August 2002

Securing handheld devices

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Securing handheld devices

Abstract

Handheld devices are becoming an indispensable part of everyday life. In this paper, we review the security of handheld devices, which are based on the Pocket PC operating system. We then identify the risks and threats of having these handheld devices connected to the Internet, and propose several methods to protect against the threats. We point out some advantages and security threats of porting server applications to handheld devices. We also consider the newest technology in mobile applications using the Microsoft's NET framework and provide the security risk analysis for it. We conclude the paper with an open problem.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was published as: Susilo, W, Securing handheld devices, 10th IEEE International Conference on Networks, 27-30 August 2002, 349-354.

SECURING HANDHELD DEVICES

Willy Susilo

Centre for Computer Security Research
School of Information Technology and Computer Science
University of Wollongong
Wollongong 2522, AUSTRALIA
Email: wsusilo@uow.edu.au

ABSTRACT

Handheld devices are becoming an indispensable part of everyday's life. In this paper, we review the security of handheld devices, which are based on Pocket PC operating system. We then identify the risks and threats of having these handheld devices connected to the Internet, and propose several methods to protect against the threats. We point out some advantages and security threats of porting server applications to handheld devices. We also consider the newest technology in mobile applications using the Microsoft's .NET framework and provide the security risk analysis for it. We conclude the paper with an open problem.

1. INTRODUCTION

Handheld computers/devices have come a long way in just a few short years, but currently they have become indispensable part of everyday's life. Handheld devices comprise the largest group of Internet connectable devices [10]. Their popularity is due to their size, lightweight and fit into pockets. The main part of a handheld device is a *stylus* which is used to tap on a touch screen to activate applications or enter data. They provide the flexibility that is needed by an increasing number of mobile work force, and equipped with a wide range of advanced functionalities which are predicted to overtake other forms of communication in near future.

Currently, there are two major operating systems that are applied on handheld devices: Palm OS and Windows CE. We also note that there are several other operating systems ported to handheld devices, such as Linux [13, 37] and Java [9]. However, so far they have not gained popularity due to the lack of applications that have been written for these platforms.

Palm 1000 was introduced by 3Com in 1996. Devices running Palm OS have dominated the handheld market and currently having around 83% of the market share [38]. Palm OS handhelds are ultraportable. They offer a wide range in functionality and price. Data transfer can be done easily, both in synchronizing to a desktop PC and between Palm users (via IrDA). Palm OS is also adopted by companies such as HandSpring and IBM with applications ranging from Palm OS native applications including Address Book and Date Book, to other applications that are available as freeware, shareware or commercial products from third-party providers, including fax software, calculators, and software development kits for developers. Nevertheless, Palm OS devices are known to be insecure [15].

The second major operating system for handheld computers is Windows CE [10] which takes almost the rest of the market share. Windows CE is developed by Microsoft with the aim to extend today's Personal Computer platform to mobile computers. The latest version of Windows CE, also known as Pocket PC, is licensed to a wide variety of companies, such as Casio, Compaq and Hewlett-Packard. The operating system starts from the same premise as Palm OS. It provides essential PIM (Pocket Information Manager) style applications on the handheld, as well as expansion, synchronization and handwriting recognition. However, the implementation is dramatically different from manufacturer to manufacturer. The OS is condensed to work on a PDA (Personal Data Assistant) with a small display and no keyboard. The Pocket PC operating system is immediately familiar to Windows users. The most current version of the OS is Pocket PC 2002. Pocket PC 2002 directly supports mobile-phone hardware to provide a standard user interface for phone functions across all devices. Therefore, this operating system can be directly applied to create a Pocket PC based mobile phone (such as Hewlett-Packard's Jornada 928 and Fujitsu-Siemens's Pocket Loox). Currently, the successor of Pocket PC 2002, known as Windows CE .NET, is being developed by Microsoft [23].

There have been several studies in the security of Palm OS devices, for example [11, 15, 16, 17, 34]. However, there is no report regarding the insecurity of Pocket PC devices. In this paper, we will discuss the threats to Pocket PC handheld devices and the techniques used to secure them. Securing such devices includes providing protection for, wireless communication, stored data and against viruses and malicious codes, together with making provisions for the theft of the device. We consider the Pocket PC handheld device when it is used by itself and also when it is connected to the Internet (either via an Ethernet Compact Flash/PCMCIA card, Bluetooth or via a modem).

The rest of the paper is organized as follows. In the next section, we briefly explain Pocket PC (or Windows CE based) handheld devices. In section 3, we review the threats of using a Pocket PC handheld, together with some possible applications that could prevent against those risks. In section 4, we present the new risks of having a Pocket PC in a network environment that is connected to the Internet and suggest a way to protect the device against malicious codes or users from the Internet. In section 5, we describe new risks of porting server applications to Pocket PC devices, in addition to its advantage of having such applications. In section 6, we discuss the newest technology introduced by Microsoft (known as .NET platform) and its relation with the use of handheld devices in the mobile environment. Section 7 concludes the paper.

0-7803-7533-5/02/$17.00 ©2002 IEEE.

2. POCKET PC HANDHELD DEVICES

2.1. Windows CE/Pocket PC Operating System

Windows CE (or Pocket PC) is a new operating system, which is not a port from Windows NT or Windows 9x [28]. Windows CE was designed to run in the smallest devices that has limitation on their memory footprint of ROM and RAM. In Windows CE, the dynamic library DLL has been redesigned in a new DLL called `COREDLL.DLL` which is the "heart" of Windows CE. This DLL contains some functions of many Windows NT core DLLs like `kernel32.dll`, `user32.dll` and `gdi32.dll` [28]. If a new application is created and it only uses functions of the `COREDLL.DLL`, then this application can be run on every device that claims to be running Windows CE. Windows CE provides an "instant-on" feature which means that Windows CE starts and stops instantly. With the newest version, Pocket PC 2002, it even allows for real-time applications [24].

2.2. Pocket PC Hardware

There are mainly four types of Pocket PC hardware, namely Palm-size Pocket PC (Ps/PC), Pocket-size Pocket PC (P/PC), Handheld PC (H/PC) and CEPC (a standard desktop PC running Windows CE). Starting from Pocket-size Pocket PC version 1.1 (equipped with Windows CE 2.11), a standard Windows CE device supports the ADO, MFC, ATL and eVB code [28]. The standard processor used in the current Pocket PC device is a Strong ARM which runs at 206 MHz.

2.3. Development Systems

There are several development systems available for Pocket PCs. However, the most common one is produced by Microsoft and it is known as Embedded Visual Tools (eVT) [21] which is available free of charge. eVT includes two languages which are inherited from its predecessor, Microsoft Visual Studio 98, namely Embedded Visual C++ (eVC) and Embedded Visual Basic (eVB). Windows CE API calls can be incorporated in each program written in eVT. eVT also includes a utility called Remote Heap Walker which makes development in Pocket PC easier for creating a bug-free code. Database access is also supported by SQL Server 2000 version for Windows CE known as SQL Server CE [26].

2.4. Synchronization

A Pocket PC device can be synchronized with a desktop PC through a synchronization software called "ActiveSync" [19, 20]. The synchronization can be done through either a serial port, a USB port, an infrared port, a Bluetooth "virtual" port or a network (Ethernet connection) and Remote Access Service (RAS) server.

ActiveSync serves four major purposes: data synchronization, file management, file backup and software installation. Files from a Pocket PC device can be copied to a desktop PC or vice versa by incorporating ActiveSync. ActiveSync will perform all the conversion needed (for example, when a user wants to upload/download a Microsoft Word file or a Pocket Word file).

Synchronization is the process of keeping information consistent on a Pocket PC device and a PC. What and how information is synchronized is stored in a *partnership*, which is created between Pocket PC devices and PCs by using ActiveSync's Partnership wizard. When the same piece of information is changed on both the PC and the device, ActiveSync will identify the conflict and provide the user with the opportunity to designate which change the user wants to keep. ActiveSync resolves conflicts according to the conflict resolution configuration which is set by the user.

2.5. Expansions

Pocket PC devices usually support expansion slots, which include PCMCIA slot (either Type I or II), Compact Flash (CF) slot and Secure Digital (SD) slot. SD slot is becoming a new standard for a Pocket PC device which is started by Compaq iPaq H3800 series. From these expansion slots, the Internet connection, via modem, wired and wireless LAN or Bluetooth connection, can be performed.

3. THE RISKS OF POCKET PC HANDHELD DEVICES

Due to its small size, handheld devices are often easily lost or stolen. The main problem in this matter is the value of the data which might be of value to competitors. For example, there have been a number of high profile cases where confidential data has gone astray when a laptop has been stolen from government officials. Fortunately, Pocket PC device is protected by a power-on password, which is built into the hardware itself [25]. We note that the length of the password is not enough especially with the *domain* of the password which normally only includes numbers. In contrast to Palm OS device, the password mechanism is done through software and hence it can be stolen [34]. However, we note that if the device uses removable storage, such as removable CF cards or removable SD cards, then the data on these should also be encrypted and password protected. An application such as CryptoGrapher for Windows CE [30] can provide such protection by encoding every information each time it is recorded on the storage and then decoded when it is read. When 25 misidentifications are received, then all the information kept on the storage is destroyed.

To date, Pocket PC OS have not yet to become a significant target for malicious code (c.f. Palm OS [11]). However, when the feature of Pocket OS is improved in the future, for instance by allowing a macro VBA program in Pocket Office applications, then malicious codes will have a new target. This feature has been attempted by several third party software, for example ClearVue Suite [36] which can read a Microsoft Office document (without being translated to Pocket Office format by ActiveSync). To protect against this possibility, developers of anti virus software are beginning to acknowledge the need to protect the enterprise against malicious software on handheld devices [5, 18]. There are two approaches that the anti virus software uses. The first approach is to "host" the anti virus in a PC and it will scan all the files in the handheld whenever the device is synchronizing with the PC (e.g. McAfee VirusScan). The second approach is to treat the handheld device as if it is a normal PC, by installing the anti virus to the device itself (e.g. PCCillin [31]).

Authentication during synchronization is also an important point that needs to be protected [25]. By appearing as a registered user, the *hacker* can steal data or compromise the system in some way. When the device is connected to the network, this requirement becomes stronger. With handheld devices having smaller memory and power capabilities compared to a normal PC, much of the access rights and authentication work needs to be done at the net-

work end, with as little software as possible forced upon the user [25]. Strong authentication such as smart cards or RSA Security's SecurID tags [32], which generate a one-time, non-repeatable numeric passcode can also be built in a handheld device to support a strong authentication.

Public Key Cryptography Algorithms vs Handheld Devices

Public key cryptography algorithms are the main cryptographic tool for unique identification of users and digitally signing electronic documents. Both of these operation are are indispensable in a building trust in transactions. The amount of computation required by the mostly widely adopted public key algorithm, that is RSA, immediately rules out rules its direct application. Due to its limited memory, handheld devices might not be able to perform some computations which involve a public key cryptography algorithm. Server-aided versions of this algorithm [2, 3] distributes the computation between the handheld device and the server such that the server will not learn the secret stored in the card and at the same time the more expensive part of the computation is performed by the server. Elliptic curve cryptography (ECC) algorithms provide high security with much shorter keys and have been proposed. ECC algorithms provide high security with much shorter keys compared to RSA. The level of security offered by 1,024 bits key RSA algorithm can be achieved with only 160 bits ECC algorithm [4]. This has enabled ECC to be implemented in low powered devices such as smart cards [4].

4. THE RISKS OF HAVING POCKET PC HANDHELD DEVICES CONNECTED TO THE INTERNET

To provide a mobile solution to the enterprise, the "anywhere capabilities" needs to be attached to handheld devices [7]. To achieve this purpose, handheld devices must be connected to the Internet, so every user can enjoy the *availability* of information whenever they need. Pocket PC devices can be connected to the Internet by using one of the following methods: via a modem and telephone, via a wired LAN connection, via a WiFi 802.11b wireless LAN connection, via a Bluetooth connection, via a TCP/IP connection with the help of a PC and via Infrared connection with a cellular phone (Figure 1). In contrast to an ordinary mobile phone, Pocket

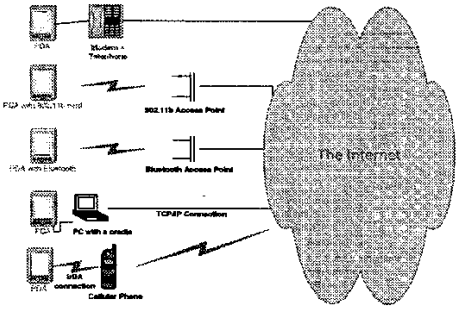

Figure 1: *Several ways used by a PDA to connect to the Internet*

PC OS supports the real online Internet (Web) via Pocket Internet Explorer and WAP is supported through third-party browsers (such as Klondike WAP browser [1]) as well as off-line support for disconnected usage [7]. We note that with PPC2002 Operating System, WAP is supported as part of the Pocket Internet Explorer [19].

With the introduction of 3G mobile phones, very high data rate and seamless connectivity for a free roaming user are promised [33]. This enables a seamless connection from a handheld device to the Internet with the "always-on" connectivity [12, 33].

Threats

By connecting the handheld devices to the Internet, the new security threats will arise. We will focus on the external threats directed either to the handheld devices or to the PC connected to the handheld devices. The scenario of a handheld device together with another PCs is shown in Figure 2. The purpose of the attacker is

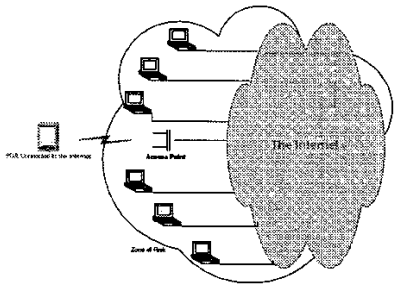

Figure 2: *Zone of Risk of a PDA connected to the Internet*

either to attack the handheld device itself or to use the handheld device as a *tool* for attacking the other PCs. For example, a malicious user can upload a file which contains a macro virus to the handheld device which has no harm, but when the handheld sends this file as an attachment to the PCs around it, then these PCs will be infected. Before discussing about the way to protect against these attacks, firstly we need to define the "zone of risk", as shown in Figure 2. We identify that the PCs in the network (where the handheld is attached) will be threatened by any "new" malicious code that could be sent by the handheld device. On the other hand, the PCs can be used as a way to infect the handheld device because a malicious code that works in one platform will not be compatible to the other platform.

It is noted in [14] that in this kind of environment, issues concerning security and trust are becoming very important. Due to the distributed nature of the networks, there is no longer physical aspect of security and the concept of user authentication to a domain is not possible. A solution that is provided in [14] is by introducing a distributed trust among the distributed environment. This solution only considers the way to share the available resource in the network, however it does not protect against a malicious code inserted to the network once the trust is granted. This means that the zone of risk is still vulnerable against malicious codes that might be inserted by a *temporary* trusted user in the network.

We propose two solutions to protect the zone of risk, which should be incorporated to protect against the threats mentioned above. The first solution is by placing a firewall in the zone of risk [29] as shown in Figure 3. The external screening router accepts packets transmitted from the handheld device that contains the handheld's IP address approved by the internal system administrator. The bastian host is responsible for receiving and forwarding email, for providing connections through incoming FTP and telnet requests and for answering DNS queries. The internal screening router accepts packets that have valid IP addresses that are part of the internal network and accepts packets that request connections

by specifying appropriate port numbers with respect to the set of rules specified for packet filtering.

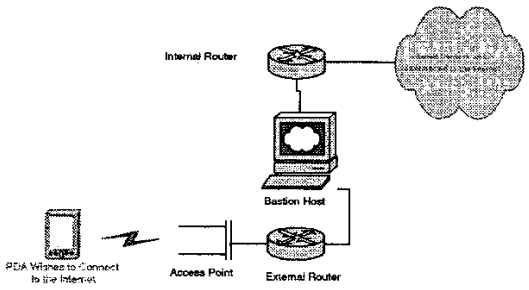

Figure 3: *Schematic Solution of Model with a Firewall*

The first solution works well, but when the device is carried to the other networks (which does not have such protection as above) and it is connected again to the network, then the other problem might arise. For instance, an infected file that has been uploaded to the handheld could infect the other PCs in the network.

To protect against this problem, in the second solution, we provide a personal firewall that is installed in each handheld device. This personal firewall will guarantee that although the device is used outside the predetermined network, the handheld can still "survive" against the attacks from the external malicious users.

Having such combination between personal firewall and the configuration in Figure 3, a secure environment can be achieved. In the following, we will describe the design and implementation of a personal firewall on handheld devices. The implementation is based on the Pocket PC SDK (Software Development Kit), which is very similar to the one for Handheld PC SDK.

Design and Implementation of a Personal Firewall for Handheld Devices

Pocket PC SDK provides a mechanism to support TCP/IP programming via WinSock Control programming [8]. The program can be written either using eVT, ASP or ASP.NET pages with code written using Microsoft Visual Interdev or Microsoft Visual Studio .NET to access data directly through ADO (ActiveX Data Objects) or ADO.NET or to use middle-tier COM+ components written using Visual C++, Visual Basic 6.0 or Visual Studio .NET. The data can be returned in HTML or XML.

For our purpose, we demonstrate the HTTP request with a simple code written in eVB as shown in Figure 4. Once a connection can be made successfully, the `InternetReadFile` command can be used to retrieve the data. To enable the program to act as the packet filtering software, we can use the method as shown in Figure 5. Using the above method, a packet filtering software can be constructed in a Pocket PC device. Adding new capabilities to the current platform is also possible by extending the idea mentioned in this section.

A more general type of Firewall implementation can be obtained from an open source code such as Firewall Toolkit (FWTK) [35] which is portable to handheld devices which use Linux as their operating system.

```
Dim s On Error Resume Next
WinSock1.RemoteHost = txtServer.Text
WinSock1.RemotePort = 80 'HTTP Port
WinSock1.Connect
txtOutput.Text = ""
If Err.Number <> 0 Then
   MsgBox "Error: " & Err.Number & \\
vbCrLf & Err.Description
End If
s = "GET " & txtResource.Text & \\
vbCrLf & vbCrLf
WinSock1.SendData s
If Err.Number <> 0 Then
   MsgBox "Error: " & Err.Number & \\
vbCrLf & Err.Description
End If
```

Figure 4: *A simple HTTP request written in eVB*

```
If(!InternetReadFile(oHTTPRequest, \\
        charBuffer, 500, dwRead)
   'Perform the IP and packet filtering.
   'The rules are stored in a text file
   'that will be read in this function.
End If
```

Figure 5: *Performing Packet Filtering in Pocket PC SDK*

5. NEW RISKS: PORTING SERVER APPLICATIONS TO HANDHELD DEVICES

An interesting feature of Pocket PC devices is their Pocket PC SDK which is equipped with networking ability. Using this feature, handheld devices can be turned into a "server" which provides information to the other PCs or handheld devices connected to the intranet corporate or Internet. For example, porting BSD telnet protocol, BSD SSH (Secure Shell) protocol, FTP server daemon or Web server can be done using this feature.

Having a web server in a handheld will have some advantages in some circumstances. Consider a situation where a meeting is held in one particular meeting room, but the information is needed instantenously by the other people in the other room. This situation is common during the press meeting where the news are the 'hot' topic that needs to be printed immediately. Having a web server installed in a handheld would become a nice solution. The person in the meeting room can use his handheld to write the information needed in his device, and this information will be broadcasted to the other room instantenously only by using a Web browser.

Having an FTP server installed in a handheld is another good feature (Figure 6). With this implementation, a handheld user can exchange information with his host computer (or his corporate network) with a normal FTP client. The information in the handheld can be fetched from any other PCs or handhelds by using a normal FTP client.

Threats

Having a server protocol installed in a handheld will have some

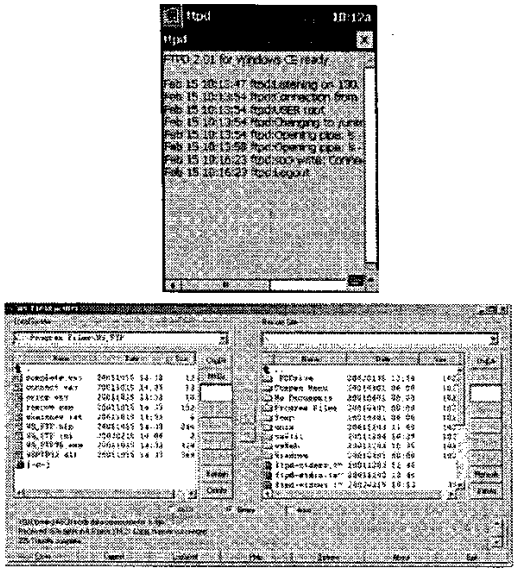

Figure 6: *An FTPD installed in a PPC iPaq Handheld Device and a connected FTP client session*

advantages. However, this will produce new security threats, such as malicious codes, which could be uploaded by running an FTP server. Moreover, the malicious user could also destroy the contents of the handheld if the server is not configured properly.

Having a personal firewall installed (as discussed in the previous section) can solve the problem. The routine in the packet filtering section needs to be updated with the possibility of the malicious codes received and that will protect against the threats to the device.

6. A NEW TARGET: MICROSOFT'S .NET FRAMEWORK

As mentioned earlier, recently there are a lot of new mobile devices, which include Pocket PCs, Palm OS devices, J2ME devices and many others. We note that in the big picture, they require a completely different approach to development. For example, creating an application for WAP phones, Web browsers and PalmOS devices has therefore meant creating a different version of the application for each unique client platform. As the number of unique client platforms multiplies exponentially over the next decade, this approach will become quite impossible.

The alternative answer to this problem is answered by the newest Microsoft technology: Microsoft .NET. The purpose of the .NET platform is to enable the developer to write the application once and expect it to work with any client without significant customizations [6, 22]. This idea is different from the idea of Sun's Java technology. The key difference is illustrated as follows. When a programmer writes an application in Java, then he is not writing it for any device or computer in particular. Instead, he is writing it for something known as a *virtual machine*, where different virtual machine implementations exist for a wide variety of different platforms. With .NET platform, the programmer writes an application for Microsoft's virtual machine called the *Common Language Runtime* (CLR), which runs on a Windows server and not the client devices themselves. It is the CLR's responsibility to adapt the application's output for consumption by different kinds of devices, rather than to run on the devices themselves.

Using the new framework, an application can be written once and it should be targeted to the CLR. By default, .NET framework supports Pocket PC (Pocket IE), PC (Internet Explorer), WML/WAP and cHTML (i-Mode). To enable the framework in some other platforms such as PalmOS and J2ME, a .NET Compact Frameworks (.NETcf) can be used. Microsoft has integrated the .NETcf completely and seamlessly into the new Visual Studio architecture called Visual Studio .NET.

Threats

Having a new framework like .NET framework enables an application to be run in multiple platforms. When a mobile device is connected to a system that supports this framework, a new security threat appears. A multi-platform malicious code can be created and invented and it is more difficult to protect. Code Red virus [27] showed how infection can spread through Web servers. This concept can be applied to the .NET framework and therefore it can infect multi-platform mobile devices. When this case happens, it is very difficult to determine whether a malicious code is running due to different forms that are produced by the .NET framework. The way to protect against this threat is still an interesting open problem for future research.

7. CONCLUSIONS

In this paper, we reviewed the security of handheld devices based on Pocket PC. We identified the threats of having such device connected to the Internet and proposed several methods to protect against these threats. We also considered the possibility of porting server applications to handheld devices which will have some useful applications, but we also noted that these applications will also open a new security threat towards the enterprise. Finally, we considered the new Microsoft's .Net framework and its new security challenge. We exposed an open problem that is caused by a malicious code that can be run in multiple platforms.

8. REFERENCES

[1] Apache Software. Klondike WAP Browser. *Online: http://www.apachesoftware.com/.*

[2] N. Asokan, G. Tsudik, and M. Waidner. Server-supported signatures. *Journal of Computer Security vol. 5 no. 1*, pages 91 – 108, 1997.

[3] D. Boneh, N. Modadugu, and M. Kim. Generating RSA keys on a handheld using an untrusted server. *The First International Conference on Cryptology in India, Indocrypt 2000, Lecture Notes in Computer Science 1977*, pages 271 – 282, 2000.

[4] Certicom. Elliptic curve cryptosystem for smart cards. *Online: http://www.certicom.com/research.html.*

[5] Computer Associates. InoculateIT for CE. *Online: http://www3.ca.com/ Solutions/ProductFamily.asp?ID=128.*

[6] D. Ferguson. *Mobile .NET.* APress Publisher, 2002.

[7] C. Forsberg and A. Sjöström. *Pocket PC Development in the Enterprise.* Addison Wesley, 2002.

[8] N. Grattan. *Pocket PC, Handheld PC Developer's Guide.* Prentice Hall, 2002.

[9] N. Haines and N. Beaulieu. Java Applications are taking the Pocket PC. *Pocket PC Magazine electronic edition*, March 2002.

[10] U. Hansmann, L. Merk, M. S. Nickous, and T. Stober. *Pervasive Computing Handbook.* Springer-Verlag, Berlin, 2001.

[11] D. Harris. The "Liberty" Crack: The first Palm OS Trojan Horse. *Online: http:// www.sans.org/ infosecFAQ/PDAs/ liberty.htm.*

[12] L. Harte, R. Levine, and R. Kikta. *3G Wireless Demystified.* Osborne McGraw-Hill, 2002.

[13] O. W. Hoie. Sharp launches new Linux-based handheld. *InfoSatellite Online. Available at http:// www.infosatellite.com/ news/2001/12/ h071201sharp_sl550.html.*

[14] L. Kagal, T. Finin, and A. Joshi. Trust-Based Security in Pervasive Computing Environments. *IEEE Computer*, pages 154 – 157, 2001.

[15] Kingpin and Mudge. Security analysis of the Palm Operating System and its weaknesses against malicious code threats. *Proceedings of the 10th USENIX Security Symposium*, pages 135 – 151, 2001.

[16] W. Knight. Palm Pilot open to a Denial of Service Attack. *Online: http://www.zdnet.co.uk/news/1999/44/ns-11273.html.*

[17] N. Leavitt. Malicious code moves to mobile devices. *Computer, December 2000*, pages 16 – 19, 2000.

[18] McAfee.com. Virus Scan Wireless. *Online: http:// www.mcafeeb2b.com/ products/ virusscan- wireless/ default.asp.*

[19] F. McPherson. *How to do everything with your Pocket PC, second edition.* Osborne McGraw-Hill, 2002.

[20] Microsoft. ActiveSync 3.5. *Online: http:// www.microsoft.com/ mobile/ pocketpc/ downloads/ activesync35.asp.*

[21] Microsoft. Embedded Visual Tools 3.0. *Online: http://www.microsoft.com/mobile/ downloads/emvt30.asp.*

[22] Microsoft. Microsoft .NET. *Online: http:// www.microsoft.com/ net/.*

[23] Microsoft. Microsoft Windows CE .NET demonstrates wireless leadership with Bluetooth Qualification. *Online: http://www.microsoft.com/PressPass/press/ 2001/Oct01/10-30TaliskerBluetoothPR.asp.*

[24] Microsoft. Mobile Devices: Pocket PC 2002. *Online: http:// www.microsoft.com/ mobile/ pocketpc/default.asp.*

[25] Microsoft. Pocket PC Security. *Online: http://www.microsoft.com/mobile/ enterprise/papers/security.asp.*

[26] Microsoft. SQL Server CE. *Online: http://www.microsoft.com/ sql/CE/default.asp.*

[27] Microsoft TechNet. Cumulative Patch for IIS: CodeRed Worms. *Online: http:// www.microsoft.com/ technet/ treeview/ default.asp?url=/technet/ security/bulletin/MS01-044.asp.*

[28] C. Muench. *The Windows CE Technology Tutorial.* Addison Wesley, 2000.

[29] U. Murthy, O. Bukhres, W. Winn, and E. Vanderdez. Firewalls for Security in Wireless Networks. *Proceedings of the 31st Hawaii International Conference on System Sciences (HICSS '98)*, 1998.

[30] Paragon Software. Powerful Crypto Protection of your Windows CE devices. *Online: http:// www.penreader.com/ WinCE/ CryptoGrapher.html.*

[31] PC-Cillin. Trend Micro PC-cillin for Wireless 2.0. *Online: http:// www.antivirus.com/ download/ documentation/ netdesk/files/readme.txt.*

[32] RSA Security. RSA SecurID. *Online: http:// www.rsasecurity.com/ products/ securid/.*

[33] R. Safavi-Naini, W. Susilo, and G. Taban. Towards Securing 3G Mobile Phones. *The 9th IEEE International Conference on Network (ICON 2001)*, pages 222 – 227, 2001.

[34] @Stake. PalmOS Password Retrieval and Decoding. *Online: http:// www.securiteam.com/securitynews/ PalmOS_Password_Retrieval _and_Decoding.html.*

[35] TIS FWTK. Firewall Toolkits. *Online: http://www.fwtk.org/fwtk/.*

[36] Westtek. Westtek ClearVue. *Online: http://www.westtek.com/clearvue.htm.*

[37] M. Williams. Korean start-up works hard to Pocket Linux. *InfoWorld News Online. Available at http:// www.infoworld.com/ articles/ pi/ xml/00/04/17/ 000417pilinuxpda.xml.*

[38] W. V. Winkle. Palm Business Applications. *The ultimate guide to mobile computing: 2002 mobile Technology, Special Issue Laptop*, pages 82 – 89, 2002.