 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 the 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 a database, the changes only take place in memory, unlike the traditional desktop method of temporarily storing it in RAM and then writing it out to storage. Such memory storage provides a home for new application databases (executable code), which can be introduced in a variety of different ways.

Vectors of Delivery

Any method that allows the introduction of executable code onto the *Palm* device represents a vector of delivering potentially malicious code. While there are many methods of introducing code, 'HotSyncing' currently represents the primary method and, in the future, Internet access will actually pose the greatest threat. In the following paragraphs a brief description of three potential vectors of delivery is presented.

HotSync: The primary method by which applications are transferred onto the *Palm* is via the HotSync functionality. This is used primarily to synchronize data stored on the device with data stored on the desktop computer, back up data to the desktop computer, and install new applications to the *Palm* from the desktop computer.

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

IrDA: The *Palm* contains IR (InfraRed) communication capabilities. Such capabilities are compliant with IrDA (Infrared Data Association) specifications. Thus, the user can directly interface with the IR capabilities of the *Palm*. However, the majority of programs utilize the *Exchange Manager*. The *Palm Exchange Manager* provides a simple interface for *Palm* OS applications to send and receive data from a remote device using standard protocols. With IR capabilities, the *Palm* is 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. This 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.

Network Access: By adding optional modem hardware to the *Palm* or utilizing newer wireless models, one has access to many standard Internet protocols. In general, clipped Web browsing is available and so is email access with attachments. The user can easily receive emails with *Palm* applications attached, save those attachments, and execute them. Such applications could contain malicious code. Also, the net library allows *Palm* OS applications easily to 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 *Palm* mail client or Web browser, but can open listening server ports allowing remote access, sending of confidential data, or receiving additional malicious code. Such network access is an open invitation to fast spreading worms.

While the vectors of delivery provide the doors to enter the *Palm* device, it follows that architectural design provides the keys for opening or exploiting those doors.

Programmability

Many of the applications which run on *Palm* OS are programmable. A third party program can interact with the other programs through a standard application-programming interface. Specifically, applications can send launch codes to each other. Using these launch codes an application can direct another application to perform some action or modify its data.

For example, 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.

File System

The file access functions in *Palm* OS allow the user to read, write, seek, truncate, and do everything else you would expect to do with a desktop-style file. Such functionality is all that is needed for a viral threat to spread. Viral threats may find other application databases on the device and append themselves to those application databases, changing the entry point of the program thereby ensuring future execution and continued replication.

The *Palm* does not employ any inherent access control to databases and records. System application databases are easily modified as regular user applications. This allows malicious code not only to modify system files, but also to destroy system files. With a single click, one could wipe out all the applications and data on the device.

Libraries

The *Palm* OS is distributed with many libraries including the net library allowing *Palm* OS applications to establish a connection easily with any other machine on the Internet, and the IR (InfraRed) library allowing a direct interface to the IR communications. Such libraries make programming high-level threats very easy.

Without low-level knowledge of IR communications, a user could easily create an agent that monitors incoming IR data requests. By monitoring incoming IR requests rogue executables could communicate with other infected devices. Also, the net library allows programmers to create Berkeley sockets-style network programs. Programs like these 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 allowing hackers remote access.

Spreadability

While the creation of viruses, worms, and Trojans are all possible for the *Palm* OS, their potential in-the-wild spread is influenced by a variety of factors. It would not be surprising if a malicious threat is discovered tomorrow; however, it would be surprising if such a threat posed an immediate widespread threat.

Firstly, while *Palm* holds the largest market share of Personal Digital Assistant users, the number of PDA users is magnitudes lower than the number of PC users. In addition, at this moment the number of network connected PDA users is also orders of magnitude lower than the number of people with access to the Internet. Thus, a malicious *Palm* OS application would not spread nearly as fast as, for example, a *Windows* worm.

Secondly, the model of data exchange for PDAs is still asymmetrical. Users still download applications and data from a few primary sources rather than a situation where many PDA users exchange information with many other PDA users. This symmetrical nature of code exchange can dramatically increase the threat of viral spread as demonstrated with macro viruses.

As the cost of PDAs continues to decrease and they become standard productivity devices issued in the corporate space, the threat increases dramatically. If we reach a day where we check email via our *Palm* and trade documents and other executable attachments via our *Palm*, the chances of malicious code being inadvertently executed will rise. In addition to this, if the marketplace consolidates to a single vendor, the susceptibility of the average PDA user will rise. Once such executable code is run, the possibilities are limitless. *Palm* devices as discussed are open for infection and can aid email worms by their robust programmability.

Summary

Palm is only one of many vulnerable devices. Unfortunately, there is not a digital device that is 100% secure. To be 100% secure, one should revert to the old *Filofax*.

However, on the bright side, while there is a threat there are also potential solutions. Those who are interested in further details regarding threats to PDAs or corporations which are beginning to consider a PDA as a standard productivity device are encouraged to attend the *Virus Bulletin* conference this September.

In Florida, I will demonstrate such malicious programs and also prototype solutions, which can detect and block such threats. I will explore in more technical detail the functions that allow threats to be created and some simple steps that can be taken today to reduce one's susceptibility.