Adventures in Embedded Device Exploration and Exploitation
About this talk

This talk is:

- An introduction to embedded device hacking
- An inventory of useful tools, and how to Macguyver around them.
- A tale of some of the things I’ve learn, and screwed up

This talk is not:

- A case study or deep dive. Come back this afternoon for “Oh Dear... vulnerability hunting in access controls”
Hi! I’m Bobby.

I show people how to use things. Like Pentesting Software.

I get to pentest things.

I break stuff and call it research.

And I love my job.
What’s the Problem?

Ethernet and 802.11 chips are stupid cheap:
- Thousands^WMillions of new “network enabled” devices
- Embedded systems programming is very different…
- “Experience” is a problem
Basically...

Internet of Things

= Internet of Code
What kind of “things”

- Industrial Control
- Access control and physical security
- Cameras
- Power management
- Environmental Controls
- Appliances
- Printers
- MRI Machines
- IV Drug Pumps
Who owns the Embedded Devices?

A subject for Meditation…
Security Practices for Embedded Devices are stuck in the 90s...

And not the good part of the 90s.
Common Problems to hunt

- Default passwords
- Hardcoded, undocumented passwords
- Command injection
- SQL Injection
- No update path
- Crappy or non-existent crypto
- Key Management? Say What?
Show me the... Hardware
Hospira Lifecare PCA pump

- Unauthenticated Telnet as root
- Hardcoded Passwords
- Plain text wireless creds
- Directly editable drug database
- Common keying
HID Edge/VertX Card readers

- Unauthenticated Command Injection allows doors to be unlocked
- Vulnerable base OS
Cisco ASA Firewalls

- Memory Corruption
- And other goodies
Sounds fun. How can I play?
Step 0

Find something to hack on!

Look at what’s new, or interesting, or cheap.

Check out recent research and conference presentations

Protip: Get at least 3 of them, especially if it’s from China
Step 1

Identify the Attack Surface

Where does data enter or exit?

Management software or web services…

Examine the firmware
Step 2

Examine the hardware

Take it apart…

Look for interesting chips…

Look for interesting breakouts, vias, or pads
Useful Tools – Part 1

Good screwdrivers and tips, including security bits
Figure out what’s what

- Lots of pins: Interesting
- Big chips: Interesting
- Google everything printed on the chip
- Datasheets are your friend
Protip: Magnification is good. So is getting your eyes checked.
Useful Tools – Part 3

Multimeters

<- $5.99 at Harbor Freight

$300 at many, many places ->
Protip:

Keep several cheap DMMs on hand to test “iffy” circuits. You will cry less when they blow up 😞
Step 3

Get the firmware

The easy way: Firmware is downloadable from the website

The middle way: Reverse Engineer management software to get URL

The hard way: Hardware hacking: UART, JTAG, SPI, Chip Off, Glitching
Step 3 – The Hard Way

Start with the easy, and less invasive methods first.

<table>
<thead>
<tr>
<th>How Dangerous?</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mostly Harmless*</td>
<td>UART</td>
</tr>
<tr>
<td></td>
<td>JTAG</td>
</tr>
<tr>
<td>Watch your ground!</td>
<td>Probing flash chip leads</td>
</tr>
<tr>
<td>Magic Smoke Release</td>
<td>Chip-off flash reading</td>
</tr>
<tr>
<td>Probable</td>
<td>Microcontroller glitching attacks</td>
</tr>
</tbody>
</table>

*I may or may not have accidentally destroyed several hundred dollars worth of targets*
Useful Tools – Part 4

Protip: Learn to solder. Please.

The BusPirate
$30
Decent temperature controlled soldering rig ->
$90-ish

The JTAGULATOR
$150
Protip: MacGuyvering

No JTAGulator, no problem.

Turn device OFF
Use multimeter in continuity mode, datasheet, and magnifier to trace pins to confirm JTAG pinout
Useful Tools – Part 5

- DSLogic Pro: $100
- The Shikra: $45
- BusPirate: $27
Protip: MacGuyvering

The BusPirate can be turned into a low fidelity logic analyzer with the right firmware.
Useful Tools – Part 6

- ChipQuik Alloy
  - $17ish

- TMN-5000 EEPROM/Flash Programmer
  - $300

- Chip adapter
  - $40-70
Protip: MacGuyvering

You can use a BusPirate or Shikra and an adapter to dump flash memory.
Useful Tools – Last Ditch

ChipWhisperer
$300
Step 4

Extract the firmware

Binwalk is awesome and free.

<table>
<thead>
<tr>
<th>DECIMAL</th>
<th>HEX</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1288</td>
<td>0x588</td>
<td>CFE boot loader, little endian</td>
</tr>
<tr>
<td>65536</td>
<td>0x10000</td>
<td>Broadcom 96345 firmware header, header size: 256, firmware version: &quot;8&quot;, board id: &quot;6348GW-10&quot;, -CRC32 header checksum: 0x7FBD17C6, -CRC32 data checksum: 0x440BF79</td>
</tr>
<tr>
<td>65792</td>
<td>0x10100</td>
<td>Squashfs filesystem, big endian, version 2.8, size: 2623358 bytes, 428 inodes, blocksize: 65536 bytes, created: Thu Sep 17 18:07:36 2009</td>
</tr>
<tr>
<td>3426366</td>
<td>0x34483E</td>
<td>Sercomm firmware signature, version control: 0, download control: 0, hardware ID: &quot;DG834GT&quot;, hardware version: 0x4100, firmware version: 0x16, starting code segment: 0x8, code size: 0x7300</td>
</tr>
</tbody>
</table>

Get Binwalk at http://www.binwalk.org
Step 5

Audit and Reverse Engineer

If you’re lucky, it’s a Linux or unix-like RTOS

Look for weird services
Hardcoded passwords
Certificates or keys
Audit and Reverse Engineer

Disassembly tools are needed to dive deeper:

- **ILSpy** for .NET assemblies
- **IDA Pro** – Supports almost every CPU architecture. Expensive
- **BinaryNinja** – New, supports x86, x64, and ARM. Extensible.
Step 5

You found something, now what?

Hardware manufacture can be… squirrelly.
Coordinated disclosure should be your first option…

Full disclosure is a very big hammer. Use it sparingly.

Have fun!
Step Fin

HACK

ALL THE THINGS
And now…

Continue the discussion

bkuzma@coresecurity.com
@BobbyAtCore
http://www.coresecurity.com