Palm*™ Handheld Viruses: A Significant Threat?

Simon N. Foley† and Robert Dumigan,
Department of Computer Science,
University College, Cork, Ireland

Since its introduction in 1997, the Palm range of hand-held computers has become a popular choice, with many millions of units sold worldwide. A large number of third-party applications exist, from personal organizer programs to spreadsheets, electronic mail and web browser clients. Many of these applications are low-cost shareware, available for download from individual web sites, centralized web repositories such as `www.palmgear.com`, or from CD collections. With such a large number of applications available from disparate sources it is interesting to note that only recently (August 2000) was there a report of the first public dissemination of a malicious Palm application. When run, this malicious Trojan-horse program deletes all applications on the handheld; it does not infect other programs and is rated as low-risk by antivirus companies [2]. To date there have been no reports of Palm viruses 'in the wild' [4]. We would like to think that the Palm handheld, at least as it is currently used, is not a very attractive proposition to a malicious virus-writer.

Palm software is distributed as a serialized 'PRC' file of the application which can be stored on the user's local file-server/workstation. When the handheld and the local file-server/workstation are synchronized (hotsync'ed) the PRC file is installed as an executable on the handheld.

A Palm virus spread in this way will have limited success: following initial infection, subsequent spread to other programs is generally limited to the handheld unit alone. Once a program is infected (on the handheld) it must be uploaded to the user's workstation/file server and distributed from there before it can become a threat to other handheld units. While users regularly synchronize databases (containing application data) on the handheld with their workstation, it is not usual to upload programs since the original backup copies are already saved on the user workstation.

It is unlikely that a Palm virus would be accidently distributed by a developer. Palm applications are typically coded and cross-compiled on a local workstation, generating a PRC file ready for downloading. If a development environment does come in contact with a Palm virus then, while infection may occur on development handhelds (or emulators), it should not transfer to the development PRC files so long as these files are not uploaded from the Palm

*Palm is a registered trademark of Palm Inc.

†Correspondence should be directed to Foley at `s.foley@cs.ucc.ie`.

handheld. Of course, there is always the possibility of a PC virus that infects PRC files with a Palm virus.

Once an infected application is downloaded, virus spread *within* a single handheld is quite effective. The Palm Operating System (PalmOS) is designed to indirectly launch applications as the result of certain events. For example, the 'Find' operation invokes each application, in turn, requesting a search for a given string within the application's databases. Other events that lead to indirect invocation include synchronization, alarms, certain operating system preference changes, and soft-reset of the unit.

The paradigm of synchronizing the Palm with a local (or remote) system provides a simple barrier: while an infected program can be installed and spread within a handheld, it cannot easily spread to other programs beyond the handheld unit. With synchronization, Palms should tend to be directly infected from the same source: a virus released in an application will be spread only by that application. This will make reporting and isolating the malicious code easier. However, this synchronization barrier fails once handheld units become more connected. Any protocol that facilitates direct communication between Palm handhelds may facilitate the spread of a virus.

Third-party applications providing alternative methods for installing Palm programs are now emerging. For example, a third party POP/IMAP client runs on the Palm and allows PRC files to be sent and received as Palm email attachments. While the recipient must still explicitly install the attached program, the normal synchronization 'barrier' has been bypassed.

September 1998 saw Palm Computing introduce infra-red irDA support in PalmOS Version 3, allowing programs and data to be 'beamed' directly from one handheld to another. Even though this beaming normally requires explicit user authorization by both sender and receiver, it can nevertheless be used to spread infected programs. While we do not know how prevalent the practice of beaming of applications is, there is reason to believe that it is not uncommon. For example, commenting in a Wired interview on his Palm quickwriting program presented at the 1998 ACM User Interface Software and Technology Symposium, Ken Perlin (New York University) remarked *"By the time I gave my talk, she had beamed it [via the PalmPilot's infrared connection] to several people in the room. It spread through the room like a virus"*

An analogy can be drawn between virus threats to Palm handhelds (PalmOS3) and to Personal Computers during the 1980s before networking was widespread. Bulletin boards have been replaced by downloads from the web, exchanging floppy-disks is similar to beaming, and shrink-wrapped software is not unlike using a reliable web repository or CD distribution. Viruses from that era, such as Brain (1987) and Lehigh (1987) typically spread via Bulletin boards and floppy-disks. If the beaming of Palm applications was to become as prevalent as was PC application sharing via floppy-disks during the 1980s, then one could conjecture that the Palm would be similarly vulnerable to virus infection.

Poorly implemented applications may facilitate the spread of viruses between handhelds. For example, a popular third-party document reader allows the beaming of document databases as application databases between handhelds. However, the recipient of a beamed database has no way of knowing, in advance, whether the beamed database is a document database or is an application database that possibly contains a virus. We suspect that this vulnerability was the result of a coding short-cut, using an existing PalmOS API call that beams applications, rather than developing specialized code for beaming document databases.

The PalmVII introduced in May 1999 uses a two-way radio service and greatly increases the potential for connectivity. At present the service provides instant messaging and indirect access to the Internet. Internet access is achieved via a web-clipping proxy service that 'clips' specific web-pages and delivers the low-bandwidth result to their associated Palm Query Applications on the handheld. While it should be technically possible for a Palm Query Application to use the web-clipping service to deliver a PRC file and subsequently install it, we believe that such an application, if developed, would not be widely used for downloading arbitrary applications. Web-clipping is intended for low-bandwidth Internet queries, taking under 10 seconds to deliver up to 500 Bytes as a result of a typical request from a Palm Query Application [3]. This would make downloading even the smallest Palm applications (1-5KBytes) unattractive.

But what of the future? Using synchronization to install applications can inhibit the spread of Palm viruses, but this resistance diminishes with the development of applications and hardware that facilitate or encourage direct sharing of programs between handhelds. At the time of writing we are unaware of any Palm applications that use active content or mobile code. Given the widespread effect that Macro viruses have had on PCs, we would hope that any developments in this direction would properly address the security issues.

References

[1] F Cohen. *A Short Course on Computer Viruses.* John Wiley, 2 edition, 1994.

[2] Trend Virus Encylopedia. Palm.Liberty.A. http://www.antivirus.com.

[3] Palm Computing. PalmVII connected organizer whitepaper, 1999.

[4] WildList Organization International. http://www.wildlist.org.

Sidebar: Keeping your Palms Clean of Infection

In the current absence of any known Palm viruses in the wild, we make some general recommendations on safely handling Palm programs.

- *Obtain applications from reliable sources.* Palm applications should be downloaded/registered directly from a trusted repository.

- *Check new software before installing.* It is usually a good idea to run any new software first using POSE—an easy to use Palm OS emulator that runs on the user's workstation. This may help to identify potential problems before installation on the handheld unit.
- *Synchronize regularly.* This ensures that up to date 'backups' of application databases are maintained on the user's workstation/file-server. Since regular database synchronization between a handheld and workstation is an integral part of how a handheld is used, we suspect that average users probably have a superior backup regime for their handheld than is common for a PC.
- *Avoid uploading programs from a handheld.* If an infected program is uploaded to the workstation/server, the possibility of subsequent distribution will spread the virus.
- *Remember that applications can be indirectly invoked.* Not launching a suspect application will not stop PalmOS from invoking it.
- *Avoid direct program exchanges between handhelds.* Emailing or irDA beaming a program from one handheld to another is probably the easiest way for a virus to spread.

Sidebar: Anatomy of a Simple Palm Virus

The Palm handheld range is built around the Motorola 68328 DragonBall series processor: a 16MHz micro-controller based on the Motorola 68000 processor. The Palm operating system (PalmOS) is based on a pre-emptive multitasking real-time kernel with (user-interfaced) applications effectively running within a single thread. A memory manager manages that part of memory used to hold applications and data, and is comparable to the file system on a conventional system. This memory is organized as a heap and is structured as a collection of databases, each one storing a program or data. The records of a program (database) provide the code, forms, global variable data and other resources that make up the application.

Both Palm applications and Palm hacks (user provided PalmOS patches) can be used to host and spread a virus. For the purposes of our study, we prototyped a simple virus that spreads by attaching itself to Palm application code resources. The virus was developed using the GNU C tool-set and its size is less than 700 bytes. It has a benign behavior, simply infecting other programs without any malicious attack. Its behavior is not unlike a conventional transient file-style virus which is active only when its host program runs [1].

When an application is invoked, control is passed to its `CODE1` resource, which contains the necessary startup code, followed by the application specific code. Startup code launches the application, looking after the allocation of a new stack-frame, global variables, and so forth. A good place to execute virus code is after the startup and before the application-specific code executes. We

```
on launch of host application:
    find start of viral code in CODE1 resource of host
    select CODE1 resource of random target program
    if target not already infected
       insert host viral code to target
    if triggered then attack, possibly based on launch-code;
```

Figure 1: A Simple Palm virus

coded our virus as a C function, such that, when executed, it searches for its own executable code within its host application's CODE1 resource, appends this code to the CODE1 resource of the target, and then inserts a call to itself between the startup code and application code.

Launching a Palm virus may trigger a variety of attacks from simple data diddling of standard databases such as calendars and address books, to more catastrophic failures, including corruption of PalmOS flash memory. These triggers may be conditioned on the PalmOS events that launch the application. For example, the first time a virus executes, it might send a request to the Alarm Manager to wakeup its host application at some future point in time. This wakeup is recognized as the trigger for the attack; other invocations (of the host) may result in different attacks based on different triggers. Attacks might degrade performance, such as slowing down the response or interfering with power management. For example, efficient power management on the Palm ensures that it can operate for up to two months on just two AAA batteries. A virus that activates the serial port can have a detrimental effect on this.