How We’re Failing To Secure The “Internet Of Things”

Mark Stanislav <mstanislav@duosecurity.com>

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]

[1] http://www.gartner.com/it-glossary/internet-of-things/

[2] http://en.wikipedia.org/wiki/Internet_of_Things

[3] http://www.gartner.com/newsroom/id/2636073

[4] 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?

Other People’s Research

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

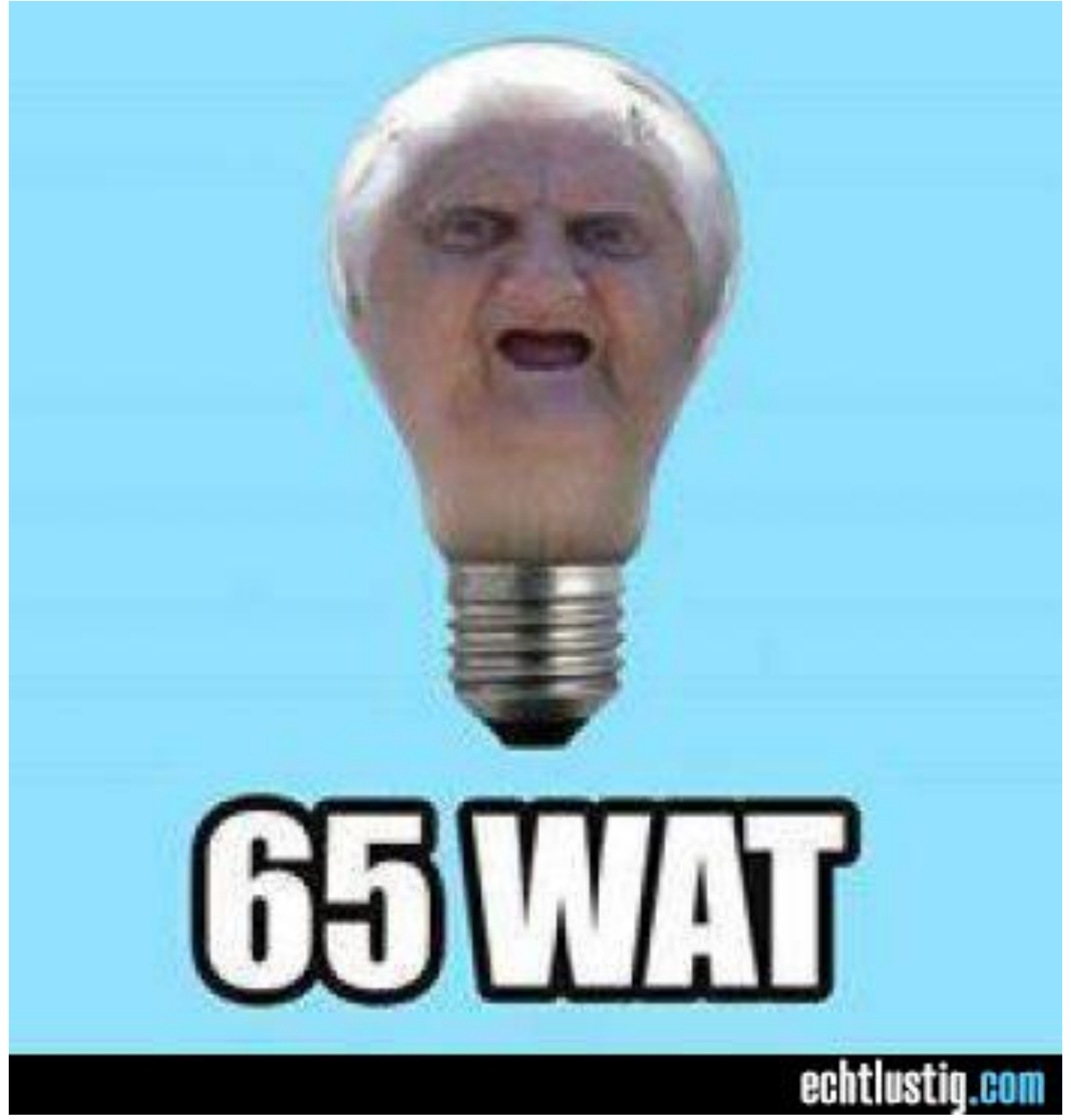
You Get Keys, and You Get Keys... EVERYBODY GETS KEYS!

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

http://www.ioactive.com/news-events/IOActive_advisory_belkinwemo_2014.html

MD5(MAC Address) != A Good Secret Token

- **Issue:** Nitesh Dhanjani found that Philips Hue bulbs used the MD5-sum of the MAC address for an “authorized” administrative device as a secret token.
- **Hypothetical Exploit:**
  - Find all Hue lightbulbs on the network
  - Loop each MAC address from *arp -a*, MD5 it, and then test to see if you can run a command
  - Make people think they are going insane
- **Fix:** If it can be found on a network, in clear-text, without even a MITM… probably not a good “secret”.



http://www.dhanjani.com/blog/2013/08/hacking-lightbulbs.html

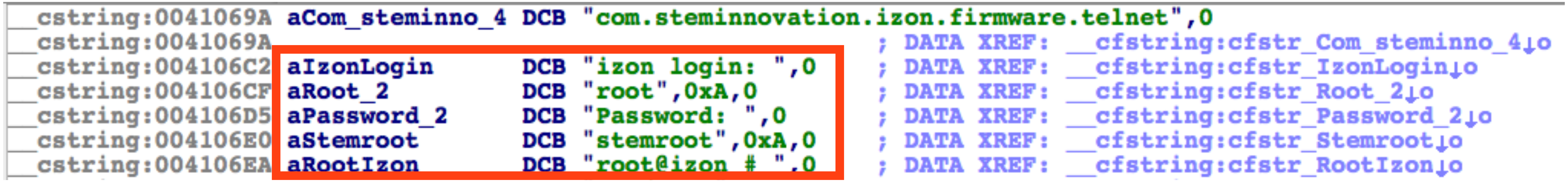
I Have a Gift For You: New Firmware!

- **Issue:** Dan Crowley noted that the MiCasaVerde VeraLite allows "guest" users to upgrade firmware despite that function being intended for admins.
- **Hypothetical Exploit:**
  - Google "inurl:/cgi-bin/cmh/upgrade_step1.sh"
  - Visit the device at a super-duper secret URL: http://device-ip/upgrade_step2.sh?squashfs=[url]
  - Test out your new "features"
- **Fix:** Verify administrative functions actually map appropriately to available roles. Seriously, just try to.

http://www.exploit-db.com/exploits/27286/

IZON IP Camera

Telnet And A Hardcoded Root Password? Let's Do This!



```
__cstring:0041069A aCom_steminno_4 DCB "com.steminnovation.izon.firmware.telnet",0
__cstring:0041069A                                                   ; DATA XREF: __cfstring:cfstr_Com_steminno_4↓o
__cstring:004106C2 aIzonLogin      DCB "izon login: ",0              ; DATA XREF: __cfstring:cfstr_IzonLogin↓o
__cstring:004106CF aRoot_2         DCB "root",0xA,0                  ; DATA XREF: __cfstring:cfstr_Root_2↓o
__cstring:004106D5 aPassword_2     DCB "Password: ",0                ; DATA XREF: __cfstring:cfstr_Password_2↓o
__cstring:004106E0 aStemroot       DCB "stemroot",0xA,0              ; DATA XREF: __cfstring:cfstr_Stemroot↓o
__cstring:004106EA aRootIzon       DCB "root@izon # ",0              ; DATA XREF: __cfstring:cfstr_RootIzon↓o
```

- **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?

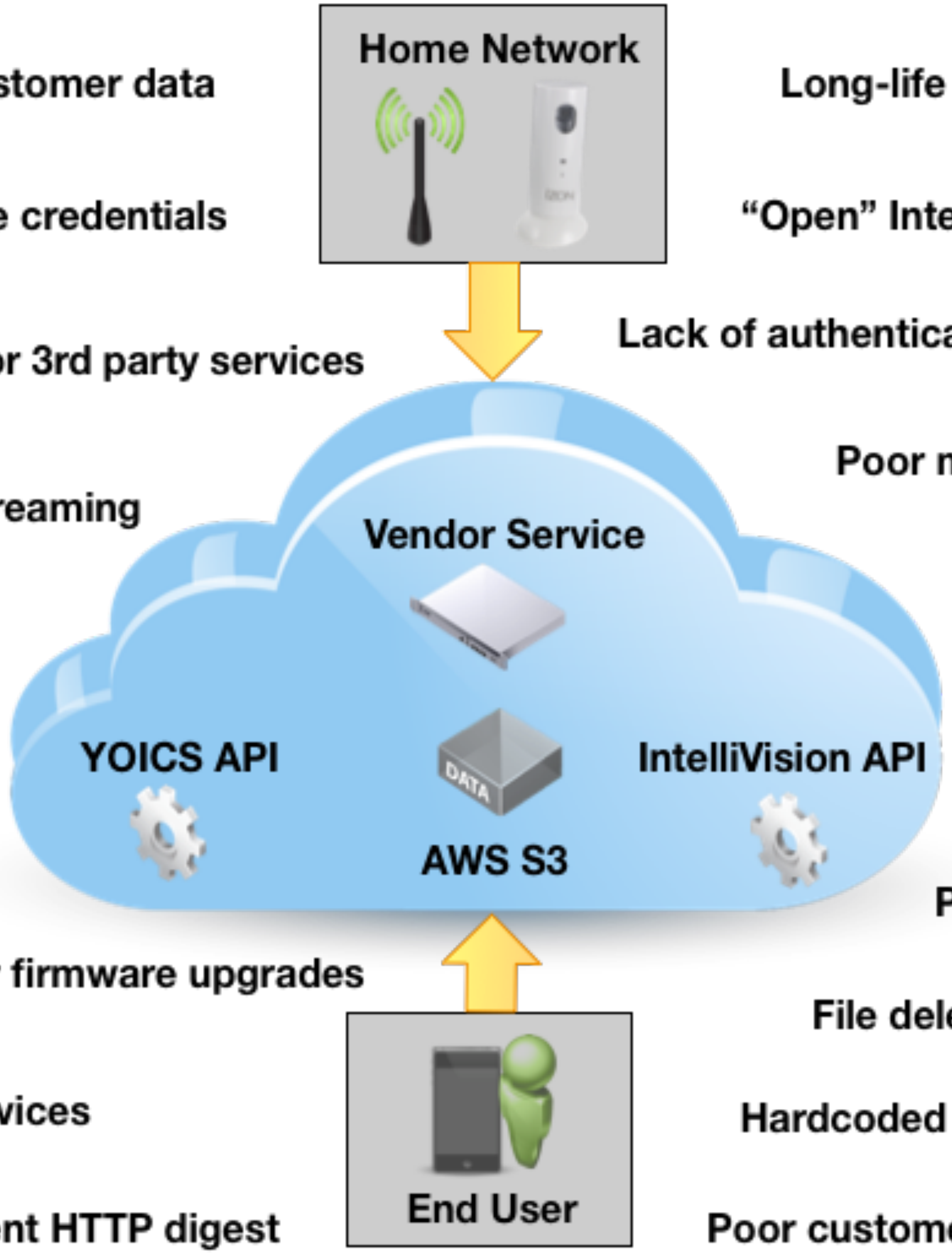
If You Want To Protect Data… Protect It.

- **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 are all real. The IoT device will not be named as the vendor is actively remediating these problems for their users.

Coordinated disclosure is rarely perfect :)

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.

I'M GLAD WE'RE SWITCHING TO 64-BIT, BECAUSE I WASN'T LOOKING FORWARD TO CONVINCING PEOPLE TO CARE ABOUT THE UNIX 2038 PROBLEM.
WHAT'S THAT?
REMEMBER Y2K? THIS COULD BE EVEN WORSE!

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.

The Government Is Watching

June 3rd, 2013

Software & Information Industry Association asks FTC to be careful with IoT[1]

November 21st, 2013

Internet of Things - Privacy and Security in a Connected World Workshop[2]

January 8th, 2014

FTC Commissioner Maureen Ohlhausen sits on panel at CES about IoT[3]

February 7th, 2014

FTC approves final order settling charges against TRENDnet, Inc.[4]

February 18th, 2014

US CERT works with IOActive to resolve Belkin WeMo vulnerabilities[5]

[1] https://www.siia.net/blog/index.php/2013/06/siia-to-ftc-internet-of-things-requires-technology-neutral-policies-and-flexible-privacy-framework/

[2] http://www.ftc.gov/news-events/events-calendar/2013/11/internet-things-privacy-and-security-connected-world

[3] http://www.adweek.com/news/technology/will-washington-move-quickly-regulate-internet-things-154863

[4] http://www.ftc.gov/news-events/press-releases/2014/02/ftc-approves-final-order-settling-charges-against-trendnet-inc

[5] http://www.kb.cert.org/vuls/id/656302

10 Hints to More Secure IoT Development

1. Do not hide sensitive data in your firmware or mobile apps
2. Utilize a platform that supports auto-update functionality
3. If it goes over a network, it almost certainly should be encrypted
4. Verify API requests (e.g. HMAC signature) even with SSL transport
5. Encrypt and segment customer's data from one another at rest
6. Research an chipsets you buy from an ODM for known issues
7. Use bcrypt/scrypt/PBKDF2 for one-way data storage
8. Automate tests to verify data is only accessible when expected
9. Work with reputable vendors and don't trust their security, either
10. Data outside of your server is no longer trustworthy data

What If *My* Company Is Contacted About Issues?

Do:

- Inquire about what issues they found, any proof-of-concept examples they can provide, and a timeline/scope of testing
- Ask if they are planning to disclose this research on any specific date and request they delay disclosure if you need more time
- Thank them for their time/expertise to research this technology

Do Not:

- Say you will sue them for violating some law you don't understand
- Ignore their emails and assume these flaws will all fix themselves
- Think your team can handle fixing all of the problems without at least a quick conversation to hammer out the details

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
  - Postscapes[1] and Wolfram Alpha[2] 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

Announcing BuildItSecure.ly

Our Mission

- Provide resources, guidance, community for small IoT developers/builders to make informed security decisions
- Incentivize vulnerability research and reporting for these devices

Why?

- Small vendors don't understand "security research"
  - "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
- They also have little money to spend on a security consultant from a firm to review their technology

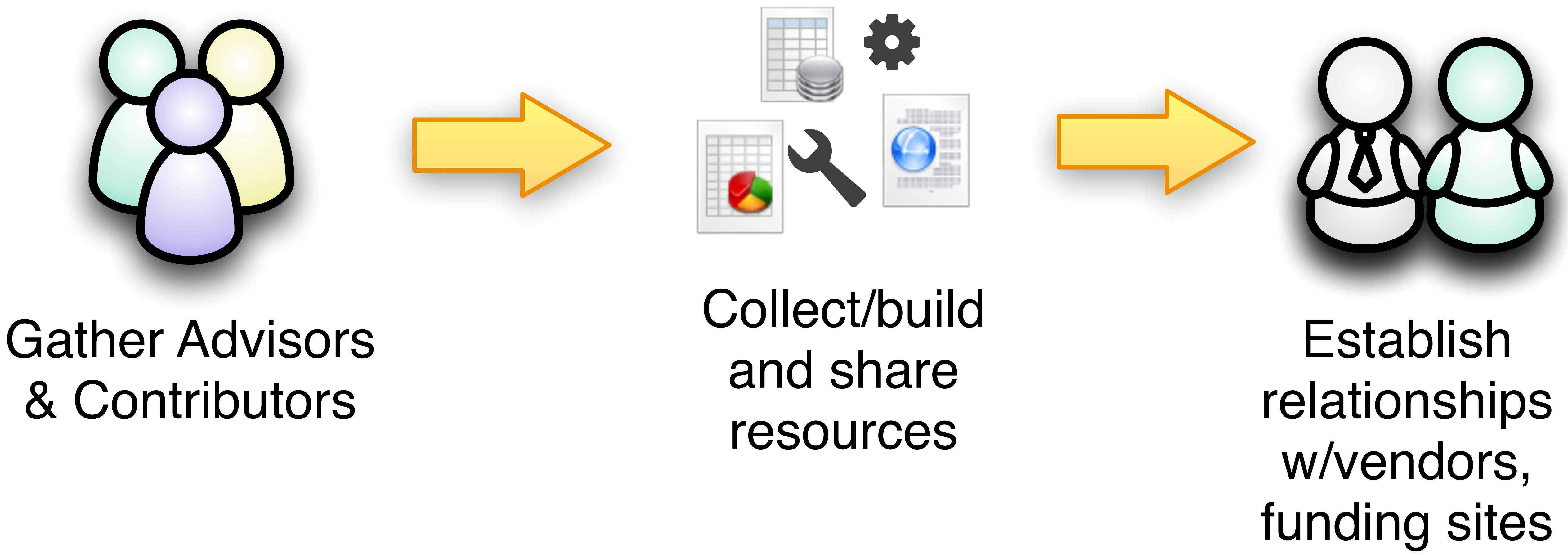
BuildItSecure.ly

Zach Lanier

Sr. Security Researcher

DUO

Phase 1 - Initial Build Out/Partnering (est. April, 2014)



Gather Advisors & Contributors

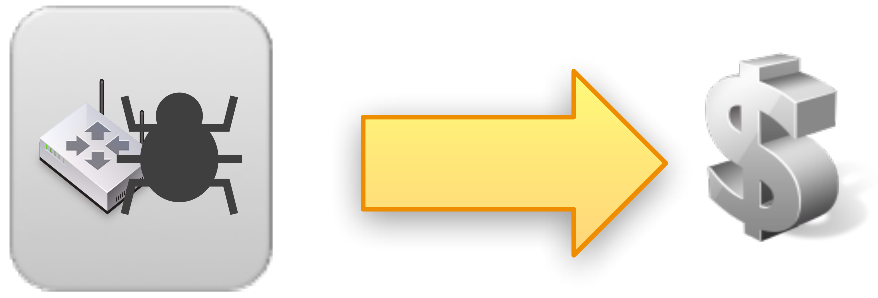
Collect/build and share resources

Establish relationships w/vendors, funding sites

- Establish a core team of advisors and content contributors
  - Curate secure development documents and disclosure guidelines
  - Build new diagrams, flow charts, info graphics to transfer knowledge
- Form relationships with crowd-funding sites, IoT-centric hardware vendors/platform providers, and other relevant organizations

Phase 2 - Rewards/Incentivization (est. July, 2014)

- Build a reporting and reward/ incentive program
  - Partnering with **Bugcrowd** on this
- Rewards could include:
  - Recognition
  - Monetary reward
  - Device reward
  - Schwag!



Thanks! Questions?

mstanislav@duosecurity.com @markstanislav

http://www.uncompiled.com

goto; conference

Please evaluate this talk via the mobile app!

BuildItSecure.ly

http://BuildItSecure.ly/

@BuildItSecurely

DUO