Hacking from the Palm of your Hand

Paul Clip

DEFCON - August 01, 2003

(Abbreviated Presentation)

@stake

Where Security & Business Intersect[SM]

Agenda

- **Goals**

- **Past**

  – Overview of the Palm Platform

  – Hacker Tools on the Palm

- **Present**

  – AUSTIN - A Palm OS Vulnerability Scanner

  – Architecture

  – Features

  – Demos

  – But wait, there's more!!!

- **Future**

  – New Features

@stake

Where Security & Business Intersect℠

Goals

- **Overview of Palm OS as a hacking platform**

- **Walkthrough of a Palm OS-based vulnerability scanner**

  - Architecture

  - Features & how they're implemented

  - Lessons learned

- **Release a new tool for Palm OS**

- **Have Fun!**

The Past

The Palm Platform

- **Old**

  - Motorola 68K processor

  - Max speed 66MHz

  - RAM 2-16MB

  - Typical resolution 160^2

  - Some color, some b/w screens

  - Serial/USB port

  - IR

  - Some expansion slots

  - PalmOS 4.x and below

- **New**

  - ARM processor

  - Max speed 150? 200? 400? MHz

  - RAM 16-32MB

  - Typical resolution 320^2

  - All color

  - USB port

  - IR

  - Expansion slots

  - PalmOS 5.x and above

Security Tools

- **Password Generators**
  http://www.freewarepalm.com/utilities/passgen.shtml
  http://www.freewarepalm.com/utilities/passphrase.shtml

- **Encryption**
  http://cryptopad.sourceforge.net/
  http://linkesoft.com/secret/

- **Password Crackers (old)**
  http://atstake.com/research/tools/password_auditing/

- **War Dialer**
  http://atstake.com/research/tools/info_gathering/

Communication Tools

- **Telnet**
  http://netpage.em.com.br/mmand/ptelnet.htm

- **SSH (v1 only)**
  http://online.offshore.com.ai/~iang/TGssh/

- **Web & Mail**
  http://www.eudora.com/internetsuite/

- **Ping**
  http://www.mergic.com/vpnDownloads.php

Communication Tools (continued)

- **FTP**
  http://lthaler.free.fr/

- **IR Tools**
  http://pamupamu.tripod.co.jp/soft/irmenu/irm.htm
  http://www.harbaum.org/till/palm/ir_ping/
  http://www.pacificneotek.com/omniProfsw.htm

Dev Tools

- **RPN Calculator**
  http://nthlab.com/

- **Longtime**
  Search on http://palmgear.com/

- **Filez**
  http://nosleep.net/

- **RsrcEdit**
  http://quartus.net/products/rsrcedit/

- **OnBoard C**
  http://onboardc.sourceforge.net/

Useful/Interesting Hardware

- **Serial/USB cable**

- **Keyboard**

- **GPS**

- **Modem**

- **Expansion slot gadgets**

- **Tilt switch**

- **IR booster**

- **Speedometer**

- **Robotics**

- **…**

The Present

Palm Vulnerability Scanner

- **Why?**

- **What?**

    - TCP & UDP scanning

    - Multiple hosts/ports

    - Banner grabbing

    - Save results in re-useable format

    - Standalone/self-contained program

- **What about other scanners?**

Choosing a Development Environment…

- C / C++

- Assembly

- CASL

- AppForge

- NS Basic

- Satellite Forms

- DB2 Personal App Builder

- Java (many flavors)

- Forth

- PocketStudio (Pascal)

- PocketC

- Smalltalk

- Perl

- Python

Even more tools at: http://www.palmos.com/dev/tools/

@stake

Where Security & Business Intersect℠

Technical Features

- **Must have**

  – Leverage Palm UI

  – Responsive

  – Extensible

  – Development on PC

- **Nice to have**

  – Development on Palm

- **Most important**

  – Re-use other components
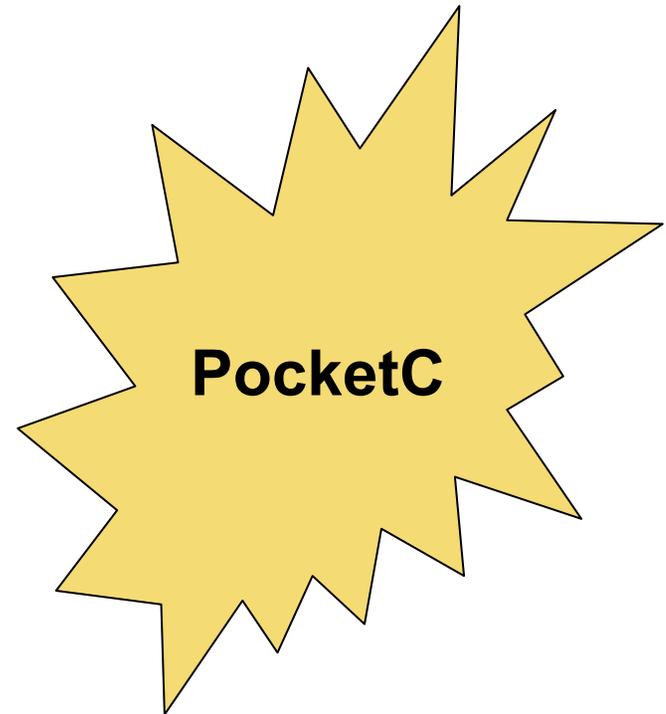
PocketC

PocketC Overview

- **Interpreted C-like language**

- **Variable types: int, float, char, string, pointer**

- **Multi-dimensional arrays**

- **Structs possible through a (minor) hack**

- **Reasonably fast**

- **Allows development on Palm + PC platforms**

- **Extensible**

Example:

```
//helloworld.pc

main()

{

    puts("Hello world!\n");

}
```

http://www.orbworks.com/pcpalm/index.html

@stake

Where Security & Business Intersect℠

Extending PocketC

- **Can be done in two ways**

  - PocketC include files

  - Native (C/C++) libraries

- **Must-have PocketC library**

  - Pocket Toolbox by Joe Stadolnik
    http://www.geocities.com/retro_01775/PToolboxLib.htm

  - Features:

    - Full access to Palm OS GUI functions

    - Database functions

    - Graphic functions

    - Much more...

Presenting… AUSTIN

- **AUSTIN stands for**

  - **A**t Stake

  - **U**ltralight

  - **S**canning

  - **T**ool (for the)

  - **I**nter-

  - **N**et

AUSTIN Architecture

AUSTIN	Scan.h	GUI.h	Prefs.h	Net.h	…

PocketC		
	Pocket Toolbox	AUSTIN NetLib

Palm OS

Palm Hardware
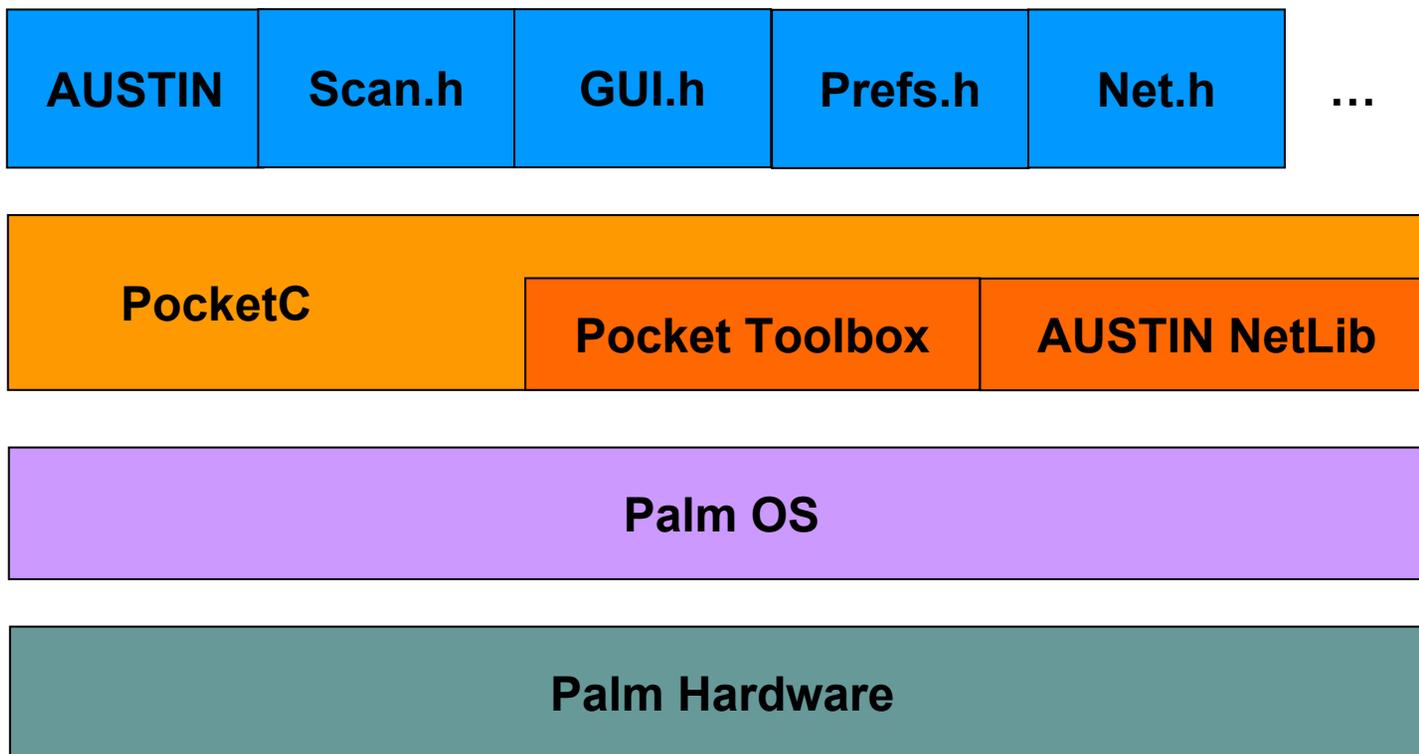
Tools Used To Develop AUSTIN

- **POSE - Palm OS Emulator**
  http://www.palmos.com/dev/tools/emulator/

- **PDE - PocketC Desktop Environment**
  http://www.orbworks.com/pcpalm/index.html

- **PRC-Tools - Includes gcc and other tools used to create Palm executables**
  http://prc-tools.sourceforge.net/

- **Palm SDK**
  http://www.palmos.com/dev/tools/sdk/

- **PiIRC**
  http://www.ardiri.com/index.php?redir=palm&cat=pilrc

@stake

Where Security & Business Intersect℠

Palm OS NetLib

- **Provides network services to Palm OS applications**

  - Stream-based communications using TCP

  - Datagram-based communications using UDP

  - Raw IP available too

- **In addition to native Palm OS function calls, NetLib also supports the Berkeley Socket API**

Lesson Learned:

Using the native NetLib calls gives you much better control over network communications, such as the ability to set timeouts.

Lesson Learned:

Close sockets as soon as you no longer need them, you only have half a dozen to play with!

Native Network Library

- **AUSTIN Net Lib implemented in C as a PocketC native library**

- **Implements the following calls**

  - `netLibInit(…)`

  - `netLibVersion(…)`

  - `netSetTimeout(…)`

  - `netGetError(…)`

  - `netLibClose(…)`

  - `netTCPConnect(…)`

  - `netSocketConnect(…)`

  - `netSocketOpen(…)`

  - `netSocketReceive(…)`

  - `netSocketSend(…)`

  - `netSocketClose(…)`

Lesson Learned:

Default timeout is 5 seconds, you may need to increase this if you're on a slow connection, see the Preferences database.

Example: netSocketSend()

```
// sends data via socket
// int netSocketSend(int socket, string data, int length,
                      int flags, pointer error)
// returns number of bytes sent

void netSocketSend(PocketCLibGlobalsPtr gP) {
   Value vSocket, vString, vLength, vFlags, vErrorPtr, *errP;
   char *buf;
   Int16 bytes;

   // get parameters
   gP->pop(vErrorPtr);
   gP->pop(vFlags);
   gP->pop(vLength);
   gP->pop(vString);
   gP->pop(vSocket);
```

Example: netSocketSend() (continued)

```
// dereference the error ptr
errP = gP->deref(vErrorPtr.iVal);

// lock string before modification
buf = (char *) MemHandleLock(vString.sVal);

// send data, capture number of bytes sent
bytes = NetLibSend(AppNetRefnum, vSocket.iVal, buf, vLength.iVal,
                   vFlags.iVal, 0, 0, gP->timeout, &(gP->error));

// cleanup
MemHandleUnlock(vString.sVal);
gP->cleanup(vString);

// return number of bytes sent, set error ptr
gP->retVal->iVal = bytes;
errP->iVal = gP->error;
}
```

HTTP HEAD with AUSTIN Net Lib & Net.h

```
//http_head.pc
library "AUSTIN_NetLib"
#include "Net.h"

main() {
   int err, port, socket, bytes;
   string result, host, toSend = "HEAD / HTTP/1.0\r\n\r\n";

   err = initNet();
   host = getsd("Connect to?", "192.168.199.129");
   port = getsd("Port?", "80");

   socket = tcpConnect(host, port);
   if (socket >= 0) {
        bytes = tcpWrite(socket, toSend);
        bytes = tcpRead(socket, &result, 200);
        puts("Received " + result);
        tcpClose(socket);
   }
   clearNet();
}
```

@stake

Where Security & Business Intersect℠

More Lessons Learned about Native Libraries

- **Read <u>all</u> the PocketC documentation on native libs (i.e. that one file in the docs/ folder :-)**

- **Make sure you have your dev environment set up correctly, i.e. all the include files and all the lib files**

- **Go to the PocketC forums and read the discussions that have mentioned native libs (some have code samples)**

- **Use AUSTIN Net Lib as a basis for your own libs (and re-use the makefile too!)**

Database Access

- **Pocket Toolbox manipulates two DB formats**

  - Pilot-DB (GPL)

  - HanDBase (Commercial)

- **Databases are used throughout AUSTIN**

  - Preferences

  - Web vulnerabilities

  - Results

Graphical User Interfaces

- ## Two ways to create GUIs on Palm OS

  - Dynamically (i.e. programmatically)

  - Resource files (i.e. using PilRC to create a resource file)

- ## Part of AUSTIN's resource file

```
FORM ID 4000 AT (0 0 160 160)

NOFRAME

MENUID 8000

BEGIN

  TITLE "AUSTIN"

  BUTTON "Scan!" ID 4201 AT (121 2 AUTO 9) FONT 1

  LABEL "Options:" AUTOID AT (0 78) FONT 0

  CHECKBOX "TCP Scan" ID 4301 AT (48 62 AUTO AUTO) FONT 0
```

Scheduled Scanning

- **AUSTIN can scan at regular intervals**

- **Users can specify**

  - Number of scans

  - Minutes between scans

  - Whether to scan or sleep first

Tying it all Together

palmos.com

RCP

Icons

Source

Creator ID

PDE

PilRC

PAR

AUSTIN

Note: AUSTIN Net Lib could also be embedded inside AUSTIN but is kept separate to facilitate reuse
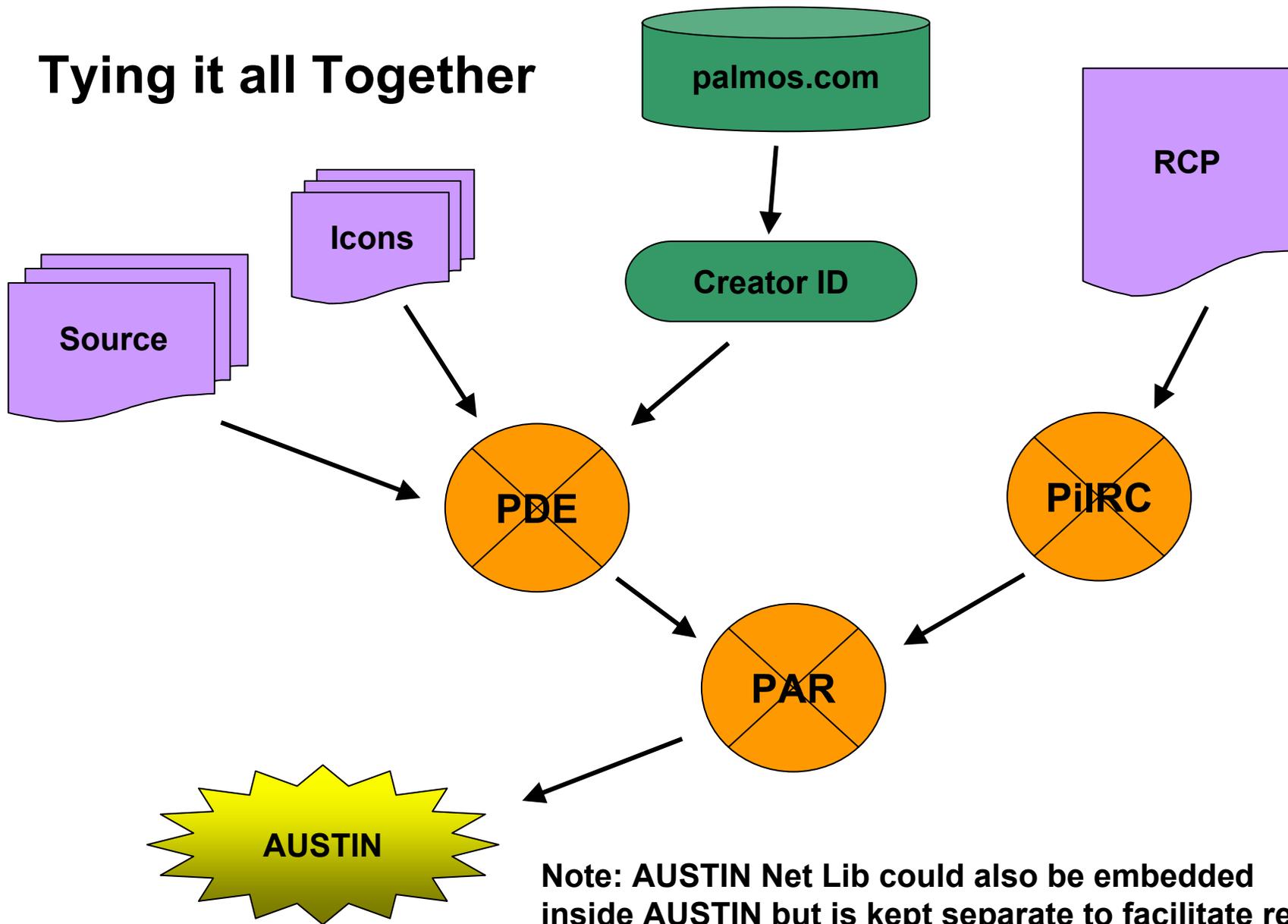
The Future

NASL Scanning

- **Idea**

  – How to leverage the work that the Nessus team has done?

- **Issues**

  – (Nearly) All tests written in NASL

  – Nessus/NASL not made to run on a Palm

  – Complexity is higher

Comparing NASL and PocketC

- **Similarities**

  - Basic C syntax

    - for and while loops

    - Control flow

    - Blocks

  - No memory management

  - Ints, chars, strings, and arrays should cover most (all?) NASL var types

- **Differences in NASL**

  - Comments (# vs. //)

  - No need to declare variables

  - Named function parameters

  - Varargs

  - The "x" operator

  - The "><" operator

  - Specific functions

More Ideas for Features

- **Creation of custom IP packets**

  – Enable SYN, FIN, XMAS scans

  – Useful for NASL functions too

- **Network tools (e.g. IP<->Hostname lookups, ping, traceroute, etc.)**

- **SSL scanning (probably wait for Palm OS 5 device)**

- **VulnXML support for URL scanning**

- **Download updates to URL vuln database**

- **Other suggestions?**

Thanks
for listening!

Any questions?

You can download AUSTIN here:
http://atstake.com/research/tools/vulnerability_scanning/