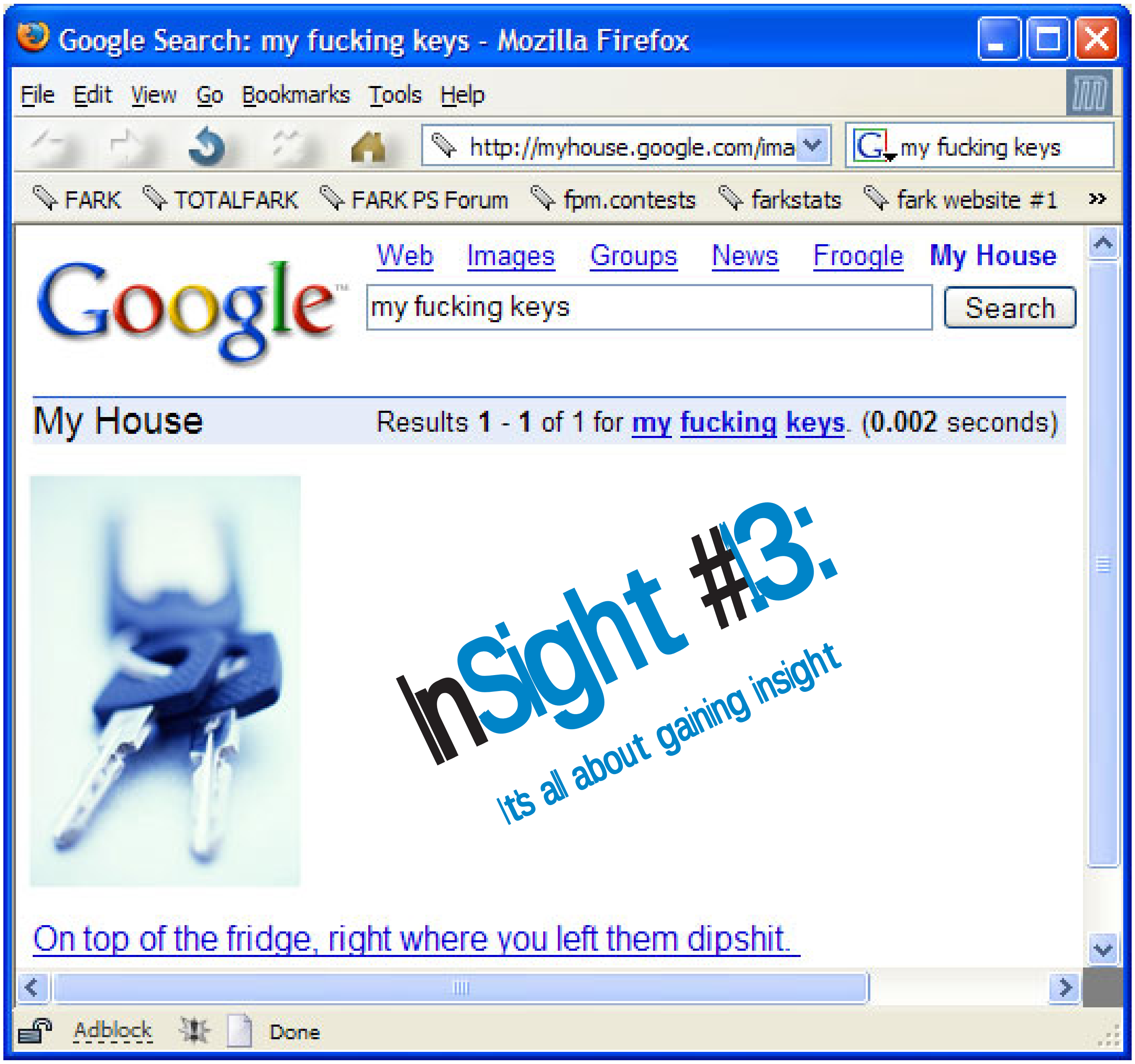
AntiOnline.com

Hackers Know The Weaknesses In your System. Shouldn't You?



Google Search: my fucking keys - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://myhouse.google.com/ima

my fucking keys

FARK TOTALFARK FARK PS Forum fpm.contests farkstats fark website #1

Web Images Groups News Froogle **My House**

my fucking keys

Search

My House Results **1** - **1** of 1 for **my fucking keys**. (**0.002** seconds)

Insight #13:

It's all about gaining insight

On top of the fridge, right where you left them dipshit.

Adblock Done

Image of Google Search sourced from http://forums.fark.com/cgi/fark/comments.pl?IDLink=930918

Table of Contents

MsMittens' Editorial

One of the things I've always liked about what goes on in the forums is our ability to find things. This is, in my opinion, still one of the better sides to the internet. And it is this ability to do research and experiment that makes many of the members stand out compared to other sites.

Certainly we have our fair share of those that supply links (moi included) or offer up the obvious ("Use Google, dammit, Jim!" — again, moi included) but it's not all that. We do have a vast amount of knowledge at AO and even more importantly, we know how to tap it when needed. Being a "know-it-all" isn't the answer to everything. Being a "know-where-to-find-it", however, can be. Particularly when you learn something as you discover that "it".

As an example, I recently ran into issues with the SO's laptop. The taskmanager didn't have all the tabs. Searches for spyware and virus/worms turned up nothing. Then a colleague at work lamented on the same issue. Hrmmm. Something else was possibly at play. Perhaps a user issue? Sure enough, a quick search at Microsoft's Knowledge Base and voila: http://support.microsoft.com/default.aspx?scid=kb;en-us;314227 . Tiny Footprint mode for Task Manager. Who knew?

To me, the person that doesn't want to learn is the one that looks for others to simply provide an answer for whatever question they have. Rather than doing the poking around, much akin to the "hackers" of old, they want the answer on a silver platter with all the footnotes.

MEH

Must be the educator in me. I bring all this up for a reason (you knew that already, didn't ya?). I suspect people wonder why I don't just take a few tutorials and fill the InSight with those. Well, one of the big things about learning is the sense of adventure and discovery that comes with it. That is, it's something new every time. Even if it's something small, it's new. This is why I prefer to fill these pages with stuff that hasn't appeared elsewhere (at least not immediately – you're welcome to post it in Tutorials or elsewhere after the newsletter is put forth). It gives reason to people wanting to learn.

I'm hoping that this revival of the InSight will mean new paths to adventures to new things we've never thought of. What of WiMax? VoIP? New attack vectors? The power of Pharming and Phishing? New modes of defense? Teaching the old dogs (employees) new tricks (edu-ma-cating 'em)?

If you'd like to submit for the next issue here are the guidelines:

1. Generally, I accept anything that is written in clear english and is original. Ideally focus it on security based topics.
2. Submit it in Rich-text format (RTF) or plain text. I end up importing this into Pagemaker before putting it into PDF format. Include any images as a seperate zip or tar.gz file. It makes it easier on me for formatting.
3. There is no limit on number of words but a minimum should be 750 (about 2-3 pages)
4. Next deadline: June 30th.

Enjoy the issue!

P.S. I'm playing with some design stuff so i fyou have ideas on layout, colours, fonts (things you like and things you didn't like) pop me a PM.

Pen-testing Tools for the Pocket PC

(or "Is that a port scanner in your pocket or are you just happy to see me?")

By Irongeek (Adrian Crenshaw)

As some of you may know I run a website with information on using the Sharp Zaurus PDA as a Pen-testing tool. Since the Zaurus runs Linux, porting over security apps meant to run on a Linux PC is pretty simple. But what about the other, more popular side of the fence, the Microsoft Pocket PC (usually abbreviated PPC)?

Unfortunately, the choice of good pen-testing tools for Pocket PC is pretty limited. Your best bet, if you want to use your PPC device as a Pen-testing tool, is to see if you can find a distribution of Linux that supports your model and install it, forgoing the PPC/Windows Mobile/WinCE.net OS entirely[0].

With those caveats stated, let's dive into what tools are out there that would be useful to the mobile pen-tester. I'll concentrate on free tools since I have no budget and abhor the idea of paying for a tool that does a worse job than an Open Source alternative running on a Linux/PC platform. I'll also be sticking to tools that are useful for pen-testing and network reconnaissance, ignoring tools for securing the PPC itself like firewalls, encryption apps and anti-virus packages (as of this writing, there seem to be more AV apps for the PPC then there are actual viruses). For my test system I'll be using a Dell Axim X5 with PPC 2003 and a Linksys WCF12 compact flash Wi-Fi card.

Installing Software

I'll gloss over the installation of PPC software; it's pretty easy. There are basically three different scenarios when installing a Pocket PC application:

1. In most cases there will be an installer that you run on your desktop PC that sets ActiveSync up so that the next time you dock your PPC the application will be installed for you automatically (you may have to tap a confirm button on the PPC itself).

2. The application may come as just the binary (an exe file) and support files which you will have to copy to your PPC using My Computer->Mobile Device->My Pocket PC, then run them using the File Explorer.

3. The third and least common way is if the app comes as a CAB file. In this case just copy the CAB file to your PPC the same way as above, then find it in the file explorer and tap it to install.

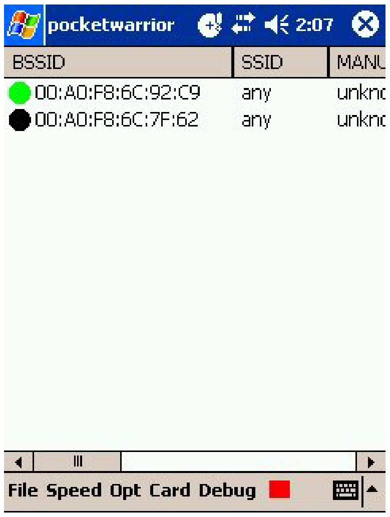
War driving (or is that walking?)

First, let's look at war driving apps for the Pocket PC. One limitation the PPC has is that there seem to be no free tools that let you put the Wi-fi card into RF Mon mode; this means you will never see cloaked SSIDs. To get some of these war driving applications to work you may have to play a little driver bingo.

For example, the drivers from Linksys for my card won't work with any of these tools. But some other older Prism2 drivers will work just fine. If you have problems getting these applications to work do a Google search on the tool and Wi-fi card you are trying to use. If you have the cash you may want to look into Airmagnet [1] since it's the only PPC tool out there that I know of that will find cloaked SSIDs. Make sure you check the supported hardware list before you buy Airmagnet since it's kind of particular about hardware. As a side note, I really wish folks on forums would stop referring to war driving tools as sniffers; it just confuses the hell out of Google searches when looking for a real network sniffer.

Here are some of the current free or Open Source PPC war driving apps:

Pocket Warrior
http://www.pocketwarrior.org/



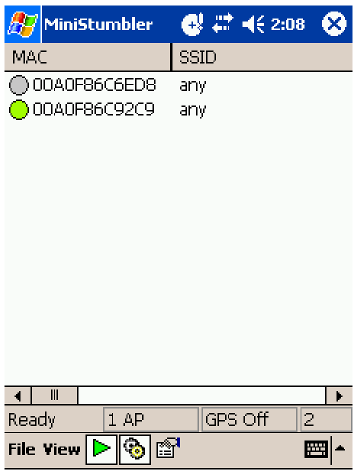
BSSID	SSID	MANU
● 00:A0:F8:6C:92:C9	any	unkno
● 00:A0:F8:6C:7F:62	any	unkno

The last version of PocketWarrior seems to have come out early in 2003, but it still works well. PocketWarrior supports a GPS and lets you save the information on the WAPs it found. All in all not bad if you don't mind missing cloaked SSIDs, but then again none of the other Wi-Fi tools I review below can see cloaked SSIDs either. PocketWarrior worked fine on my Prism2 based card as long as I used the older Senao drivers.

WiFiFoFum
www.wififofum.org

WiFiFoFum does not seem to have an installer; you just copy the files to your PPC and run the executable from File Explorer by tapping it. On my Axim X5 it just quit without giving an error message, but I've seen it in action before on an Axim X3 and worked quite well. WiFiFoFum has a radar like display that indicates how strong of a signal you're getting from a WAP, cute but it misleads some folks into thinking that the display is indicating the direction of the WAP. WiFiFoFum also support a GPS if you have Compact Framework SP2 installed.

MiniStumbler
http://www.netstumbler.com



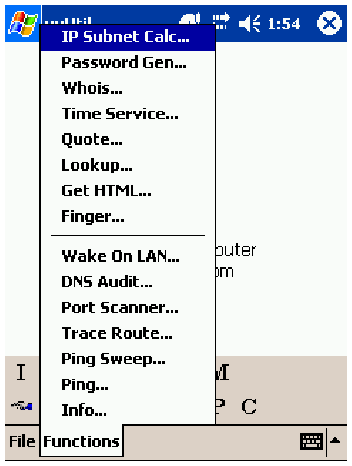
MAC	SSID
00A0F86C6ED8	any
00A0F86C92C9	any

Ready | 1 AP | GPS Off | 2

MiniStumbler is the little brother of the Windows PC tool NetStumbler. It supports quite a few Wi-Fi chip sets and the current version (0.4.0 as of this writing) worked flawlessly with my Prism2 card. It has GPS support and a very intuitive interface. If you're familar with NetStumbler for Windows then you should feel right at home with MiniStumbler. It supports 802.11a as well as 802.11b/g networks. Since MiniStumbler saves its session files in the same format as NetStumbler you should have no problem using mapping programs meant for NetStumbler or uploading your finds to Wigle.net[2].

General Network Information Tools

I'll lump general tools that allow you to find out more about the network you're on into this category. Pocket PC ships with almost nothing built in for exploring the network you're connected to, but luckily there are a few third party tools that may help a little.



vxUtil
http://www.cam.com/vxutil.html

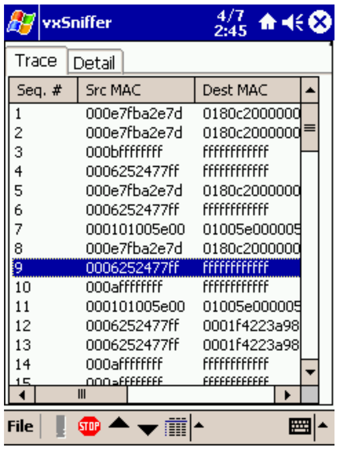
As far as free network information tools for the PPC go there's not much that can touch vxUtil. In some ways it's like SamSpade for the PPC. VxUtil

Personal is several small applications rolled into one and supports the following functions:

DNS Audit
DNS Lookup
Finger
Get HTML
Info (sort of like IP config for Windows)
IP Subnet Calculator
Password Generator
Ping
Ping Sweep
Port Scanner
Quote
Time Service
Trace Route
Wake On LAN
Whois

While most of these applications are pretty rudimentary they are quite useful and fill a spot left vacant by the tools that come with Pocket PC 2003. The port scanner is slow but it works; just don't expect all of the speed, stealth and packet options of a tool like Nmap.

vxSniffer
http://www.cam.com/vxsniffer.html



Seq. #	Src MAC	Dest MAC
1	000e7fba2e7d	0180c2000000
2	000e7fba2e7d	0180c2000000
3	000bffffffff	ffffffffffff
4	0006252477ff	ffffffffffff
5	000e7fba2e7d	0180c2000000
6	0006252477ff	ffffffffffff
7	000101005e00	01005e000005
8	000e7fba2e7d	0180c2000000
9	0006252477ff	ffffffffffff
10	000affffffff	ffffffffffff
11	000101005e00	01005e000005
12	0006252477ff	0001f4223a98
13	0006252477ff	0001f4223a98
14	000affffffff	ffffffffffff
15	000affffffff	ffffffffffff

A pretty rudimentary network sniffer for the cost of $60 bucks, but there is a 30 day evaluation version. You can save out the network captures as a text file so make sure you invest in an SD card to write large dumps too.

Airscanner Mobile Sniffer
http://www.pdagold.com/software/detail.asp?s=223
http://www.airscanner.com/

Airscanner Mobile Sniffer only supported PPC 2002 (try it on PPC 2003 and you will likely get the error "Windows CE failed to load the packet capture driver"). It's kind of hard to find now since Airscanner dropped support for it but it's still mirrored on various sites. The sniffer's interface itself is not very good, but its one cool feature is that it can dump what it sniffs into a TCPDump format file which can then be loaded into more capable sniffers like TCPDump, Ethereal, Ettercap, etc.

vxSNMP
http://www.cam.com/vxsnmp.html

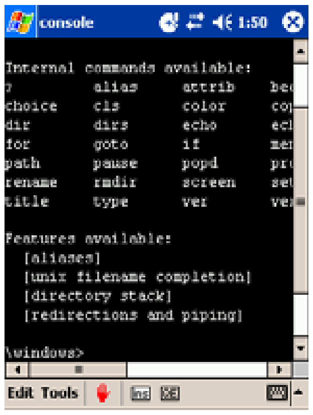
This simple tool lets you read and set SNMP values (if you know the right community names, which can be sniffed since they are passed as plain text in versions 1 and 2 of the SNMP protocol).

Tiger Tools
http://www.tigertools.net

Tiger tools claims to support all sorts of pen-testing tools, but as it does not have an evaluation version I did not test it. It claims to be able to do multi-threaded port scanning, FIN scans and run simple exploit scripts. From what I can see on their web site it looks to be written in eMbeddedVB which gives me some doubts.

PocketConsole (and related tools)
http://www.symbolictools.de/public/pocketconsole/

Pocket console makes it easier for developers to port applications that use stdout to the Pocket PC and other Windows CE devices. Here are a few of the related project (hosted on the same site as PocketConsole) that you should be aware of:



```
Internal commands available:
?          alias      attrib     be
choice     cls        color      co
dir        dirs       echo       ec
for        goto       if         me
path       pause      popd       pr
rename     rmdir      screen     se
title      type       ver        ve

Features available:
  [aliases]
  [unix filename completion]
  [directory stack]
  [redirections and piping]

\windows>
```

PocketCMD

After installing PocketConsole this is probably the next app to setup. It works in a similar fashion as the Windows command prompt and is needed to run some of the apps listed below.

NetTools (Ping, Ipconfig, Route, Net, Netstat)

A few basic network tools Microsoft left out of Pocket PC. They may not be as full featured as their Windows XP cousins but they are still useful. The function of Ping and Ipconfig are obvious. Route lets you set up IP routing information (I'm not sure how useful this is since I never plan to use my PPC as a gateway device, but it's still cool that someone spent the time to figure it out). Net allows you view SMB shares and map share points. Netstat gives you various network statistics.

SNMPUtils

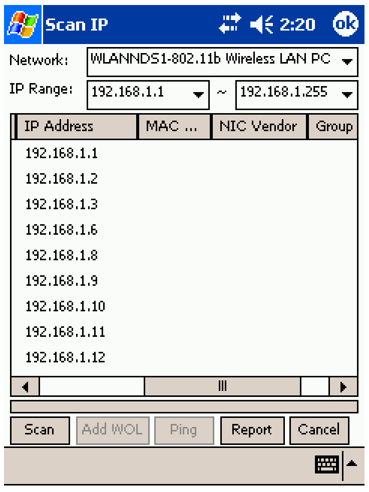
Allows you to retrieve and set SMNP values.

Telnet

Not as pretty as PocketPuTTY (mentioned below) but since you can specify ports it's more useful for doing banner grabs. For example, if you want to do a banner grab to find out what version of SSH a box is running you could use a command something like the following: open some.server.com 22

If you are a developer you may want to look into using PocketConsole when porting over your Windows console apps.

PocketLAN
http://www.pocketgear.com/software_detail.asp?id=2825



Network: WLANNDS1-802.11b Wireless LAN PC
IP Range: 192.168.1.1 ~ 192.168.1.255

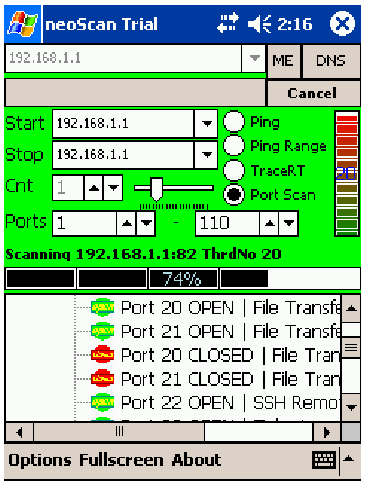
IP Address	MAC ...	NIC Vendor	Group
192.168.1.1			
192.168.1.2			
192.168.1.3			
192.168.1.6			
192.168.1.8			
192.168.1.9			
192.168.1.10			
192.168.1.11			
192.168.1.12			

PocketLAN costs $14.99 and seems like a nice tool for mapping share points to SMB file servers and finding out what machines are around you. I like the network scan function that does a quick ping sweep and reverse DNS lookup, then tells you information like the network card vendor (based on MAC address) and Domain/Workgroup the hosts belong to. You can view a report of the hosts it finds in HTML format, then copy the reports off of the PPC for later viewing (Quick tip: dock your PPC and look in My Computer->Mobile Device->My Pocket PC\\Program Files\Z2\PocketLAN to find the report).

V-Mobile Network Browser
http://www.pocketgear.com/software_detail.asp?id=14818

VM Network Browser costs $17.95 and does the same basic things as PocketLAN, but looks more like the classic Network Neighborhood interface. Unlike PocketLAN, VM Net Browser does not seem to do a ping sweep, but instead pulls its information from NetBIOS traffic or the Windows Browsemaster on the network (it's hard to tell without talking to the developers). VM Net Browser is not as responsive as PocketLAN and it doesn't give as much information about the hosts it scans.

NeoScan
http://www.dotnetux.net/



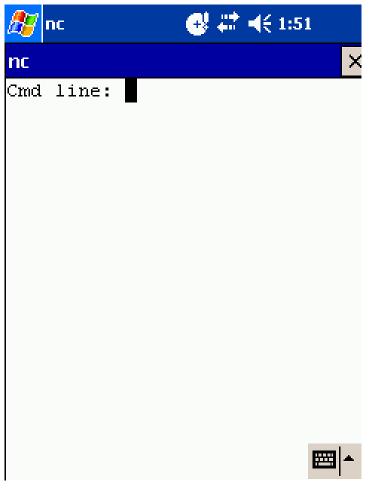
```
192.168.1.1
Start 192.168.1.1
Stop  192.168.1.1
Cnt   1
Ports 1  -  110
Scanning 192.168.1.1:82 ThrdNo 20
74%
Port 20 OPEN | File Transfe
Port 21 OPEN | File Transfe
Port 20 CLOSED | File Tran
Port 21 CLOSED | File Tran
Port 22 OPEN | SSH Remo
```

I don't know about paying $15 for a port scanner but of the ones I've tested it seems the fastest. Luckly there's a demo version. Just make sure you set it to only port scan hosts that respond to a ping, otherwise you will be waiting awhile. The bad side: I see no way to save your scans for later viewing.

NbtstatCE
http://sourceforge.net/projects/nbtstatce/

NbtstatCE does ping sweeps and is supposed to retrieve NetBios info. It appears to have no way to save the scan. Nice, not as slick as other tools but hell, it's open source. As of right now I can't seem to get it to actually pull NetBIOS info, but keep an eye on this app since it shows promise.

Netcat
http://prt.fernuni-hagen.de/~bischoff/wince/#netcat



```
Cmd line:
```

Yep, there's a version of Netcat, the network Swiss army knife, for Windows CE. Netcat can be a bit clumsy to use but it's very versatile. With it you can shovel shells, port scan, do banner grabs and a host of other things. See the following website for many of the possible uses for Netcat:

http://www.giac.org/certified_professionals/practicals/gsec/0436.php

If Netcat loses the connection before you can see the output check the files nc-stdin.txt, nc-stdout.txt and nc-stderr.txt located in the same directory as the Netcat executable. One bug with Netcat for Windows CE is that the backspace key does not work, so be carefull when you type in a command. To give you one example of usage, here is how you could use Netcat to do a quick banner grab to find out what version of sshd a host is running:

1. Start Netcat by tapping on nc.exe in the File Explorer.

2. Issue the command (replace "targethost" with the name or IP of the host you are connecting to):

```
targethost 22
```

3. Hit the enter key and open nc-stdout.txt if Netcat closes before you can read the output.

Clients

While not technically pen-testing tools, every pen-tester needs various clients to access the services they are targeting. Here's a short list of clients I find quite useful on the PPC platform.

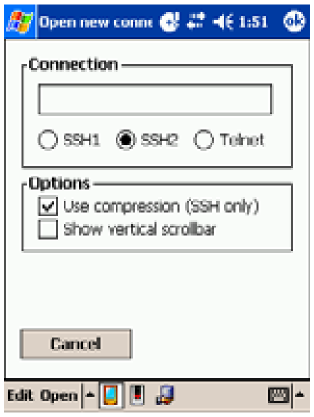
Terminal Services Client

Not much to say here. Terminal Service Client comes with Pocket PC and it works fine if you can tolerate the small screen size and a lot of scrolling. Damn useful for connecting to a Windows Terminal Server or an XP box running Remote Desktop.

VNC Client
http://www.cs.utah.edu/~midgley/wince/vnc.html
or
http://sourceforge.net/projects/dotnetvnc/

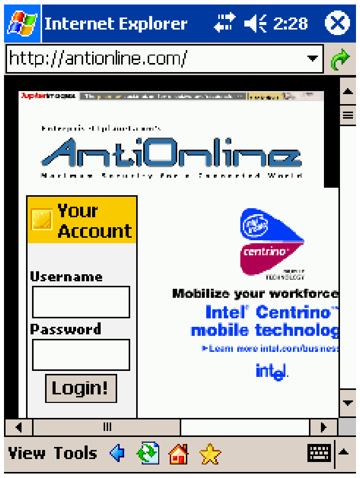
While both of these VNC clients work, they seem slower than my grandma at the grocery. I think the slowness has something to do with my VPN or just a limitation of Windows CE networking (the VNC client on my Zaurus runs plenty fast). Neither has an installation wizard so just copy them to your PPC and run them using File Explorer. The cool thing about the .Net VNC client is that you can use the same executable on both your PPC and your Windows PC.

PocketPuTTY
http://pocketputty.duxy.net/



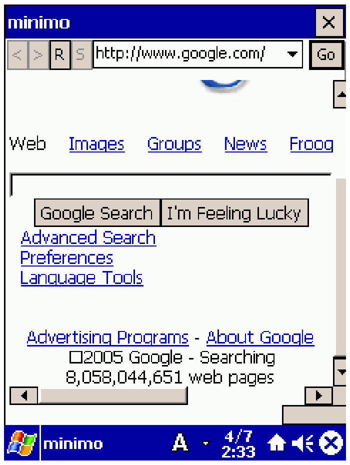
PocketPuTTY is the Pocket PC port of the popular Putty SSH client (now try to say that out loud). PocketPuTTY is pretty much your best option for connecting to your *nix box from the PPC. PocketPuTTY does not come with an installation wizard so just copy the files to your PPC. There are two different versions out on their site as of this writing. Make sure you get v0.1-prealpha-0.53b if you're running PPC2003 and newer or download v0.2-alpha-2k2-0.53b if you're using PPC 2002. One huge downside to this app is that I see no place to set it to connect on a non-standard port, so if you want to try banner grabbing use Netcat or the PocketConsole version of Telnet mentioned above.

Pocket IE (PIE)



Pocket IE comes bundled with PPC. The version that ships with PPC 2003 and later shows many speed improvements over the older one in 2002. Till Minimo is ready for prime time I'd recommend sticking to Pocket IE. Keep in mind that Microsoft does not keep PIE all that up to date, so you may want to make yourself aware of some of its vulnerabilities [3].

Minimo
http://www.meer.net/~dougt/minimo_ce/MinimoCE_0.002.zip
http://www.mozilla.org/projects/minimo/


minimo
http://www.google.com/
Go
Web Images Groups News Froog
Google Search I'm Feeling Lucky
Advanced Search
Preferences
Language Tools
Advertising Programs - About Google
©2005 Google - Searching 8,058,044,651 web pages
minimo 4/7 2:33

Minimo is a version of Mozilla meant for PDA size devices. They are working on porting it to Pocket PC (see a link to the early development version above). While it does not seem to be fully functional yet keep an eye on it in the future. By the way, just so you know, I did not mess up the screen shot to the right -- it really looked like that when I ran it.

neoFTP
http://www.dotnetux.net/

A simple little freeware FTP client for those that need one.

Closing

That about sums it up for the PPC tools I've found useful. Sadly, Open Source developers have by and large ignored the Pocket PC platform in favor of other more open environments. If you know of any good pen-testing tools for the Pocket PC platform please email me and I'll update my list. By the way, if you don't have a Pocket PC yourself port scan your network for 999/tcp and you may find a co-worker that does.

[0] See if you PPC device supports Linux: http://www.handhelds.org/[1] Airmagnet Homepage http://www.airmagnet.com/
[2] Wigle, my favorite hot spot finder and mapper http://www.wigle.net/
[3] Information about Pocket PC security http://www.cewindows.net/security.htm

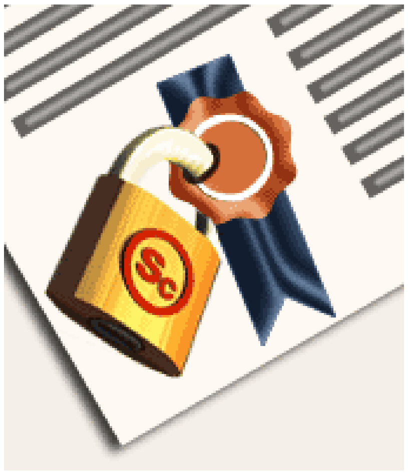
Further Research:
Ports of various *nix tools:
http://www.rainer-keuchel.de/software.html

Various freeware sites for the Pocket PC:
http://www.freewareppc.com
http://www.pocketpcfreewares.com/en/index.php
http://www.ppc4all.comhttp://ppc.palmopensource.com/?category=14

If you are having problems getting PocketWarrior or MiniStumbler working on your Intersil Prism2 base Wi-Fi card check out this thread:
http://discussion.brighthand.com/showthread.php?s=&threadid=86559

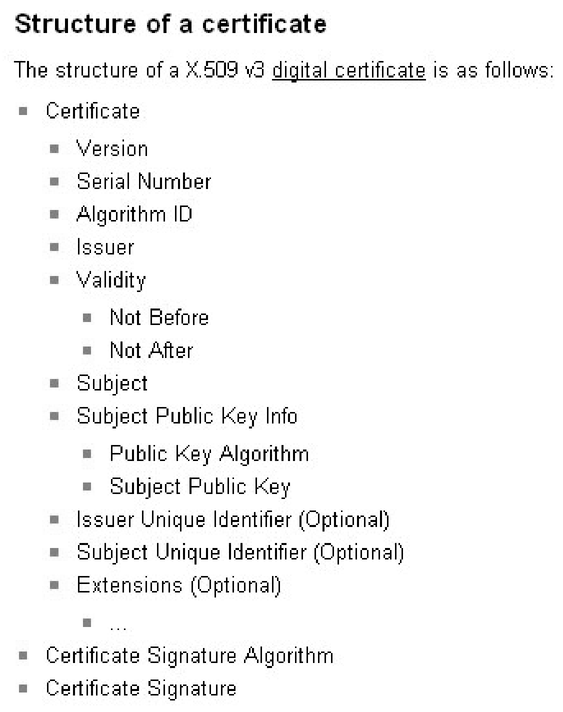
CELIB, WinCE ANSI C/POSIX library which may help you port over *nix apps:
http://sourceforge.net/projects/celib/

Security Certificates by HTRegz



The Purpose of this paper is to introduce the reader to the topic of Security Certificates. Those two words could have several broad meanings. After debating which branch of the tree I wanted to follow, I decided that I would focus on Digital Security Certificates, which is how most people interpreted those two words. I will start with what a digital certificate is, the details behind them and how you would obtain one. Following that we will look into the math behind the PKE (Public Key Encryption) of the certificate. The final section of this paper will focus in on how exactly digital security certificates provide both confidentiality and integrity and how they are indispensable to the CIA Triangle. One of the key certificates this paper will cover is X.509, the most widely used digital security certificate1. Look to the Appendices of this paper to locate both a glossary and the source code for a simple Turing program. This program is coded with specific limits due to the language; however, it will demonstrate how the math behind the generation of keys for PKE works.

What is a digital certificate? A digital certificate is one of the three required pieces to form the Public-key Infrastructure (PKI). The other items that make up PKI are public-key cryptography and certificate authorities (CA). PKI utilizes the digital certificate to provide two things confidentiality and integrity. Let's take a look at the digital certificate and then we'll attempt to figure out how it can provide us with both confidentiality and integrity. Inside of a digital certificate you'll find several items including the company name, a serial number, an expiration date and the company's public key.



Structure of a certificate

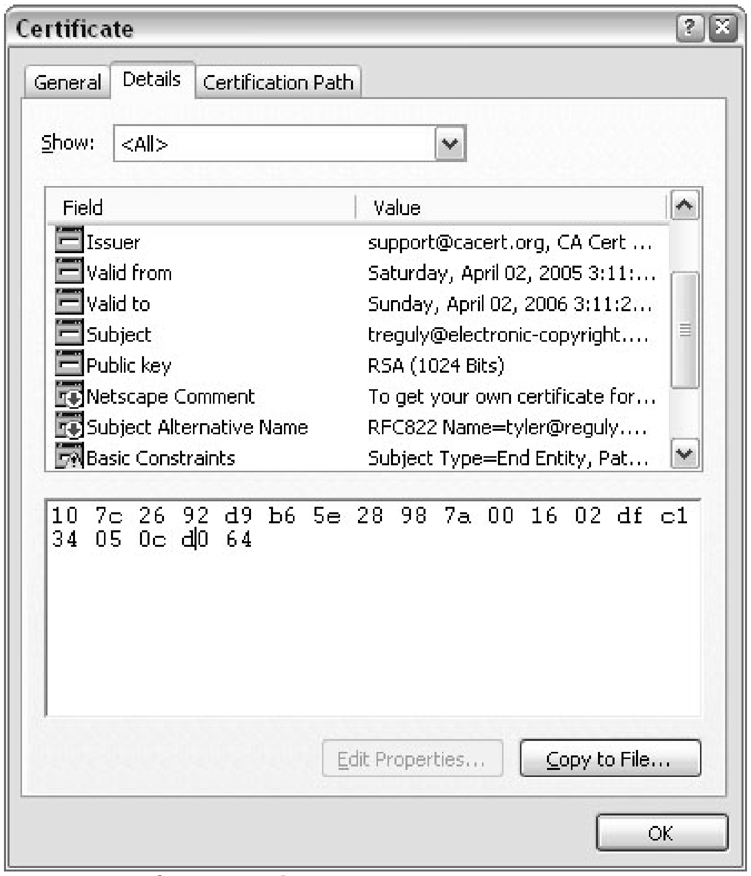
The structure of a X.509 v3 digital certificate is as follows:

- Certificate
  - Version
  - Serial Number
  - Algorithm ID
  - Issuer
  - Validity
    - Not Before
    - Not After
  - Subject
  - Subject Public Key Info
    - Public Key Algorithm
    - Subject Public Key
  - Issuer Unique Identifier (Optional)
  - Subject Unique Identifier (Optional)
  - Extensions (Optional)
    - ...
- Certificate Signature Algorithm
- Certificate Signature

Figure 1 – Structure of a X.509 Certificate

Figure 1 presents an example of the contents of an X.509 certificate. There are two ways to obtain a certificate. The first, which we will continue with throughout this document, is a CA (part of the PKI). The other option is through what is called the ‘web of trust scheme’. In a ‘web of trust’ an individual signs/endorses the certificate and indicates that the owner of the certificate is who they say they are.

There are several certificate authorities that issue digital certificates, some of which include VeriSign[3], Thawte[4] and CACert[5]. While VeriSign and Thawte charge for their digital certificates, CACert provides them free of charge. I decided to sign up with CACert so that I could demonstrate what a digital certificate actually looks like. Figure 2 shows what a certificate looks like installed in Microsoft® Windows® XP Professional and Figure 3 displays the contents of the certificate which I received.



Field	Value
Issuer	support@cacert.org, CA Cert ...
Valid from	Saturday, April 02, 2005 3:11:...
Valid to	Sunday, April 02, 2006 3:11:2...
Subject	treguly@electronic-copyright....
Public key	RSA (1024 Bits)
Netscape Comment	To get your own certificate for...
Subject Alternative Name	RFC822 Name=tyler@reguly....
Basic Constraints	Subject Type=End Entity, Pat...

```
10 7c 26 92 d9 b6 5e 28 98 7a 00 16 02 df c1
34 05 0c d0 64
```

Figure 2 – Certificate in Windows XP

Now that we’ve covered what a digital certificate is, it’s time to move on something a little more interesting. What can a digital certificate do for you and how does it do that.

As was previously mentioned there are two things that a digital certificate can provide. They are confidentiality and integrity. How does it do this? A digital certificate contains a public key. This key is one half of the required elements for asymmetric encryption. The other element is a private key. This one is obviously not stored in our digital certificate that we are distributing. The big question here is ‘How do we create these two keys so that they function together mathematically?’ The answer is actually quite interesting. The mathematics behind it is as follows6:

1. Find *P* and *Q*, two large (e.g., 1024-bit) prime numbers.

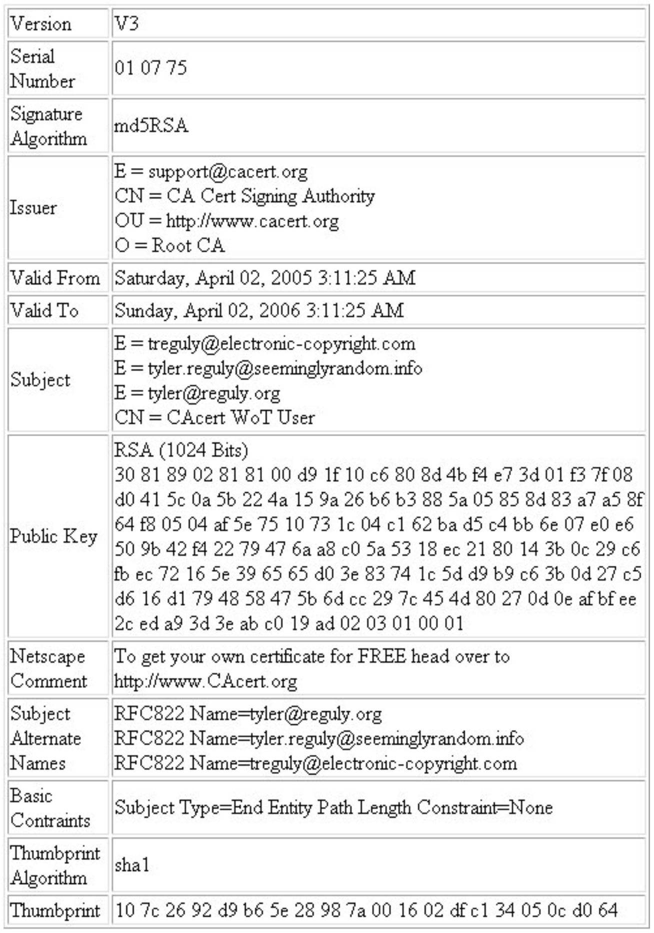
2. Choose *E* such that *E* is greater than 1, *E* is less than *PQ*, and *E* and *(P-1)(Q-1)* are *relatively prime*, which means they have no prime factors in common. *E* does not have to be prime, but it must be odd. *(P-1)(Q-1)* can’t be prime because it’s an even number.

3. Compute *D* such that *(DE - 1)* is evenly divisible by *(P-1)(Q-1)*. Mathematicians write this as *DE = 1 (mod (P-1)(Q-1))*, and they call *D* the *multiplicative inverse* of *E*. This is easy to do — simply find an integer

X which causes *D = (X(P-1)(Q-1) + 1)/E* to be an integer, then use that value of *D*.

4. The encryption function is *C = (T^E) mod PQ*, where *C* is the ciphertext (a positive integer), *T* is the plaintext (a positive integer), and ^ indicates exponentiation. The message being encrypted, *T*, must be less than the modulus, *PQ*.

5. The decryption function is *T = (C^D) mod PQ*, where *C* is the ciphertext (a positive integer), *T* is the plaintext (a positive integer), and ^ indicates exponentiation.



Version	V3
Serial Number	01 07 75
Signature Algorithm	md5RSA

Issuer	E = support@cacert.org
	CN = CA Cert Signing Authority
	OU = http://www.cacert.org
	O = Root CA

Valid From	Saturday, April 02, 2005 3:11:25 AM
Valid To	Sunday, April 02, 2006 3:11:25 AM

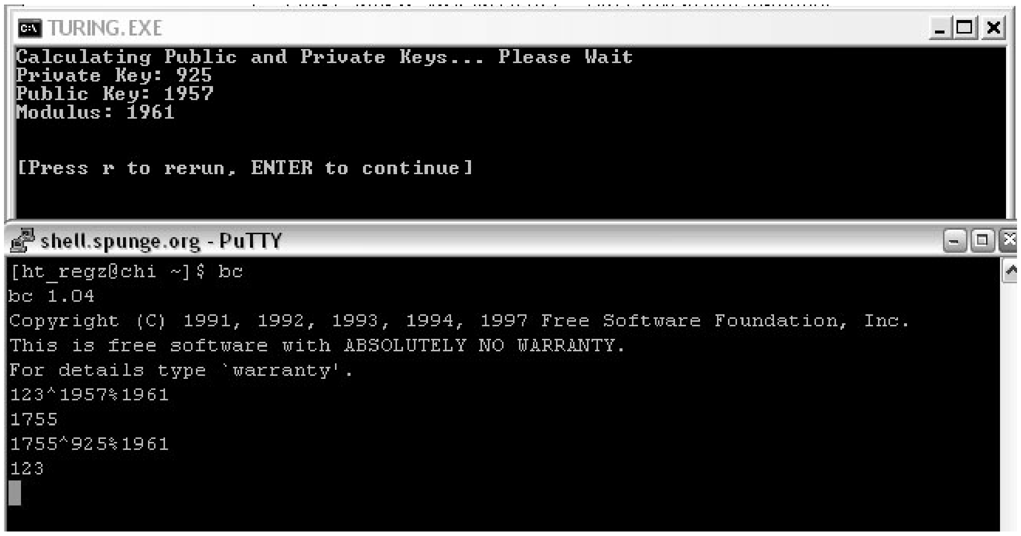
Subject	E = treguly@electronic-copyright.com
	E = tyler.reguly@seeminglyrandom.info
	E = tyler@reguly.org
	CN = CAcert WoT User

Public Key	RSA (1024 Bits) 30 81 89 02 81 81 00 d9 1f 10 c6 80 8d 4b f4 e7 3d 01 f3 7f 08 d0 41 5c 0a 5b 22 4a 15 9a 26 b6 b3 88 5a 05 85 8d 83 a7 a5 8f 64 f8 05 04 af 5e 75 10 73 1c 04 c1 62 ba d5 c4 bb 6e 07 e0 e6 50 9b 42 f4 22 79 47 6a a8 c0 5a 53 18 ec 21 80 14 3b 0c 29 c6 fb ec 72 16 5e 39 65 65 d0 3e 83 74 1c 5d d9 b9 c6 3b 0d 27 c5 d6 16 d1 79 48 58 47 5b 6d cc 29 7c 45 4d 80 27 0d 0e af bf ee 2c ed a9 3d 3e ab c0 19 ad 02 03 01 00 01
Netscape Comment	To get your own certificate for FREE head over to http://www.CAcert.org
Subject Alternate Names	RFC822 Name=tyler@reguly.org RFC822 Name=tyler.reguly@seeminglyrandom.info RFC822 Name=treguly@electronic-copyright.com
Basic Contraints	Subject Type=End Entity Path Length Constraint=None
Thumbprint Algorithm	sha1
Thumbprint	10 7c 26 92 d9 b6 5e 28 98 7a 00 16 02 df c1 34 05 0c d0 64

Figure 3 – Certificate Created by CACert.org

This can be demonstrated using software I wrote for this paper (Appendix 2). Using the numbers it provides we can quite easily demonstrate this algorithm. As you saw in the certificate presented above the key was 1024 bits long and as this algorithm states you take two large prime numbers (larger than 1024 bits). Because of the language which I used, I didn't have the power to utilize numbers that large, so I imposed limits that each of the selected prime numbers must be below 600. The numbers I chose for my example were 37 and 53. Beyond that all the work is done by the program. It generates E based on the criteria in step 2 above. Then it uses the equation from step 3 and runs through lists of numbers until the equation evaluates true. We now have all the values we require and they are outputted to the screen. To demonstrate the encrypt and decrypt equations, I'll utilize the Linux program bc which can do math with extremely large numbers. Figure 4 demonstrates the outcome of the encryption when we use the simple sequence 123.

Now we've explained the encryption process, how exactly does a Digital Security Certificate provide both integrity and confidentiality? The encryption information above demonstrates how this works, at least from the confidentiality aspect. If I were to encrypt a document with the public key found in your digital certificate, depending on the key length, it would, using current technology, be next to impossible to crack. This provides for a great level of confidentiality. The other aspect is integrity. If I were to use my private key to digitally sign the document. I would generate a message digest… a hash through some algorithm such as MD5 or SHA-1. I would then use my private key to encrypt that hash and then attach the encrypted hash to my



```
TURING.EXE
Calculating Public and Private Keys... Please Wait
Private Key: 925
Public Key: 1957
Modulus: 1961

[Press r to rerun, ENTER to continue]
```

```
shell.spunge.org - PuTTY
[ht_regz@chi ~]$ bc
bc 1.04
Copyright (C) 1991, 1992, 1993, 1994, 1997 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
123^1957%1961
1755
1755^925%1961
123
```

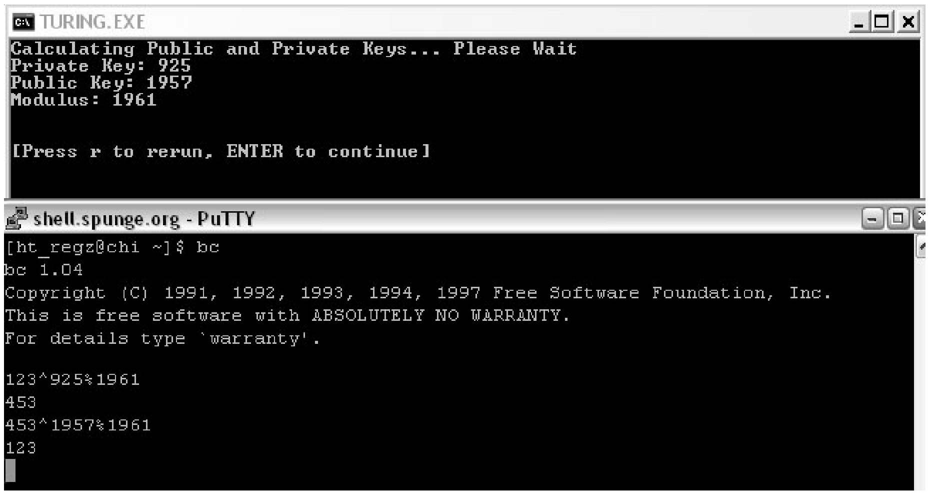
Figure 4 – Simple Encryption/Decryption Sequence

message. The recipient will then hash the message himself using the same hashing algorithm and at the same time decrypt the hash that I have sent using my public key. If the encrypted hash that I sent, matches the hash that he has generated. The integrity of the message has been maintained and confirmed.

While many people claim that PKI and therefore digital certificates are one way encryption7… it is not true. You can encrypt with either key as long as you decrypt with the opposite key. See Figure 5 for the reverse of Figure 4.

This process is almost never used because if you send something with your private key, anyone with your public key could decrypt the message. However, I could see beneficial uses of this process.

In a large financial institution with several connected networks, why not use 1024-bit or higher asymmetrical encryption rather than the lower symmetrical encryption. The people in office are allowed to see the contents, so you distribute public and private keys through out the office. Assign private keys to



```
TURING.EXE
Calculating Public and Private Keys... Please Wait
Private Key: 925
Public Key: 1957
Modulus: 1961

[Press r to rerun, ENTER to continue]
```

```
shell.spunge.org - PuTTY
[ht_regz@chi ~]$ bc
bc 1.04
Copyright (C) 1991, 1992, 1993, 1994, 1997 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.

123^925%1961
453
453^1957%1961
123
```

Figure 5 – Reversed Encryption/Decryption Sequence

areas that will receive eyes-only messages and public keys to everywhere else. Those with the public keys can encrypt content and send it to the eyes-only offices; however those with private keys can use them to send messages through-out the office that will not be readable to outside attackers. It is not the greatest use, but it is a theoretical use. However, this is why the private key encryption – public key decryption sequence is used primarily for integrity.

To sum up, digital certificates are used with communications media, most predominantly the World Wide Web and email. They provide confidentiality and integrity which are both integral to the world of security; however they do rely on a trusted third party. For this reason you should always verify the digital certificates you are dealing with. There's not much left to say about these wonderful little certificates. They provide you with security and verification and help you identify who you are dealing with.

Glossary – Appendix I

Certificate Authority (CA) – A party which provides digital certificates to individuals and corporations, they are considered to be a trusted third party.

Confidentiality – One of the three corners of the CIA Triangle. Confidentiality means who those who are supposed to see the body of the message have access to the body of the message.

Cryptography – Originates from the Greek language meaning 'Hidden Writing' or 'Secret Writing'. It is a way to hide the body of a message so that it cannot be viewed without knowing the proper key.

Digital Certificate – One of the parts of the PKI. It contains the owner's public key, along with information on both the owner and the CA which generated the certificate.

Integrity – One of the three corners of the CIA Triangle. Integrity means that only those who are supposed to have access to modify or change the body of the message have that access.

Message Digest 5 (MD5) – A 128-bit one way hash function that creates a message digest based on the body of any text.

Pretty Good Privacy (PGP) – A software package which uses digital certificates to encrypt and digital sign email messages and other forms of data.

Public Key Encryption (PKE) – A method of asymmetric encryption which has two separate keys generated based on a mathematical algorithm. One of these keys is the public key and made widely available. The other is a private key which is kept secret by the party involved. Digital Certificates include a public key.

Public Key Infrastructure (PKI) – A method through which users can utilize unsecured public networks to privately exchange data, it includes a CA, PKE, and a Digital Certificate.

Secure Hash Algorithm (SHA-1) – A 160-bit one way hash function that creates a message digest based on the body of any text. There are now more advanced versions of this including SHA-256 and SHA-512 (they are 256 and 512-bit one way hash functions).

Web of Trust Scheme – A method which provides an alternate to using CA's. This is usually found in Open Source solutions such as PGP. Instead of being endorsed by a trusted and recognized third party another user endorses you and vouches for your identity.

X.509 – The current industry standard for Digital Certificates. More information can be obtained from http://en.wikipedia.org/wiki/X.509.

Source Code – Appendix 2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
Simple PKE Demo Program
%
%
Written By: Tyler Reguly & Devon Bridge
%
%
Date: Sunday April 3rd, 2005
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Tells the program to randomize the generation of random numbers.
randomize

% Function to test if numbers are prime.
function prime (a : int) : int
    var success : int := 0
    if a = 2 or a = 3 or a = 5 or a = 7 or a = 11 or a = 13 or a =
17 or a =
         19 or a = 23 or a = 29 or a = 31 or a = 37 or a = 41 or a
= 43
         or a
         = 47 then
     success := 1
     result 1
    end if
    if a = 53 or a = 59 or a = 61 or a = 67 or a = 71 or a = 73 or
a = 78 or
         a = 83 or a = 89 or a = 97 or a = 101 or a = 103 then
     success := 1
     result 1
    end if
    if a mod 2 not= 0 and a mod 3 not= 0 and a mod 5 not= 0 and a
mod 7 not=
         0 then
     if a mod 11 not= 0 and a mod 13 not= 0 and a mod 17 not= 0 and
a mod
          19 not= 0 then
         if a mod 23 not= 0 and a mod 29 not= 0 and a mod 31 not= 0
and a
              mod 37 not= 0 then
          if a mod 41 not= 0 and a mod 43 not= 0 and a mod 47 not= 0
               and a mod 53 not= 0 then
              if a mod 59 not= 0 and a mod 61 not= 0 and a mod 67
not=
                  0 and a mod 71 not= 0 then
              if a mod 73 not= 0 and a mod 79 not= 0 and a mod 83
                   not= 0 and a mod 89 not= 0 then
                  if a mod 97 not= 0 and a mod 101 not= 0 and a
                       mod 103 not= 0 then
                   success := 1
                   result 1
                  end if
              end if
             end if
         end if
        end if
     end if
    end if
    if success = 0 then
     result 0
    end if
end prime

% Function to determine if two numbers are relatively prime.

function relative (a : int, b : int) : int
    if a mod 2 = 0 and b mod 2 = 0 then
     result 0
    elsif a mod 3 = 0 and b mod 3 = 0 then
     result 0
    elsif a mod 5 = 0 and b mod 5 = 0 then
     result 0
    elsif a mod 7 = 0 and b mod 7 = 0 then
     result 0
    elsif a mod 11 = 0 and b mod 11 = 0 then
     result 0
    elsif a mod 13 = 0 and b mod 13 = 0 then
     result 0
    elsif a mod 17 = 0 and b mod 17 = 0 then
     result 0
    elsif a mod 19 = 0 and b mod 19 = 0 then
     result 0
    elsif a mod 23 = 0 and b mod 23 = 0 then
     result 0
    elsif a mod 29 = 0 and b mod 29 = 0 then
     result 0
    elsif a mod 31 = 0 and b mod 31 = 0 then
     result 0
    elsif a mod 37 = 0 and b mod 37 = 0 then
     result 0
    elsif a mod 41 = 0 and b mod 41 = 0 then
```

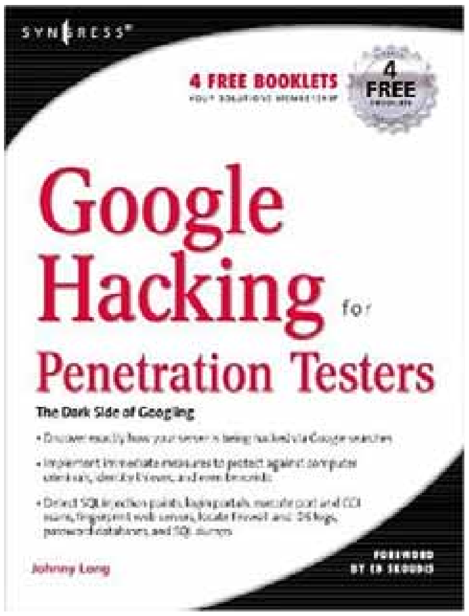
```
     result 0
    elsif a mod 43 = 0 and b mod 43 = 0 then
     result 0
    elsif a mod 47 = 0 and b mod 47 = 0 then
     result 0
    elsif a mod 53 = 0 and b mod 53 = 0 then
     result 0
    elsif a mod 59 = 0 and b mod 59 = 0 then
     result 0
    elsif a mod 61 = 0 and b mod 61 = 0 then
     result 0
    elsif a mod 67 = 0 and b mod 67 = 0 then
     result 0
    elsif a mod 71 = 0 and b mod 71 = 0 then
     result 0
    elsif a mod 73 = 0 and b mod 73 = 0 then
     result 0
    elsif a mod 79 = 0 and b mod 79 = 0 then
     result 0
    elsif a mod 83 = 0 and b mod 83 = 0 then
     result 0
    elsif a mod 89 = 0 and b mod 89 = 0 then
     result 0
   elsif a mod 97 = 0 and b mod 97 = 0 then
     result 0
    elsif a mod 101 = 0 and b mod 101 = 0 then
     result 0
    elsif a mod 103 = 0 and b mod 103 = 0 then
     result 0
    else
     result 1
    end if
end relative

% Declaration of Variables
var p, q, pq, pq1, e, d1, z : int

% Prompt the user for the first prime number and test it against
various criteria
loop
    locate (1, 1)
    put “
“
    locate (1, 1)
    put “Enter a prime number: “ ..
    get p
    if p < 10 then
```

```
     locate (1, 1)
     put “Must be greater than 10!”
     delay (1000)
    elsif p > 600 then
     locate (1, 1)
     put “Must be less than 600!”
     delay (1000)
    elsif prime (p) = 1 then
     exit
    else
     locate (1, 1)
     put “Not a prime number!”
     delay (1000)
    end if
end loop

% Prompt the user for the second prime number and test it aginst
various criteria
loop
    locate (2, 1)
    put “                                                      “
    locate (2, 1)
    put “Enter another prime number: “ ..
    get q
    if q = p then
     locate (2, 1)
     put “The two numbers cannot be the same!”
     delay (1000)
    elsif q < 10 then
     locate (2, 1)
     put “Must be greater than 10!”
     delay (1000)
    elsif q > 600 then
     locate (2, 1)
     put “Must be less than 600!”
     delay (1000)
    elsif prime (q) = 1 then
     exit
    else
     locate (2, 1)
     put “Not a prime number!”
     delay (1000)
    end if
end loop

% Generate Modulus (PQ) and Test Mechanism for Generation of E
(Public Key)

pq := p * q
pq1 := (p - 1) * (q - 1)
```

The Last Word

“The Last Word” is a new section I wanted to introduce to InSight. This can be your place to put in your words about nearly anything related to computers, networks and security. It can be a discussion on a tutorial, an article you read or whatever. Sorta think of it as the place to rant and rave. Since I’m the editor, I get first crack at it.

One of the things I’m good at is finding information. My favourity tool has been Google. This isn’t surprising. It’s a rather powerful tool that many of us truly under-utilize. We put in simple searches with or without quotes and that’s about it. Rarely do we use many of the documented (and some undocumented) features to find information.


SYNGRESS
4 FREE BOOKLETS
4 FREE
Google Hacking for Penetration Testers
The Dark Side of Googling
Johnny Long
FOREWORD BY ED SKOUDIS

And I’m not just talking about the information about what’s wrong with the computer. How about what’s wrong with your network security? Or in my case, what professor was foolish enough to use their website as a method of transporting their electronic copy of their exam for that semester? (My Google search to find these: **inurl:school.url *.doc exam**). Rather scary.

But what if you could use Google to do some pen-testing? Wouldn’t that help somewhat in determining exactly how much of an information leakage your network truly is? I stumbled across **Google Hacking for Penetration Testers** by Johnny Long (Found at http://johnny.ihackstuff.com). It’s a relatively straightforward book, ableit a bit dated. Some of the examples no longer work but that, I suspect, is due to Google changing the way their search engine works. A lot of the penetration testing or examples are based around common query types such as inurl, intitle, filetype, etc. The question is: how creative are you at using them?

Of course, not everyone can afford the $45USD ($66CDN) for the book. In that case I’ve found some nifty tutorial links that can help you understand all this stuff without shelling out. But then again, you have to figure out what examples would work for you. :)

Befriending Google: http://www.antionline.com/showthread.php?s=&threadid=264746
Google as a Hacking Tool: http://www.antionline.com/showthread.php?s=&threadid=257512
Advanced Google Hacking: http://www.antionline.com/showthread.php?s=&threadid=264928
Google Security: http://www.antionline.com/showthread.php?s=&threadid=260714

Use AO’s seach engine to find more discussions on Google and what it does. This isn’t an absolute, ultimate list of the resources we have available but rather starting points.

And that’s the Last Word.