

@stake, Inc.

www.atstake.com

## Security Advisory

Advisory Name: Palm OS Password Lockout Bypass  
Release Date: 03/01/2001  
Application: Palm OS 3.5.2 and below  
Platform: All Palm OS Devices  
Severity: Passwords and data can easily be obtained through a backdoor in Palm OS, even if the device is "locked".  
Author: Kingpin [kingpin@atstake.com]  
Vendor Status: Vendor responded via email, see response section  
CVE: CAN-2001-0157  
Reference: www.atstake.com/research/advisories/2001/a030101-1.txt

### Executive Summary:

The Palm operating system (OS) Security application provides "system lockout" functionality in which the Palm device will not be operational until the correct password is entered. The password is also used to protect and hide records by the legitimate user by marking them as "Private". These mechanisms are meant to prevent an unauthorized user from reading data or running applications on the device.

A backdoor exists in Palm OS which provides source- and assembly-level debugging of executables and the administration of databases existing on the physical device. Although this backdoor is documented for debugging purposes, it can be activated even if the Palm OS lockout functionality is enabled. This will allow an unauthorized user to perform a number of commands including, but not limited to, retrieving an encoded form of the system password, obtaining all database and record information on the device, and installing or deleting applications.

The system lockout mechanism is currently assumed by most users to be a sufficient protection feature of the Palm operating system. This is not the case and is a severe weaknesses for particular deployments of Palm OS devices.

### Overview:

The security implications of the Palm OS backdoor are further amplified when read in conjunction with [1], which details the weak, reversible encoding scheme used to protect the Palm system password on the device. Once the actual password is determined, the system lockout functionality can be disabled and a user's private data can be accessed.

Although well known in the security industry to be insecure, Personal Digital Assistants (PDAs) are ubiquitous in enterprise environments and are being used for such applications as one-time-password generation, storage of medical and company confidential information, and e-commerce [2]. It is important that those specifying

Palm OS devices for use in their organizations be aware of the discussed flaws so that policies and procedures can be put in place to help mitigate risk.

In order to enable the backdoor mode, the attacker must have physical access to the target Palm device. By nature, portable devices face the threat of physical attack external to the office environment. Additionally, the threat of physical attacks internal to a company is very real and will increase as the use of portable devices becomes even more common.

#### Detailed Description:

Designed into the Palm OS is an RS232-based "Palm Debugger". By entering a short Graffiti keystroke combination (shown on page 81 of [3]), the Palm OS device enters one of two interfaces provided by the Palm Debugger and monitors the serial port for communication. "Console mode" interfaces with a high-level debugger, such as Metrowerks Codewarrior for Palm OS, and is used mostly for the manipulation of databases. "Debug mode" is typically used for assembly- and register-level debugging. A soft-reset of the Palm device will exit debug mode, leaving no evidence of use.

Aside from the specific attack of retrieving the obfuscated system password block by using the 'export 0 "Unsaved Preferences"' console command, which is shown in the Proof-of-Concept section, it is possible to access all database and record information on the entire Palm OS device. A complete listing of console and debug commands can be found in [3]. A selection of the most pertinent, and potentially damaging, console commands are as follows:

- cardformat - Format the memory card.
- changerecord - Replaces a record in a database.
- coldboot - Initiates a hard reset on the device. A hard reset erases all data, restoring it to a factory new state.
- del - Deletes a database from the device.
- dir - Displays a list of the databases on the device.
- dm - Displays memory for a specified number of bytes.
- export - Copies a Palm OS database (or application) from the handheld device to the desktop computer.
- import - Copies a Palm OS database (or application) from the desktop computer to the handheld device. This will sidestep any HotSync or beaming operations and logging mechanisms.
- launch - Launches an application on the handheld device.
- saveimages - Saves a memory card image (all of the data on the device).

Because the debug modes communicate with the host via the serial port, this attack can trivially be carried out in a portable fashion using a laptop and HotSync cable or cradle. Additionally, it is possible to use a Palm OS-based application to emulate the required commands and, with a modified HotSync cable, be used for the retrieval of passwords or other data.

Proof-of-Concept:

The Palm OS password is set by the legitimate user with the Security application. The maximum length of the ASCII password is 31 characters. Regardless of the length of the ASCII password, the resultant encoded block is always 32 bytes. The encoded password block is stored in the "Unsaved Preferences" database on the Palm device (along with being transmitted over the serial or network port during a HotSync operation).

Even if the system is "locked", it is possible to use the Palm debug "Console mode" to retrieve the Unsaved Preferences from the device as shown below. The actual password can be determined as described in [1] using the PalmCrypt tool, which will encode and decode ASCII passwords to encoded password blocks and vice versa.

<--- begin console mode screenshot --->

Ready...

>export 0 "Unsaved Preferences"

Unsaved Preferences  
Getting info on resource 3 of 3  
Exporting resource 3 of 3  
Success!!

<--- end console mode screenshot --->

The Unsaved Preferences database is saved to the desktop PC in the .\PalmDebugger\Device directory. Using a hex editor or Palm database viewer tool on the PC, the 32-byte encoded password block can be located. Entering the encoded block into the PalmCrypt tool, the original ASCII password can be determined.

<--- begin palmcrypt screenshot --->

E:\>palmcrypt -d  
568CD23E994B0F8809021345070413440C08135A3215135DD217EAD3B5DF5563

PalmOS Password Codec  
kingpin@atstake.com  
@stake Research Labs  
<http://www.atstake.com/research/>  
August 2000

0x74 0x65 0x73 0x74                      [test   ]

<--- end palmcrypt screenshot --->

Temporary Solution:

Mitigating the risk of a backdoor has been historically difficult without an upgrade to the offending application (in this case, Palm OS). Palm OS 4.0, due to be released at the end of 2001, appears to have resolved the issue of weak password obfuscation [1] and the activation of debug functionality during the "system lockout" mode. However, it is highly recommended that a thorough analysis of OS 4.0 takes place before a security-critical application is deployed.

The most immediate recommendation would be to not use the current family of Palm devices for the storage of sensitive or confidential information. Beware of the security ramifications of other PDAs, as well. It is not possible to employ a secure application on top of an insecure foundation. Because Palm OS is inherently insecure, methods to attempt to completely secure data are moot. The U.S. Government is beginning to follow this recommendation [4].

The user should be very aware of the physical security and location of the device at all times. It should not be left unattended or loaned to a potentially untrustworthy colleague. A PDA lock such as the Kensington PDA Saver could be used, or a lanyard such as Force.com's The Bond.

Because the debug modes are accessible only through the serial port, a hardware add-on with a lock could be used which will prevent a physical connection to the port. For urgent deployments, a piece of plastic could be permanently glued into place (leaving the infrared port as the only method of HotSync). The serial port could be disabled at the circuit board level on particular Palm OS devices by opening the case and cutting the specific RS232 lines. These actions will prevent an attacker from using the debug mode even if it is activated.

Solutions exist which provide power-on protection similar to the Palm OS built-in functionality by requiring a handwritten signature, physical button taps, or other form of password before allowing access to the device. Encryption solutions, such as Secure Memopad by Certicom or Jawz, Inc. Datagator, will encrypt the data of certain Palm applications. If the secret components (e.g., encryption keys or passwords) are not stored on the Palm device, these will serve as an additional layer of security for particular deployments.

However, this does not completely mitigate all risks. Because it is possible to install applications onto the Palm OS device while the "system lockout" functionality is enabled, an adversary could install a keystroke monitoring program in which any passwords could be recorded and retrieved at a later date. This recorded data could be retrieved through the debug mode, as well. The possibility also exists for such a program to save the contents of memory after cryptographic operations take place, hence retrieving the plaintext of encrypted memos, keys, or other data.

It would behoove Palm, Inc. to completely remove all debugging features from future production versions of Palm OS, including OS 4.0. For purposes of application development, a debug-enabled ROM set should be available (and can be used in conjunction with the Palm OS Emulator

on a desktop PC). If the debugging functionality remains inherent in Palm OS, attackers may find methods to modify the operating system to re-enable the debug mode.

#### Vendor Response:

Vendor responded via email that Palm OS 4.0 will fix the problem when it ships.

#### Common Vulnerabilities and Exposures (CVE) Information:

The Common Vulnerabilities and Exposures (CVE) project has assigned the following name to this issue. This is a candidate for inclusion in the CVE list (<http://cve.mitre.org>), which standardizes names for security problems.

CAN-2001-0157

#### References:

- [1] Kingpin, "Palm OS Password Retrieval and Decoding," @stake Security Advisory, September 26, 2000, <http://www.atstake.com/research/advisories/2000/a092600-1.txt>.
- [2] Forbes.com, "Padlock Your Palm," <http://www.forbesbest.com/0226/060.html>.
- [3] Palm, Inc., Palm OS Programming Development Tools Guide, DN 3011-002, <http://www.palmos.com/dev/tech/docs/devtoolsguide.zip>.
- [4] Federal Computer Week, "The Circuit," February 5, 2001, <http://www.fcw.com/fcw/articles/2001/0205/news-circuit-02-05-01.asp>.

Advisory policy: <http://www.atstake.com/research/policy/>  
For more advisories: <http://www.atstake.com/research/advisories/>  
PGP Key: [http://www.atstake.com/research/pgp\\_key.asc](http://www.atstake.com/research/pgp_key.asc)

Copyright 2001 @stake, Inc. All rights reserved.

@stake, Inc.  
[www.atstake.com](http://www.atstake.com)

Security Advisory

Advisory Name: PalmOS Password Retrieval and Decoding (A092600-1)  
Release Date: 09/26/2000  
Application: PalmOS 3.5.2 and below  
Platform: All PalmOS Platform Devices  
Severity: Moderate. Passwords can easily be obtained and decoded allowing an attacker to access all private records on a Palm device.  
Author: Kingpin [kingpin@atstake.com]  
Contributors: DilDog [dildog@atstake.com]  
Vendor Status: Vendor Response Included  
Web: www.atstake.com/research/advisories/2000/a092600-1.txt

#### Executive Summary:

PalmOS offers a built-in Security application which is used for the legitimate user to protect and hide records from unauthorized users by means of a password. In all basic built-in applications (Address, Date Book, Memo Pad, and To Do List), individual records can be marked as "Private" and will only be accessible if the correct password is entered.

It is possible to obtain an encoded form of the password, determine the actual password due to a weak, reversible encoding scheme, and access a user's private data. In order for this attack to be successful, the attacker must have physical access to the target Palm device.

The threat of physical attacks internal to a company is very real and this advisory makes the point that security is not limited to the network/internet arena. The private records often contain passwords, financial data, and company confidential information. Our experience with physical audits has revealed that most users of Palm or other portable devices do not realize that their private information could possibly be accessed by unauthorized users.

#### Overview:

During the HotSync process, the Palm device sends an encoded form of the password over the serial, IR, or network ports to the HotSync Manager or HotSync Network Server on the desktop. The password is transmitted to enable the Palm Desktop program to protect the user's private records when being accessed on the desktop machine. However, based on an encoding scheme of XOR'ing against a constant block of data, the encoded password is easily decoded into the actual ASCII version of the password. The encoded block is also stored on the Palm device in the Unsaved Preferences database, readable by any application on the Palm device.

The transfer of a secret component (i.e. password), even if it is encoded or obfuscated, over accessible buses (serial, IR, or network) is a very risky design decision and is oftentimes considered a design flaw. It is unfortunately common practice that applications choose to simply obfuscate passwords instead of using encryption. Without proper encryption methodologies in place, the task of determining the secret data is greatly simplified as shown in this research.

This advisory is an attempt to remind users and developers of the common problem of storing secrets and the reliance on simple obfuscation.

#### Detailed Description:

The password is set by the legitimate user with the Security application. The maximum length of the ASCII password is 31 characters. Regardless of the length of the ASCII password, the resultant encoded block is always 32 bytes.

It is possible to obtain the encoded password block in a number of ways:

- () Retrieve from the "Unsaved Preferences" database on the Palm device.
- () Monitor the serial or network traffic during an actual HotSync.
- () Imitate the initial HotSync negotiation sequence in order to obtain the password (which is transmitted by the target device). This is demonstrated in our proof-of-concept tool written for the PalmOS platform.

The Palm desktop software makes use of the Serial Link Protocol (SLP) to transfer information between itself and the Palm device. Each SLP packet consists of a packet header, client data of variable size, and a packet footer [Palm OS Programmer's Companion, pg. 255]. During the HotSync negotiation process, one particular SLP packet's client data consists of a structure which contains the encoded password block (Figure 1).

```
struct {
    UInt8 header[4];
    UInt8 exec_buf[6];
    Int32 userID; // 0
    Int32 viewerID; // 4
    Int32 lastSyncPC; // 8
    UInt8 successfulSyncDate[8]; // 12, time_t
    UInt8 lastSyncDate[8]; // 20, time_t
    UInt8 userLen; // 28
    UInt8 passwordLen; // 29
    UInt8 username[128]; // 30 -> userLen
    UInt8 password[128];
};
```

Figure 1: Structure sent during the HotSync process which contains the encoded password block.

Two methods are used to encode the ASCII password depending on its length. For passwords of 4 characters or less, an index is calculated based on the length of the password and the string is XOR'ed against a 32-byte constant block. For passwords greater than 4 characters, the string is padded to 32 bytes and run through four rounds of a function which XOR's against a 64-byte constant block. It is unknown why disparate methods were implemented. By understanding the encoding schema used, it is possible to essentially run the routines in reverse to decode the password, as shown in our proof-of-concept tools. Details of each method are described below.

Neither encoding schema make use of the username, user ID, or unique serial number of the Palm device. A common practice often used for

copy-protection purposes is to use a unique identifier as input into an encoding or encryption algorithm, which PalmOS does not do. The resultant encoded password block is completely independent of the Palm device used and makes it easier to determine the original ASCII password from the block.

Passwords of 4 characters or less:

By comparing the encoded password blocks of various short length passwords (Figure 2), it was determined that a 32-byte constant (Figure 3) was being XOR'ed against the ASCII password in the following fashion:

```
56 8C D2 3E 99 4B 0F 88 09 02 13 45 07 04 13 44
0C 08 13 5A 32 15 13 5D D2 17 EA D3 B5 DF 55 63
```

Figure 2: Encoded password block of ASCII password 'test'

```
09 02 13 45 07 04 13 44 0C 08 13 5A 32 15 13 5D
D2 17 EA D3 B5 DF 55 63 22 E9 A1 4A 99 4B 0F 88
```

Figure 3: 32-byte constant block for use with passwords of length 4 characters or less

Let  $A_j$  be the  $j$ th byte of A, the ASCII password  
 Let  $B_k$  be the  $k$ th byte of B, the 32-byte constant block  
 Let  $C_m$  be the  $m$ th byte of C, the encoded password block

The starting index,  $i$ , into the constant block where the XOR'ing should begin is calculated by the following:

$$i = (A_0 + \text{strlen}(A)) \% 32;$$

The encoded password block is then created:

```
C_0 = A_0 XOR B_i
C_1 = A_1 XOR B_{i+1}
C_2 = A_2 XOR B_{i+2}
C_3 = A_3 XOR B_{i+3}
C_4 = B_{i+4}
.
.
.
C_31 = B_{i+31} (wrapping around to the beginning of the constant
                block if necessary)
```

Example:  $0x56 = 0x74$  ('t') XOR  $0x22$   
 $0x8C = 0x65$  ('e') XOR  $0xE9$   
 $0xD2 = 0x73$  ('s') XOR  $0xA1$   
 $0x3E = 0x74$  ('t') XOR  $0x4A$

Passwords greater than 4 characters:

The encoding scheme for long length passwords (up to 31 characters in



length) is more complicated than for short length passwords, although it, too, is reversible.

First, the ASCII string is padded to 32 bytes in the following fashion:

Let  $A_j$  be the  $j$ th byte of A, the ASCII password

```
len = strlen(A);
while (len < 32)
{
    for (i = len; i < len * 2; ++i)
        pass[i] = pass[i - len] + len; // increment each character by len

    len = len * 2;
}
```

Example:  $A_0 = 0x74$  ('t')

$A_1 = 0x65$  ('e')

$A_2 = 0x73$  ('s')

$A_3 = 0x74$  ('t')

$A_4 = 0x61$  ('a')

$A_5 = 0x79$

$A_6 = 0x6A$

$A_7 = 0x78$

$A_8 = 0x79$

$A_9 = 0x66$

$A_{10} = 0x7E$

·  
·  
·

The resultant 32-byte array, A, is then passed through four rounds of a function which XOR's against a 64-byte constant (Figure 4):

```
B1 56 35 1A 9C 98 80 84 37 A7 3D 61 7F 2E E8 76
2A F2 A5 84 07 C7 EC 27 6F 7D 04 CD 52 1E CD 5B
B3 29 76 66 D9 5E 4B CA 63 72 6F D2 FD 25 E6 7B
C5 66 B3 D3 45 9A AF DA 29 86 22 6E B8 03 62 BC
```

Figure 4: 64-byte constant block for use with passwords greater than 4 characters

Let  $B_k$  be the  $k$ th byte of B, the 64-byte constant block

Let  $m = 2, 16, 24, 8$  for each of the four rounds

$\text{index} = (A_m + A_{m+1}) \& 0x3F$ ; // 6 LSB

$\text{shift} = (A_{m+2} + A_{m+3}) \& 0x7$ ; // 3 LSB

```
for (i = 0; i < 32; ++i)
{
    if (m == 32) m = 0; // wrap around to beginning
    if (index == 64) index = 0; // wrap around to beginning
```

$\text{temp} = B_{\text{index}}$ ; // xy

$\text{temp} \ll= 8$ ;

$\text{temp} \oplus= B_{\text{index}}$ ; // xyxy

```

temp >>= shift;
A_m ^= (unsigned char) temp;

++m;
++index;
}

```

The resultant 32-byte encoded password block (Figure 5) does not have any remnants of the constant block as the short length encoding method does. Although the block appears to be "random", it is indeed reversible with minimal computing resources as shown in our proof-of-concept tools.

```

18 0A 43 3A 17 7D A3 CA D7 9D 75 D2 D3 C8 A5 CF
F1 71 07 03 5A 52 4B B9 70 2D B2 D1 DF A5 54 07

```

Figure 5: Encoded password block of ASCII password 'testa'

#### Temporary Solution:

The Security application provides functionality to "turn off and lock device". If the Palm device is turned off and locked using this feature, the device will not be operational until the correct password is entered. This will prevent an unauthorized user from running applications on the device (hence preventing them from starting the HotSync process). This workaround is only useful if the legitimate user can be sure that the attacker hasn't attained the system password already - simply change the password to be sure. It may be possible to bypass the system lock-out mechanism by entering into the PalmOS debug mode before the lock-out features are called. This may allow an attacker to step over the security code during a debugging session.

The use of third-party encryption solutions, such as Secure Memopad by Certicom, which implement strong and tested cryptological algorithms to protect the data of certain Palm applications.

#### Vendor Response:

Thanks you for your diligence in testing our products thoroughly, we appreciate your efforts.

We have taken a close look at your advisory in detail and while this is certainly something we want to address for the future, we do not believe this poses a major risk to all our users for the following reasons :- It is not easy for someone to capture passwords accidentally, you need to have access to the device and access to the OS/software as well to run the hotsync and thence capture the data. It would also need to be a malicious, funded, attack and some data points need to be known to the attacker, making the chances of such an attack very low, but not impossible in everyday life. However we do appreciate the risk involved if the attacker is involved in some form of industrial espionage for example.

The simple way to protect against such an attack is to use products from Force.com to keep the device about your person, or to use any of the

security programs such as OnlyMe or SignOn to secure access, (as improvements over the supplied software security program) or data encryption programs such as Jaws Technology encryptors, Securememopad from Certicom to encrypt data, or Ntru encryption tools.

However we agree that any potential security issue needs to be taken seriously and we have investigated this problem and expect to have both a patch for older systems, and a solution for future releases of the PalmOS.

We respectfully ask you to post our response with your advisory, and thank you for contributing to the secure future of Palm devices.

Regards,

Palm Inc.

#### Proof-of-Concept Code:

Proof-of-concept tools have been written for the Windows 9x/NT and PalmOS 3.3 and greater platforms which demonstrate the simplicity of obtaining the encoded password block from the target device and the weak encoding scheme used to obfuscate the password. The PC version, "PalmCrypt", will encode and decode ASCII passwords to encoded password blocks and vice versa. The PalmOS version, "NotSync", will imitate the initial stages of the HotSync process via the IR port, retrieve the encoded password block of the target device, and decode and display the resultant ASCII password.

Source code and binaries for the proof-of-concept tools can be found at:

<http://www.atstake.com/research/advisories/2000/notsync.zip>  
<http://www.atstake.com/research/advisories/2000/palmcrypt.zip>

Successfully using NotSync requires two Palm devices: One device running the NotSync application and the other being the target device in which the password is desired.

Facing the two devices head-to-head, run the HotSync application on the target Palm device and initiate an "IR to a PC/Handheld" HotSync. NotSync, running on the other device, will obtain the legitimate user's encoded password block, decode the password, and display the result on the screen.

Typical usage and output for PalmCrypt is shown below:

<--- cut here --->

E:\>palmcrypt

PalmOS Password Codec  
kingpin@atstake.com  
@stake Research Labs  
<http://www.atstake.com/research>  
August 2000

Usage: palmcrypt [-e | d] <ASCII | password block>

E:\>palmcrypt -e test

PalmOS Password Codec  
kingpin@atstake.com  
@stake Research Labs  
<http://www.atstake.com/research>  
August 2000

0x56 0x8C 0xD2 0x3E 0x99 0x4B 0x0F 0x88 [V..>.K..]  
0x09 0x02 0x13 0x45 0x07 0x04 0x13 0x44 [...E...D]  
0x0C 0x08 0x13 0x5A 0x32 0x15 0x13 0x5D [...Z2..]]  
0xD2 0x17 0xEA 0xD3 0xB5 0xDF 0x55 0x63 [.....Uc]

E:\>palmcrypt -d  
568CD23E994B0F8809021345070413440C08135A3215135DD217EAD3B5DF  
5563

PalmOS Password Codec  
kingpin@atstake.com  
@stake Research Labs  
<http://www.atstake.com/research>  
August 2000

0x74 0x65 0x73 0x74 [test ]

<--- cut here --->

For more advisories: <http://www.atstake.com/research/advisories/>  
PGP Key: [http://www.atstake.com/research/pgp\\_key.asc](http://www.atstake.com/research/pgp_key.asc)

Copyright 2000 @stake, Inc. All rights reserved.