# Assigning and enforcing security policies on handheld devices

**Article**



**7 authors**, including:

Serban I. Gavrila
National Institute of Standards and Technology
**29** PUBLICATIONS **3,172** CITATIONS

SEE PROFILE

Vlad Korolev
University of Maryland, Baltimore County
**29** PUBLICATIONS **518** CITATIONS

SEE PROFILE

# Assigning and Enforcing Security Policies on Handheld Devices[1]

**Wayne A. Jansen, Tom Karygiannis**
**The National Institute of Standards and Technology**
**Gaithersburg, Maryland, USA**
**{Wayne.Jansen, Tom.Karygiannis}@NIST.Gov**

**Serban Gavrila**
**VDG, Inc.**
**Chevy Chase, Maryland, USA**
**Serban.Gavrila@NIST.Gov**

**Vlad Korolev**
**University of Maryland, Baltimore County**
**Baltimore, Maryland, USA**
**vkorol1@csee.UMBC.edu**

**Abstract** - The proliferation of mobile handheld devices, such as Personal Digital Assistants (PDAs) and tablet computers, within the workplace is expanding rapidly. While providing productivity benefits, the ability of these devices to store and transmit corporate information through both wired and wireless networks poses potential risks to an organization's security. This paper describes an approach to assigning and enforcing an organization's security policy on handheld devices. The approach relies on the device holding a valid policy certificate, obtained through synchronization with a user's desktop computer, organizational server, or other means, before conducting any security-sensitive operations. The paper describes a proof-of-concept implementation of the policy certificate issuing tool, policy specification language, certificate representation, and enforcement mechanisms that were used to demonstrate this approach, and discusses the associated benefits and drawbacks.

Keywords: Trust Management, Handheld Devices, Digital Certificate, Security Policy

[1] Contribution of the National Institute of Standards and Technology

# Assigning and Enforcing Security Policies on Handheld Devices

## Introduction

With the trend toward a highly mobile workforce, the acquisition of handheld devices such as Personal Digital Assistants (PDAs) and PC tablets is growing at an ever-increasing rate. These devices offer productivity tools in a compact form and are quickly becoming a necessity in today's business environment. Many manufacturers make handheld devices using a broad range of hardware and software. Handheld devices are characterized by small physical size, limited storage and processing power, restricted stylus-oriented user interface, and the means for synchronizing data with a more capable notebook or desktop computer. Typically, they are equipped with the capability to communicate wirelessly over limited distances to other devices using infrared or radio signals. Many handheld devices can also send and receive electronic mail and access the Internet. While such devices have their limitations, they are nonetheless extremely useful in managing appointments and contact information, reviewing documents, corresponding via electronic mail, delivering presentations, and accessing corporate data. Moreover, because of their relatively low cost, they are becoming ubiquitous within office environments, often purchased by the employees themselves as an efficiency aid.

From a risk perspective, several major issues loom over the use of such devices [2, 5], including the following items:

- Because of their small size, handheld devices may be misplaced, left unattended, or stolen.
- User authentication may be disabled, a common default mode, divulging the contents of the device to anyone who possesses it.
- Even if user authentication is enabled, the authentication mechanism may be weak or easily circumvented.
- Wireless transmissions may be intercepted and, if unencrypted or encrypted under a flawed protocol, their contents made known.
- The ease with which handheld devices can be interconnected wirelessly, combined with weak or no authentication of the parties involved, provides new avenues for the introduction of viruses or other types of malicious code, and also other forms of attack such as a man-in-the-middle attack.

For example, a business associate can unknowingly beam a Trojan horse application from her PDA to a colleague’s PDA through an IrDA port. The victim can subsequently introduce the malware to the corporate network when he synchronizes the PDA to his desktop computer. Given that the malware was not analyzed by the corporate firewall, the PDA can inadvertently serve as a channel through which network vulnerabilities are exploited. Similarly, the user can browse the Internet using a PDA via a third party ISP and download or upload data or applications that violate the corporate security policy. In short, the PDA has multiple access points over which the corporate security officer cannot exercise any control or influence.

Ideally, the enterprise security officer would like to be able to express, monitor, and enforce a corporate security policy for handheld devices that reduces or eliminates such common threats. Despite the fact that security mechanisms such as data encryption and anti-virus software are becoming available for them, handheld devices typically lack sufficient controls to enforce use of the available mechanisms in accordance with a prescribed corporate security policy.

## Overview

Several tasks need to be performed before a policy certificate can be assigned to and enforced on a handheld device. These tasks include determining how to specify and issue the policy, how to protect and

distribute the policy, and how to validate and enforce the policy. The method we devised attempts to perform these tasks as straightforwardly as possible. The key component in our approach is the use of a digital certificate called a *policy certificate* that is capable of bearing policy settings assigned by some policy-setting authority. After the policy-setting authority generates the certificate, it can be stored on either the user's desktop computer or a corporate policy-certificate server where it can then be distributed to a handheld device during synchronization. Figure 1 illustrates this form of certificate distribution. Alternatively, the user may have the policy certificate stored on an add-on hardware module such as a smart card. The underlying principle is that, until it holds a valid policy certificate, an enforcement mechanism at the device observes default policy settings, which restrict certain kinds of actions, such as external communications other than to obtain a certificate. Once a certificate is obtained and validated, the enforcement mechanism observes the new restrictions drawn from the policy settings within the certificate. The validity period on each certificate ensures regular periodic synchronization and downloading of new replacement certificates.

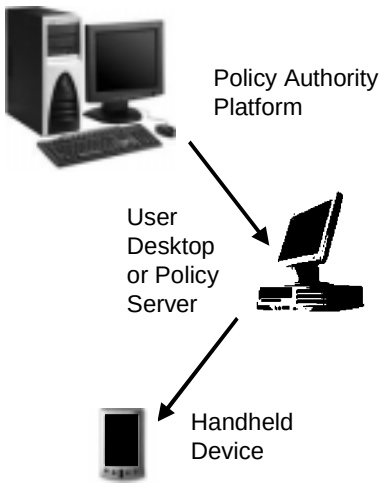


**Figure 1: Policy Certificate Distribution**[2]

Policy settings mainly govern the way communication resources of the device may be used, particularly with respect to its wireless interfaces. At the most rudimentary level, this may involve simply enabling or disabling an interface for a class of resource. Resources include information related to Personal Information Management (PIM) utilities and other applications. For example, if a particular wireless access protocol is considered unsafe, it can remain disabled entirely or selectively enabled for applications that handle mundane information, such as calendar or contact entries. Besides controlling information flow, the policy enforcement mechanism also maintains an audit log of any attempts to breach policy. The audit log is stored on the desktop computer after synchronization and retrieved by the policy-setting authority for analysis.

# Policy Certificate

The policy certificate is a structured set of information that conveys the policy assigned to an entity. A set of policy entries comprises the policy. Besides the policy entries, the elements of the certificate indicate from whom (i.e., the issuer) and to whom (i.e., the owner) the certificate was issued, its period of validity, and supplemental information needed to establish its authenticity and apply the policy. A digital signature over the other elements protects the certificate from tampering as well as attempts to forge the issuer's

[2] Compaq, the Compaq Logo, Evo, iPAQ, and the iPAQ Pocket PC product design are trademarks of Compaq Information Technologies Group, L.P. in the U.S. and/or other countries. The Compaq product images are used with permission, but such use is not intended to suggest affiliation with or sponsorship by Compaq.

signature on another certificate. In order to verify the signature and establish the authenticity of the certificate, a device must hold the corresponding public key of the issuer.

Figure 2 illustrates the elements of the policy certificate, which closely follow the form and content of X.509 attribute certificates. Instead of using X.509 certificates per se, however, we use XML to define and generate the external representation for the policy certificates, rather than rely on an ASN.1 encoding. Many aspects of certificate handling were simplified by using a human-readable representation for the policy certificate.

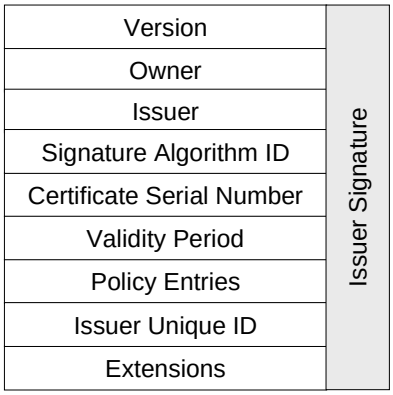


**Figure 2: Policy Certificate Elements**

In our initial implementation, a domain authority issues a single policy certificate to a device. However, the overall scheme, devised to support mobile agent systems [3], allows the flexibility for policy certificates to be issued to both users and devices, and the composite policy enforced at the device. The issuer's identification is established through an X.509 certificate, while a serial number or other uniquely assigned device identifier serves to identify the device.

# Policy Specification Language

The policy language follows a grant-style form of specification by which security-relevant actions are denied on a device unless enabled by a policy entry. Policy entries are a triple of action, source, and target fields:

- Source refers to objects on the device, such as a calendar or address book application on a PDA, able to perform some action.
- Action refers to security-relevant operations performed with the device, such as synchronizing with a home platform, or beaming information.
- Target is an optional field that refers to external points of interface or reference needed to complete the semantics of the operation, typically by adding more specificity such as qualifying the origin or destination of data to be exchanged.

The initial range of policy emphasizes control over the flow of information to and from external interfaces, augmenting rather than replacing existing access control mechanisms within the operating system. Figure 3 illustrates an example of the types of policy that can be supported and its expression using XML elements and attributes.

We attempted to describe actions common to most PDAs, such as infrared beaming, host synchronization, and wireless communication, in a generic fashion. Besides information flow controls, we have also looked at policies for controlling multiple authentication mechanisms on a device and performing various levels of audit. Having generic policy rules allows the same set of policy entries to be applied to different handheld device platforms.

```
<policyEntries syntax="PDAPolicy">
        <policyEntry source="addrbook" action= "sendbeam" target="*" />
        <policyEntry source="datebook" action="sync" target="myPC" />
        <policyEntry source="explorer" action="connect" target="*.mil" />
</policyEntries>
```

**Figure 3: Example Policy Statements**

For added flexibility, the "syntax" attribute allows a policy syntax other than our own PDAPolicy to be conveyed within the "policyEntries" element. Support for alternative syntaxes allows for experimentation with language enhancements or support of other types of policy specification languages should the need arise. The policy certificate itself is also structured to be able to convey multiple policies within the same certificate. This would allow, for example, the ability to specify the policy for a Java virtual machine environment, whose availability is increasingly becoming a supported feature on mobile devices, in conjunction with the policy targeted for the device operating system.

We have implemented a prototype policy tool that allows an administrator to select from supported source, action, and target values, and build up a set of policy entries for assignment. An existing policy certificate can also be used as the starting point for a new certificate, which simplifies the reissuing of expired certificates or the construction of slightly different but related policy. Once the policy entries are complete, the tool can automatically generate a well-formed policy certificate signed by a policy-setting authority. The policy tool can also be used by anyone to validate a policy certificate. For example, a user having problems with his device may wish to validate its certificate, if visual inspection of the certificate using any textual display tool fails to divulge the cause of the problem.

# Policy Distribution

Because the policy rules ultimately affect the behavior of a device, they must always be protected from tampering and forgery. Using a policy certificate to convey policy entries provides inherent protection, requiring only an authenticated distribution from a trusted source. This facilitates the distribution process greatly, allowing a variety of ways for policy certificates to be handled and distributed, including authenticated Web services, secure electronic mail, secure file transfer, and authenticated device synchronization.

Synchronization is the term used to describe the common means of coordinating the update of the information content on a host and handheld device to the same level. To move the policy certificate from a user's desktop, a centralized policy server, or some other host onto a device, we devised a special policy conduit for the synchronization process. For instance, on the Palm operating system, Palm OS[3], a HotSync Manager runs on a Windows host, monitoring one or more communication ports for a command from a handheld device. When a synchronization command is received, the HotSync Manager interrogates the device to determine which synchronization operations need to be run and systematically invokes registered conduits. The conduit is a dynamic link library that interfaces to the HotSync Manager program and performs the updates. Besides delivering a policy certificate to the device, the policy conduit also retrieves the audit log from the device and reports any irregularities encountered.

[3] Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

# Policy Enforcement

The policy enforcement mechanism resides on the handheld device and ensures that the user adheres to the security administrator's security policy settings specified within a valid policy certificate stored on the device. The mechanism starts up as the device is initialized and checks the issuer's signature on the policy certificate for authenticity, the well-formedness of the contents, and whether the validation period is in effect. If a policy certificate is not held or is found to be invalid, the enforcement mechanism applies a default policy having limited privileges. As mentioned earlier, the default policy blocks all information flows to external interfaces with the exception of allowing limited synchronization to obtain a valid certificate. When parsing the policy certificate, the enforcement mechanism extracts and sets aside the policy entries in a table for later use. It also provides a display for the user to review the entire certificate, including its policy contents.

Once the certificate has been validated and the security policy settings installed, the enforcement mechanism mediates any user attempts to perform security-relevant actions. By consulting with the policy settings, the enforcement mechanism grants or denies the requested action. The enforcement mechanism records security relevant events to an audit log maintained on the device. The log entries include events associated with such actions as attempts to synchronize information, beam data, or adding modules to the device. The audit log is returned during synchronization, allowing a means for the policy-setting authority to collect and review logged information. While all security-relevant actions could be logged, in consideration of the limited memory of the device, the audit mechanism has been set up to record only attempts to violate policy. We also envisioned the need for selective event auditing, based on the needs of policy-setting authority, but have not addressed the issue further.

The proof-of-concept implementation of the enforcement mechanism was developed for the Palm OS. When the enforcement mechanism is activated, it traps certain system calls and redirects them to its own routines. Thus, when an application requests the Palm OS to perform a particular action by executing a trapped system call, an internal enforcement routine is called first. The routine determines which application is currently running and then tries to find an entry in the policy table that corresponds to the application attempting the system call and the action being attempted. If a matching entry is found, the action is granted and control passes control to the system call as if no intervention occurred. Otherwise, if no entry is found, the enforcement routine denies the action, sounding an alarm, displaying an error message, if possible, and eventually returning an error code to the calling application without ever calling the system routine.

# Benefits and Drawbacks

As with any security method, our approach has both pros and cons. Some of the more obvious ones are discussed below.

- Policy certificates, being signed objects, self-protect against tampering and their authenticity is easily determined, which avoids the need for a trusted distribution process involving, for example, a hardened certificate server and encrypted communications.
- The same policy can be applied readily to multiple devices. Moreover, by stating policy rules generically, the same policy can apply to different families of handheld devices, yet be issued by the same policy generation tool.
- The general scheme we follow, which was developed originally for mobile agents, is quite flexible and able to support multiple certificates issued by different policy authorities for various entities. For example, it would be a straightforward extension to issue and enforce similar policy controls on any Java-based mobile code downloaded into the device.
- Unique device identifiers are relied on to bind a certificate to a device. For example, Palm OS handheld devices may use one of several methods for unique identification including flash ID, Mobile Access Number (MAN), device ID, and Electronic Serial Number (ESN). While some devices maintain such information in memory, others do not, requiring some other method for recording an identifier on the device. For example,

recording an identifier when the operating system is loaded into flash ROM may be one possibility. Though this process requires additional effort, it also provides the opportunity to assign hierarchical identifiers that, in turn, may serve as a group identifier for policy assignment (e.g., *.group.NIST identifies all devices in the subdomain group.NIST).

- Because the PDA's operating system serves as the foundation for the policy enforcement mechanism, it is a critical factor in the effectiveness of our scheme. An operating system having a weak security architecture or unresolved vulnerabilities undermines the policy enforcement mechanism. Similarly, any flaws in the policy enforcement mechanism may create new vulnerabilities in the operating system.

Our proof-of-concept implementation revealed several weaknesses, both in our implementation choices and in the design and organization of the Palm OS [4], which affected the security of the enforcement mechanism. They include the following items:

- The policy table resides in the handheld memory; therefore, it may be modified by rogue conduits as well as applications on the handheld.
- Rogue applications can perform disallowed actions by bypassing the Palm OS system calls and talking to the hardware directly.
- Rogue applications can patch the system call table or call the routines within the Palm OS kernel directly.
- A rogue hardware module could be constructed to copy the entire contents of the handheld device into its own non-volatile memory, once it is engaged.

We believe that many of these problems can be overcome by using an operating system that provides memory protection, domain enforcement, file and resource access controls, and process isolation. Windows CE and various distributions of Linux targeted for PDAs are examples of some potential alternatives to using the Palm OS. As it is a critical security component, the policy enforcement mechanism needs to be implemented within the operating system kernel, making Linux a particularly compelling alternative for our future research, because it is an open source distribution.

It is important to note that our approach is intended to help well-intentioned users comply with a corporate PDA security policy. A determined user with malicious intent can likely find a number of ways to circumvent the prescribed policy, such as transferring sensitive data to devices or storage media that do not have similar policy controls.

# Related Work

United States Patent 6,158,010. [6] describes a system and method for maintaining security in a distributed computer network. The scheme involves a policy manager located on a server that manages and distributes security policy, which specifies a user's access privileges to securable components, and an application guard located on a client that manages access to securable components as dictated by the security policy. The approach does not utilize policy certificates or signed policy objects. Thus, a secure implementation in general would require trusted distribution of the policy from a protected server to the application guard, in order to prevent attacks on the policy content.

The area of trust management [1] aims toward a comprehensive approach to specifying and interpreting security policies, credentials, and relationships in order to authorize security-related actions within distributed systems. Trust management engines typically do not directly enforce policy, and instead provide a policy decision to the application that invokes it. Strictly speaking, the overall policy computation mechanism we employ [3] is not a trust-management engine according to their use of the term, because rendering a decision is done in an application-dependent manner. However, the key components of a trust management system are addressed, which include a language for describing actions, a mechanism for identifying principals, a language for specifying application policies, a language for specifying and delegating credentials, and a compliance checker.

# Conclusions

The ability for a policy-setting authority, such as a security officer, to control information flow and other policy settings on a handheld device is an area that holds promise for improved security, yet has not received much attention. The approach we took is one that is relatively straightforward and flexible, and one that we believe is suitable for many organizational environments. The approach mitigates external threats by specifying the conditions under which information can be exchanged with the handheld device, and mitigates internal risks by not only specifying, but also enforcing, the corporate handheld security policy.

In order to overcome some of the weaknesses we encountered in our proof-of-concept implementation, a more robust operating system environment is needed. Strong mutual authentication between devices, workstations, and servers is also desired for improving policy distribution and when accessing corporate networks. In addition, support of smart cards by the handheld device would provide better user authentication and an alternate way to bring credentials and key material to the device. These are areas that we plan to address in our future work.

# References

[1] Matt Blaze, Joan Feigenbaum, John Ioannidis, Angelos D. Keromyti,.The Role of Trust Management in Distributed Systems Security, In Secure Internet Programming, Jan Vitek and Christian Jensen (Eds.), Springer-Verlag Lecture Notes in Computer Science (LNCS 1603), July 1999.

[2] Anup K. Ghosh, Tara M. Swaminatha, Software Security and Privacy Risks in Mobile E-Commerce, Communications of the ACM, Vol. 44, No.2, February 2001, pp.51-57.

[3] Wayne A. Jansen, Determining Privileges of Mobile Agents, Computer Security Applications Conference, December 2001.

[4] Kingpin and Mudge, Security Analysis of the Palm Operating System and its Weaknesses Against Malicious Code Threats, USENIX Security Symposium, August 2001.

[5] Neal Leavitt, Malicious Code Moves to Mobile Devices, IEEE Internet Computing, Vol. 5, No.4, July/August 2000, pp. 16-19.

[6] Moriconi, et al., System and Method for Maintaining Security in a Distributed Computer Network, United States Patent 6,158,010, December 5, 2000.