The Internet of Fails
Where IoT Has Gone Wrong and How We're Making It Right

Mark Stanislav  <mstanislav@duosecurity.com>  Zach Lanier  <zach@duosecurity.com>
The Internet of Things
About The Internet Of Things

“The Internet of Things is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment.”, Gartner IT Glossary\(^1\)

“Machine to machine (M2M) refers to technologies that allow both wireless and wired systems to communicate with other devices of the same type.”, Wikipedia\(^2\)

IoT Growth Estimates

- **Gartner**: 26 billion units by 2020\(^3\)
- **ABI Research**: 30 billion units by 2020\(^4\)

4. [https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne](https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne)
Cool! So What’s Wrong?

- **Pervasiveness:** You won’t have one IoT device, you’ll have ten.
  - That’s a lot of new attack surface to your life and/or business

- **Uniqueness:** IoT devices are a wild-west of mixed technologies.
  - How do I patch firmware on these dozen devices?
  - Which random vendor made the hardware inside this device?

- **Ecosystem:** Your vendor may be leveraging six other vendors.
  - Where’s your data going once it enters that IoT device?
  - Who has access to your network via proxy connections?
The Internet of Things “Line of Insanity”™

- Sane
- Reasonable
- Questionable
- Insane

IP Camera
Door Bell
Door Lock
Egg Tray
Cheap Hardware, Unlimited Possibilities

- Electric Imp ($25)
- Gumstix ($169)
- Arduino ($75)
- Pinoccio ($59)
- Raspberry Pi ($35)
Plenty Of Choices, How Do You Determine Security?

Philips ($60)  INSTEON ($30)  LimitlessLED ($23)

Vendors could each use different hardware, software, APIs, third-party service providers, and patching mechanisms.
IoT Ecosystem

- such network
- very service
- many API
- wow
- so security
- 100% device
- much connect
What’s Better Than One Vulnerable Device?

- If-This-Then-That (IFTTT) supports over 110 platforms, services, and devices
- Allows for event-based actions across disparate technologies
  - If the CO\(^2\) in this room is unsafe, change my lightbulb to be red to warn me
- This behavior will become a consumer expectation rather than merely a “nice to have”

Interconnected Vulnerable Devices!
The Government Is Watching — And That’s Good!

June 3rd, 2013
Software & Information Industry Association asks FTC to be careful with IoT

November 21st, 2013
Internet of Things - Privacy and Security in a Connected World Workshop

January 8th, 2014
FTC Commissioner Maureen Ohlhausen sits on panel at CES about IoT

February 7th, 2014
FTC approves final order settling charges against TRENDnet, Inc.

February 18th, 2014
US CERT works with IOActive to resolve Belkin WeMo vulnerabilities

Challenges Faced
Hardware Security

- Many devices use generic SoCs/boards
- Quick development, few security features (“HW hacking made easy”)
- Prevalence of same components, firmware, etc. means one bug affects many products
- Little expertise required to design, build, and ship an “IoT Product”

Least common denominator:

Logic analyzer + Bus Pirate + UART headers = Console!

```
# id
uid=0(root) gid=0(root) groups=0(root)
```
Software Security

- Development environments don’t necessarily make security controls/options “clear”
- Selected platform may drive/ restrict language choices

“Me write Python/Ruby/ Node/... pretty one day” (or worse, C)
- History repeating...?

Quick grep for potentially dangerous behavior in someone's Contiki project
Challenges: Software Security (Cont’d…)

- Selected platform often locks dev/vendor into given OS choice
- Proprietary OSes (such as ElectricImp) - don’t peek inside the black box!
- Linux, Contiki, QNX, et. al (all with their own issues)
- Little consideration given to least-priv, mitigations, hardening, etc.
- Third-party dependencies
  - Inherited bugs/attack surface
- Also applies to mobile
Comms/Network Security

- WiFi goofiness (“device as AP”, no WPA, exploitable behavior, etc.)
- Plaintext protocols or poor crypto at transport layer
  - ...or lack of cert pinning where SSL/TLS actually used
- Unprotected FW updates/downloads
- Otherwise seemingly unnecessary services listening
  - Telnet, SSH, FTP, you name it...
- Shared accounts/auth material for “support” or updates
- Use of technologies such as ZigBee and cellular introduce additional security considerations

"Cellular made easy"
Platform* Security

- Everything that uses an HTTP GET/POST has become an “API” to the average developer
- Input manipulation is a less obvious concern when developers do mobile and embedded
  - Yup... OWASP {Mobile,Web} Top 10
- Leveraging third-party service providers introduce exponential complexities and further increases potential attack surface
- Quick & Dirty cloud infrastructure yields poor accessibility and potentially confidentiality

* for our purposes “Platform” also includes supporting infrastructure, services, frameworks, etc.
User Awareness & Behavior

- Users may not know (let alone care) how to update device firmware or apps
- Disparity in management: web console v. mobile app v. physical "update" button
- Also they just want to use the !@#$ thing now!
- Lack of feedback or notification for updates or errors
- How does a user know their IoT device was updated or, worse, compromised?
[Failures]
Oh, You Wanted Authentication on Your Camera?

- **Issue:** Some TRENDnet IP camera models didn’t authenticate users connecting to http://camera-ip/anony/mjpg.cgi which exposed actual video feeds of people’s cameras.

- **Hypothetical Exploit:**
  - Google for “inurl:/anony/mjpg.cgi”
  - Be a big creep that nobody likes

- **Fix:** Always verify all expected “private” URL actually require authentication. This is *easily* accomplished with a curl script or Selenium.

http://console-cowboys.blogspot.com/2012/01/trendnet-cameras-i-always-feel-like.html
You Get Keys, and You Get Keys… EVERYBODY GETS KEYS!

- **Issue:** IOActive determined that Belkin’s WeMo devices were including their GPG signing key and password inside of the firmware itself.

- **Hypothetical Exploit:**
  - Retrieve firmware signing key + password
  - MITM firmware feed announcing updates
  - Own WeMo devices
  - Flip lights and stuff

- **Fix:** Don’t try to “hide” secret data in firmware, a lot of people are looking there. Signing firmware is great… just don’t let attackers sign it, too :)  
  
LIFX

**Issue:** Context found that LIFX utilized a hardcoded symmetric key for encrypting data across 6LoWPAN, including WiFi credentials

**Hypothetical Exploit:**
- Give a LIFX user a new bulb as a “gift”
- Get creepily close to their house
- Wait for them to add the new bulb, sniff traffic
- Decrypt packet capture with symmetric key
- Jump on their WiFi network and do bad things

**Fix:** No, seriously, for the last time, don’t hardcode passwords/keys/etc. in your firmware. STAHP.  
http://contextis.co.uk/blog/hacking-internet-connected-light-bulbs/
"Home Automation Gateway"

Linux-based gateway talks RESTful HTTP to "cloud"-based service, receives commands (schedules, metering data, etc.), relays commands to smart plugs/meters via ZigBee.
IZON IP Camera
Telnet And A Hardcoded Root Password? Let’s Do This!

• **Issue:** The camera’s mobile app contained hardcoded root credentials so that it could initiate firmware upgrades by connecting over Telnet and echoing out a shell script to start the process.

• **Hypothetical Exploit:**
  - Run strings on the decrypted mobile application
  - Connect to any camera you can reach via Telnet as root
  - View the *admin* password for the camera’s web interface and login

• **Fix:** Don’t use Telnet for anything… ever. Don’t hardcode passwords… ever. Promise?
If You Want To Protect Data... Protect It.

**Issue:** Unencrypted camera “alert” video clips were uploaded to Amazon S3 into one bucket and protected only by an MD5-string filename. Oh, and no SSL.

**Hypothetical Exploit:**
- Generate MD5 strings with the filename format
- Be really, really, really patient
- View random videos of cats knocking stuff over

**Fix:** Leverage the AWS Identity and Access Management (IAM) functionality to provide unique access control per customer to only their own data.
API = Always Poorly Implemented

**Issue:** API calls for third-party services were done without SSL and used an MD5-sum of the user’s password as a secret.

**Hypothetical Exploit:**
- Go to Starbucks and hopefully get a PSL
- MITM network traffic
- Wait for someone to check their video camera
- Retrieve their MD5’ed password, crack, repeat

**Fix:** If you setup third-party credentials for your customers, do NOT transmit their real account password. Also, make sure your vendors understand the basics of API security.
The Bigger Picture of This Research, FWIW...

- Unencrypted storage of customer data
- Hardcoded web service credentials
- Passive customer sign up for 3rd party services
- Unencrypted local video streaming
- Information leakage
- Poor password security
- Insufficient data security for firmware upgrades
- Numerous network services
- Failure to properly implement HTTP digest
- Long-life (clear-text) API tokens
- "Open" Internet proxy
- Lack of authentication for customer data
- Poor mobile security
- Generic ODM firmware
- Clear-text API calls
- Passive Wi-Fi recon.
- File deletion control broken
- Hardcoded OS credentials
- Poor customer data segmentation
[Redacted]*

*The issues presented shown are real but the context has been changed

Coordinated disclosure is rarely perfect :}

[Redacted]
Session Handling: Time is Not On Your Side

- **Issue:** Session IDs are “generated” by using *only* the exact UNIX epoch timestamp of when you logged into the service for this IoT device.

- **Hypothetical Exploit:**
  - Enumerate 172,800 recent epoch timestamps
  - Set your session ID to each timestamp
  - Send a GET request to determine validity
  - “Become” a user with a browser header

- **Fix:** Use your web framework’s default session handler that hopefully isn’t non-random.
Don’t Trust What You Haven’t Verified

**Issue:** Purchasing in-app credits for use with the device via the app store lets the mobile application dictate that a purchase occurred.

**Hypothetical Exploit:**
- Pick a number... any number
- Make an API call with that number
- Gain that many “things” for your account

**Fix:** Don’t let a mobile app be the authority on any account balances. Always use a transaction log on the backend to reconcile what purchases have occurred and what balances should be.
Hiding in Plain(text) Sight

- **Issue:** A chicken-and-egg problem existed where sensitive details about a user were provided *prior* to authorization from said user.

- **Hypothetical Exploit:**
  - Ask a user to be your friend
  - Data is transmitted over the wire about that user
  - User gets to decide if they want to share data
  - ...wait a second...

- **Fix:** Don’t transmit data ahead of authorization, even if the user interface won’t expose it. If it goes over the wire, it’s out of your control now.
Heart Bleed And The Internet Of Things

- PKI is (sadly?) a critical component of IoT security
  - Mobile apps speaking to embedded devices
  - Embedded devices speaking to services
  - Services speaking to other services
  - Developers leveraging service APIs
- How many IoT vendors are concerning themselves over Heart Bleed implications?
- Do you think most random Chinese ODMs are going to rush to get your devices patched and re-keyed? Or even release a notice?
Vendors You’ve Never Heard Of
There's A Shift Underway You Should Know About

- The IoT growth that we’re all expecting won’t just be from large vendors like Belkin, TRENDnet, Cisco, and Ericsson
  - Postscapes¹ and Wolfram Alpha² list a few hundred IoT-related companies, *most of which you’ve likely never heard of*

- Crowd-funding web sites are going to produce many of the newest IoT devices we all want to use
  - Entrepreneurs likely have no experience with information security, nor the budget to afford help
  - They also won’t know what a “security researcher” is or why you’re contacting them…

---

¹ http://postscapes.com/companies/
² http://devices.wolfram.com
Vulnerability Handling & Disclosure Awareness

- Small vendors (and some big ones) fail to get it, **or just simply don’t know**
  - “But, **why** would anyone want to hack this device? And why would they want to tell us or talk about it publicly?”
- Few-to-no resources for small vendors to handle this
- Nascency of “IoT” means some researchers may not know either
  - And we’d like for them to stay out of jail
A New Initiative
A New Initiative — BuildItSecure.ly

Our Goals for the "Internet of Things"

- FOCUS effort towards crowd-funded, small commercial and bootstrapped vendors
- BUILD partnerships and goodwill between IoT vendors and the security community
- COORDINATE efforts to incentivize security researchers for reporting vulnerabilities
- CURATE informational resources to help educate vendors on security best practices
- PRESENT research at relevant events and be a point of contact for press inquiries
Our Timeline

**February, 2014:**
+ Initial announcement of this initiative at BSides San Francisco

**March - April, 2014:**
+ Established relationships with an initial set of partners/researchers

**June, 2014:**
+ Spun-up Bugcrowd for our launch vendors to leverage for bug triage

**July, 2014:**
+ Pinoccio ships researchers their first hardware for testing!

**August, 2014:**
+ Presented at DEF CON on status and announced three new vendors

**September, 2014:**
+ Started discussions with future researchers and vendors
What’s Next?

• **Complete our first round of vendor hardware testing**
  • Pinoccio has provided hardware for two researchers to test
  • Bugcrowd has been setup for our vendors + researchers to utilize
  • We need to figure out testing/reporting/triage/fix timelines still
  • Get some bugs reported, get them fixed, reward researchers (?)

• **Figure out what works, what doesn’t — and then expand!**
  • We want to better understand our processes before growing
  • As we add vendors to test for, we’ll add more researchers
    • We’re a big fan of *results* and not just pretending help people ;)
  • We’re growing slowly, on purpose, and happy about it
Conclusion
Lessons Learned

- People are way less cynical in information security than you may think — don’t be afraid to ask people for their input and help.

- Take your timeline and double it. The same people who are going to provide the best help/effort are already pretty damn busy.

- Focus on quality, not quantity when it comes to people/partners.

- Find measured successes and create specific milestones.
Conclusion

- IoT is still malleable enough to help make a positive impact
- BuildItSecure.ly could help consumers make better decisions
  - Plenty of vendors — let’s find a few that care about security
  - Worst case? We do some research and help a few companies!

ONE DOES NOT SIMPLY

FIX IOT SECURITY ALONE
Thanks! Questions?

Mark Stanislav
mstanislav@duosecurity.com
@markstanislav

Zach Lanier
zach@duosecurity.com
@quine

http://BuildItSecure.ly/
buidlitsecurely@duosecurity.com
@BuildItSecurely