

Late-Breaking Computer Attack Vectors



PaulDotCom Enterprises, LLC

May 2008
Larry Pesce
PaulDotCom Enterprises, LLC
larry@pauldotcom.com

Concepts

- Phone wiping and information disclosure
- Who “owns” the firmware?
- Web input/output control, or lack thereof
- Un-authenticated authentication
- Defensive Spotlight - Document Metadata

“Hi, my name is Werner Brandes. My voice is my passport. Identify me.”
- Sneakers, 1992

Don't forget to wipe...

- A classic tale of information disclosure
- *It only takes a little bit of information!*
- Think about all of that info on such a small device such as a cell phone
- Great opportunities for further information gathering and potential social engineering
- How about other potential attacks from the data?

Boy buys phone. Boy loves and uses the heck out of phone. Boy breaks phone. Boy sends phone for repair, and offer him a replacement.

Replacement ends up in someone else's hands. With all of his info allegedly wiped by the manufacturer...

So with just the typical information we can begin to build a picture....

God forbid he does some form of online trading and forgot his password, that e-mails the account info and password to his phone...

Phone home?

- Refurbished iPhone bought from Apple
- Perform iPhone “restore”
- Available phonebook, e-mail, browsing history and passwords!



At this point, Apple doesn't have an official wipe utility that can do these things! Fortunately Rich Mogul has released a method – restore firmware, do not sync, then transfer a playlist the size of the device three times, effectively a 3 level wipe – however with the way that Flash memory allocates sectors, you could probably get away safely with only one write. These reasons are why it is difficult to do forensics on flash memory for file deletions – there are effectively no slack space.

Fix me?

- How exactly do you wipe this thing?
- More importantly, how do I wipe it if it is non-functional?
- Can I trust the manufacturer / refurbisher?
- Much larger problem than just a phone



So, what happens when this thing is busted, say for example I drop it in the toilet while powered on, and now I have no access to the device? I'm officially at a loss, if I want a replacement and or repair. The only real way to ensure my safety is to not trade-in/repair the existing and go and buy a new one. That gets real expensive real fast, certainly with an iPhone.

Now think about those hard drives that failed that the manufacturer wants bad for warranty exchange! When the drive won't even spin up, what type of data is there? Are you sure? Was it encrypted (whole drive or otherwise)? How can you be certain if the drive won't spin up.

How do you resolve? Keep the drive, and order a replacement? Some distributors (dell for example) will allow you to keep the drive and they'll send a replacement - they do, however charge a premium for this service.

This is where a risk analysis of these types of devices come in.

A real world example

- Buy used phone from e-bay
- Retrieve call history, address book, owner's name
- Google for (common) owners name, fail.
- Google for address book numbers and names.
- Determine industry
- Search again for (common) owners name in the industry...

In the process of finding an older Nokia phone that was vulnerable to some particular bluetooth attacks, I picked up a used Nokia phone off of ebay. When it arrived, the phone required a password to unlock. This particular phone was branded through Cingular, and where I'm already a customer, I figured I'd add it to my plan, and see if the nice lady behind the counter could unlock it for me. I gave her the story about how I bought it, and so on, so she used the super secret unlock code to do so, and sold me a prepaid plan.

I browsed through the phone in the store before I left, and noted that the address book was still intact. The phone was also named "Bob Smith's Phone". The nice lady behind the counter noted that it was odd, as they tell customers that all of that info is supposed to be stored on the SIM chip, and transferred to the new phone.

Just goes to show, don't always believe the cute sales girl.

Once I got home, I Googled "Bob Smith" with way too many results. So, I googled some of the names and numbers in the Address book. I found that all of the numbers were in a fairly large geographical area of Texas, and most had to do with the oil industry. Armed with that information, I was able to reduce my search for "Bob Smith" for Texas, and even looked at those results within the oil industry. Guess who I found. Yep. Bob Smith does land management and assessment for the oil industry. Basically, he's got the scoop on where the oil in Texas is, where you should buy land, and where you shouldn't. That would yield some interesting stuff from social engineering and or further attacks.

This without any access to e-mail. Just phone numbers and names in an address book.

Defenses

- Evaluate risk, Establish policies, educate
- Develop appropriate wiping mechanism
 - May require additional agreements from the vendor or increased costs
 - May involve a manual process depending on device!
- Physical destroy yourself

Who "owns" your firmware?

- Sure, we all patch and update our intel (and other) systems.
- We've got firewalls, AV, IDS/IPS, and a ton of other nifty security widgets for those systems.
- What about the other non traditional "systems"?
- All have an OS, a network connection and process information...

That's a lovely bunch 'o coconuts!

- Think about these devices!
 - Wireless routers
 - VOIP Phones
 - Firewalls
 - Network switches
 - ...and so on.



Trenchcoat Phlasher

- Recent talk of a Phlash
 - Send own firmware that breaks the device
 - Create a brick
 - Deny service
- A valid attack, but once delivered the phun is over



How about this?

- Reverse engineer firmware
- Add your own goodies, or replace all together
- Make user interface look, feel and act the same
- Offer to internet, and flash to systems you find
- Use “owned devices” for evil
- How do you know you are owned?

Toaster example

Wireless worm..

recording calls, traffic, passwords....

misdirecting traffic

Hack a Bike?

- Found one unlocked bike!
- Reverse engineered the hardware
- Grabbed the firmware
- Programmed their own unlock code
- Rode for free
- Hacked every bike they



Market Share = Power

- How about adding code to something that runs on a LOT of devices
 - Run a common OS
 - In use in fortune, 5 through 500 companies, Governments, healthcare, and carry traffic for a good portion of the internet
- Enter the Cisco IOS rootkit!
 - That's no bike!

Talkin' defenses

- Examine md5 checksums!
- Restrict where updates can come from
 - TFTP from the internet? I don't think so!
 - Separate management interface/network
 - Restrict management and upgrade to specific workstations at multiple points
- Proactive and regular monitoring

How do we detect these changes? Where's all the nifty detection tools? Where's the space to run them! Even with the recent Cisco exploits, you need to be real careful who (or what) has access to your management network

On device, firewall, intermediary switches and workstation

Manage changes, such as configuration

Pull the firmware from the device, and compare md5 sums from a separate repository

Web Input / Output Control

- Web applications that:
 - Require the user to abide by time limitations
 - Provide unique information
 - Sanitize form input
 - Rely on cookies to determine any of the above
- ...should all be done at the server side, not at the client

Here's a story

- Radio station contest for most adorable kitty:
 - Only vote once per day
 - Then added once per day, per e-mail.
- Different browser on same system = additional vote
- No confirmation to e-mail address for vote.
- All time and e-mail information stored local to participant browser!

Hack the Vote!

- Use a web proxy tool - Paros, etc and analyze/modify content and cookies in transit.
- Don't accept cookies at all! No cookie, no concept of time, or unique e-mail
- Further analysis revealed potential scripting possibilities - script random user agent, e-mail addresses and IPs (via TOR)
- This method puts all of the control of your controls in the hands of the end user!

Applicable to XSS

- Use javascript to sanitize form data?
- User can elect to drop javascript
 - use a proxy
 - turn off support in browser
- No more sanitization!
- Now we can deliver cookie/credential theft, hidden iframe



Defend!

- Process the submitted data on the backend
- Sanitize, then store for more processing
- Follow up with e-mail confirmation!
- XSS - allow all input, but sanitize on the server before passing on for more processing!

What authentication?

- Use Caller ID to identify and authenticate a phone user
- Sounds good, right?
 - How about spoofing Caller ID?
 - Easy to do, free if not cheap.
- Who's authenticating Caller ID?

My voice is my passport

- Bell Canada using voiceprint identification?
- Speak a phrase, get authenticated
- So, what happens is someone records you speaking the appropriate phrase and plays it back?
- Happens in the movies, circa 1992
 - much better, smaller recording gear

...identify me?

- Sure, those methods are great
 - Need to be combined with another form of authentication
 - Add a DTMF PIN
 - The longer the better!



A picture is worth a thousand words

- Just ask 0x80!
- Did an anonymous article, had a picture taken, admitted to cybercrime
- Photo was syndicated with EXIF Metadata
 - contained location data
 - Some research revealed a real small town
 - put two and two together...

Reveals more than you think

- More and more tools to analyze and information gathering!
- Metagoofil 1.4
 - Can analyze Office GUID info to reveal MAC addresses
 - Now we know office info, but potential hardware
 - Dell/Intel wireless with old drivers!

/* End */

- Web: <http://pauldotcom.com/>
- Wiki: <http://pauldotcom.com/wiki/> - Show notes for each episode
- Twitter: <http://twitter.com/haxorthematrix>
- Email: larry@pauldotcom.com



PaulDotCom Enterprises, LLC

