Stalking a City for Fun and Frivolity

“Pull pin, point toward privacy insurance claimant”

Brendan O’Connor
Malice Afterthought, Inc.

DRAFT SLIDES
Please go to http://www.maliceafterthought.com for updated slides

Hi! I’m Brendan

- CTO/DSS, Malice Afterthought
- Law Student: IANAL(Y)
  - Anything you hear in this presentation isn’t legal advice
  - Most of it isn’t even a good idea
  - You’ve been warned

Quick Tangent: Weev

- U.S. v. Auernheimer
- "Brief of Meredith Patterson, Brendan O'Connor, Sergey Bratus, Gabriella Coleman, Peyton Engel, Matthew Green, Dan Hirsch, Dan Kaminsky, Samuel Liles, Shane MacDougall, Brian Martin, C.\ Thomas, and Peiter Zatko as Amici Curiae Supporting Appellant" -- I got to write this!
- Still, though: IANAL. Yet. (One more year!)

DARPA Cyber Fast Track

- CREEPYDOL IS NOT CFT WORK. I CANNOT EMPHASIZE THIS ENOUGH: DARPA CFT WAS NOT IN THE BUSINESS OF HAVING ME CREATE CREEPY THINGS.
- That said, two CFT contracts did let me build two of the core systems: Reticle, and the visualization system.
- Thanks!

Foreword: Democratizing Surveillance

http://www.flickr.com/photos/68979377@N00/3745750194

I. Foreword: The Democratization of Surveillance

A. "Security is really the government's area."

1. This was actually said to me by my sister recently, indicating that I'm failing in my duty to educate my family.

2. Those of us in this room know that the government isn't very good at securing things by means *other* than throwing them in prison for large amounts of time.

3. Nonetheless, the government has a near-monopoly on surveillance.

“Only the Good Guys”

http://www.flickr.com/photos/chberge/3753079527

4. When it doesn't, the perception of the general public is that "only good guys" have access to terrifying surveillance technology. This is *our fault* for not correcting this misperception, though groups reporting on, e.g., all the BlueCoat boxes they've found in repressive governments are certainly helping. Heck, PRISM was leaked, and this is *still* the thing I’m hearing: people think “hey, the NSA needs that.”

“Sunlight is the best disinfectant”

http://www.flickr.com/photos/andyz/3857625392

B. "Sunlight is the best disinfectant."

1. A recent study showed that cops wearing sunglass cameras were 88% less likely to commit actions resulting in complaints, and 60% less likely to use force; when they did use force, those officers wearing lapel cameras were consistent in using the least amount of force possible in a situation. This effect was not duplicated in officers refusing to wear the cameras.

2. If we can see what's going on---if we can look back at our government---we have the opportunity to make sure it works as efficiently and safely as possible. If not, we are subject to blackmail, extortion, and threats. (See Aaron Swartz.)

So we need sunlight---but we need it quickly, and where our natural inclination, our natural sunlight, is not. Those of you who are weapons buffs may know that this isn’t a photo of the sun: it’s a picture of the blast caused by Tsar Bomba, the largest nuclear weapon ever detonated.

So I get called a stalker

http://www.flickr.com/photos/simplyjessi/6333279524/

Wait, wrong stalker. This is an adorable cat, apparently named Stalker. People don’t call me an adorable cat.

So I get called a stalker

http://www.flickr.com/photos/dcagne/424124810

Much better. As I was saying,
C. Why I do "creepy" work.

1. The only effective way to raise the issue of creeping surveillance and loss of privacy is to make clear that *anyone*, not just "the good guys," can use this technology for good or for evil.

2. The only way to make it clear is, of course, to release software that does it in a nice, user-friendly package.

Extremely Serious Disclaimer

To be clear: I do not endorse using this software, or any software, for criminal purposes. We're hackers, not criminals. I want the fact of this software's existence to help shape habits and, hopefully, the next generations of mobile devices; perhaps they won't be designed (at the protocol level) to leak so much information so widely.

Goal: Passive Wireless

http://www.flickr.com/photos/library_of_congress/2163911718

II. Goals

A. How much data can be extracted from passive wireless monitoring?

1. More than just from a network trace---remember that when not connected to a wireless network, WiFi devices send out lists of their known networks, asking if anyone can help them.

2. As soon as a device thinks it's connected to WiFi, all its background sync services will kick off again---DropBox, iMessage, all the rest. So we'll immediately know that certain services will be in play.

3. Over unencrypted WiFi, all the traffic sent by a device is exposed. Even if we can't see both sides of every message, we can learn a lot from what we do see---especially if we know how a given protocol operates.

4. How much better could we do if we had not one sensor, but ten? Spread out over an area? Now we have geolocation, time and place analysis, etc.

5. If we're tracking over a large area, we don't just want to know traffic and devices: we want to know people. Can we take data and find people? (I don't want your SSN, I want your name.)

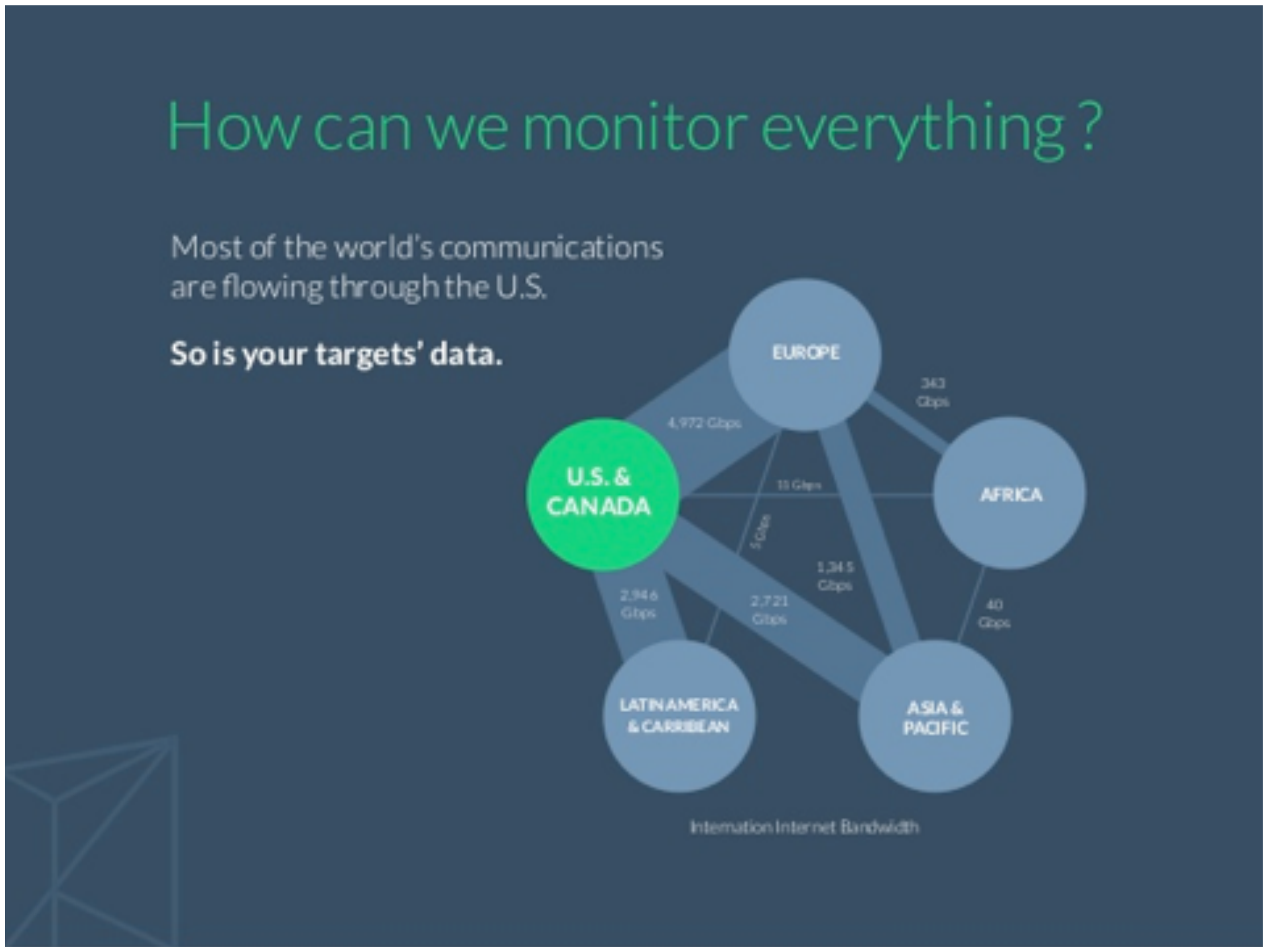
Goal: Large-Scale Sensing Without Centralized Communications

http://www.flickr.com/photos/christmaswithak/2732857205

B. Can we do large-scale sensing without centralized communications?

1. If we centralize communications, life is simple; everyone phones home---but a compromised node gives every attacker the location of the mothership.

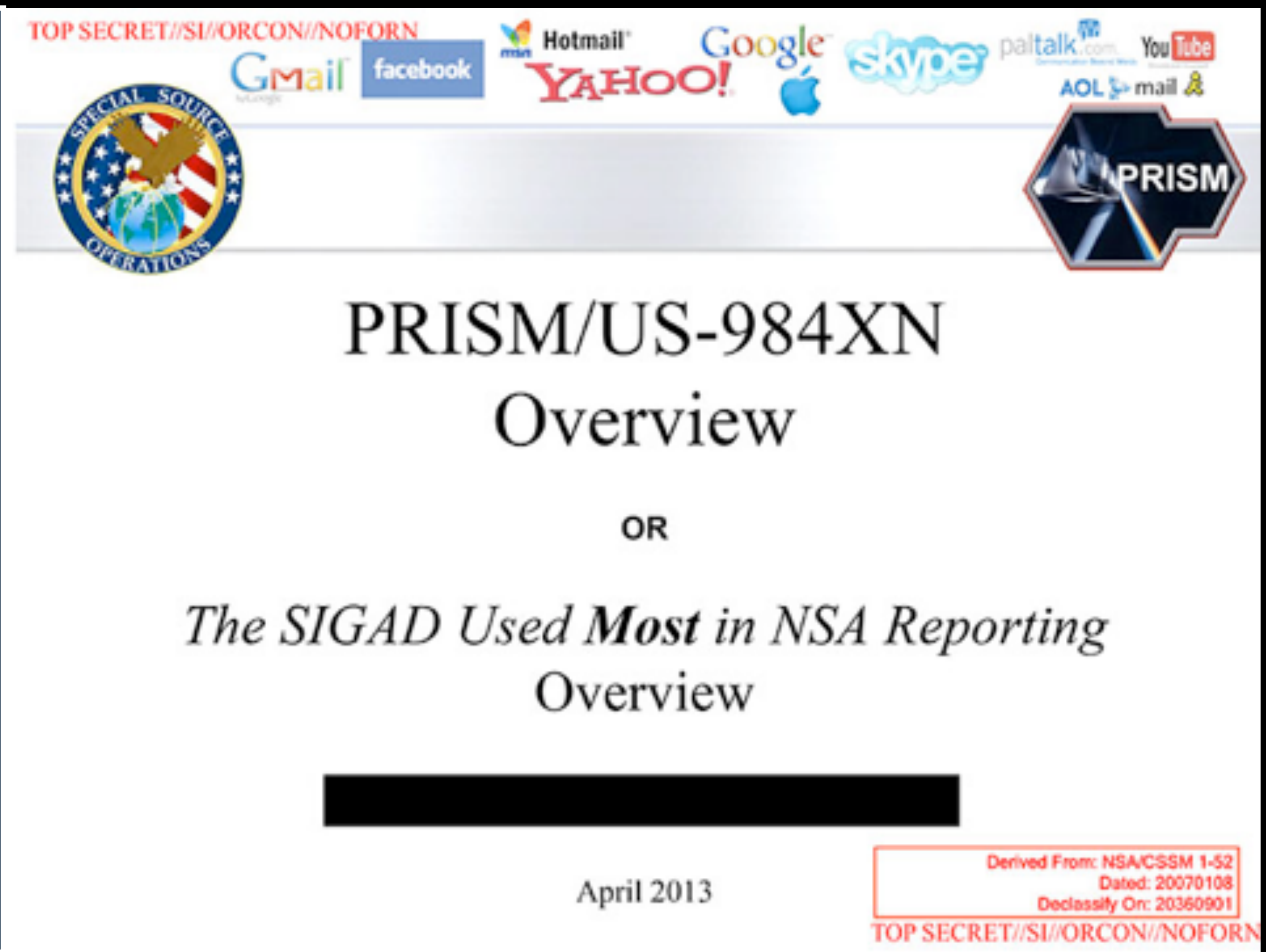
2. Centralized communications decrease resistance to attack, and prevent you from responding agilely to attack.



How can we monitor everything?

Most of the world's communications are flowing through the U.S.

So is your targets' data.



GMail
facebook
Hotmail
YAHOO!
Google
skype
paltalk.com
YouTube
AOL mail
SPECIAL SOURCE OPERATIONS
PRISM

TOP SECRET//SI//ORCON//NOFORN

PRISM/US-984XN
Overview

OR

*The SIGAD Used **Most** in NSA Reporting*
Overview

[redacted]

April 2013

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20360901

TOP SECRET//SI//ORCON//NOFORN

Goal: Intelligibility

http://www.slideshare.net/EmilandDC/dear-nsa-let-me-take-care-ou

C. Can we present massive amounts of this data in a way that is intelligible by mortals? User-friendly? Still secure?

1. Group One of high security products: incredible technology, terrible UI. This causes low adoption, or (possibly worse) mistakes in use. Systems fail, people die. Examples: Pidgin-OTR, or PGP/OpenPGP.

2. Group Two: Concerns about technology, great UI. This causes adoption, but can cause massive problems later (if the concerns are borne out). Examples: HushMail, Silent Circle.

3. Group Three: Good technology, great UI. This is wonderful, but incredibly hard to do (because UI masters are usually not security wizards). Example: CryptoCat, RedPhone.

4. We would aspire to have CreepyDOL be in Group Three, through a variety of methods to ensure secure communication in relatively-intelligible ways. *This is an ongoing process.* Our code is open source, to allow verification.

Background: Academic Sensor Networks Rock!

http://www.flickr.com/photos/22240293@N05/3912598338

(This is the MIT CS building, if you're wondering. They have an awesome sensor network, and their papers are always accompanied by the *weirdest* floor plans.)

III. Background

A. Sensor Networks

1. Academic researchers have spent tons of time and resources on these. MANETs, other advances in technology have resulted.

2. A lot of these have uW power levels, and sacrifice languages, OS, and cost to get there---especially cost, with many nodes costing $500 or more. Each.

3. I can't afford this. I want something I can afford to break, to lose, and even to have stolen. I want it an order of magnitude cheaper, and I want it to run Linux. (Ubuntu or Debian, if possible.)

Background: Large-Scale Surveillance

- Believe it or not, we knew this was happening before PRISM
- In my original outline: "One can assume that they have solved all of the problems involved in CreepyDOL before me, and that they should, rightfully, be cited as prior art. I'd love to do so; as soon as they publish their work, I'll be happy to cite them."
- Heh heh heh.
- Pour one out for the Intelligence Community: a lot of this stuff is a pain to figure out

Hardware!

So now let’s talk about system architecture. First: Hardware.

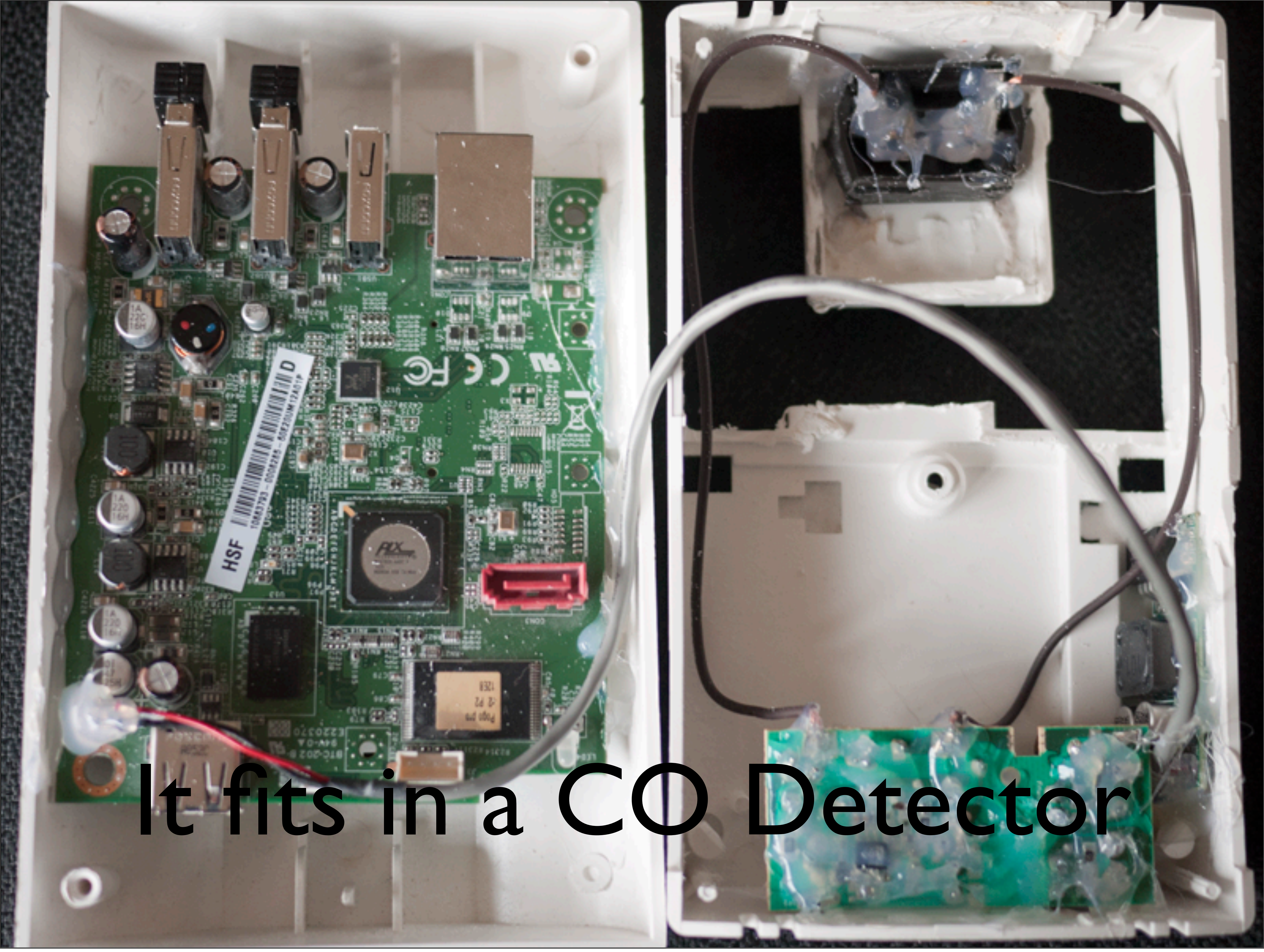
F-BOMB v.1 (ShmooCon 2012)

IV. System Architecture

A. Hardware: F-BOMB, version 2 (Falling/Ballistically-launched Object that Makes Backdoors)

1. Originally presented at ShmooCon 2012, but major advances since then; components now all off-the-shelf, significantly reduced volume, weight, power draw.

It fits in a CO Detector

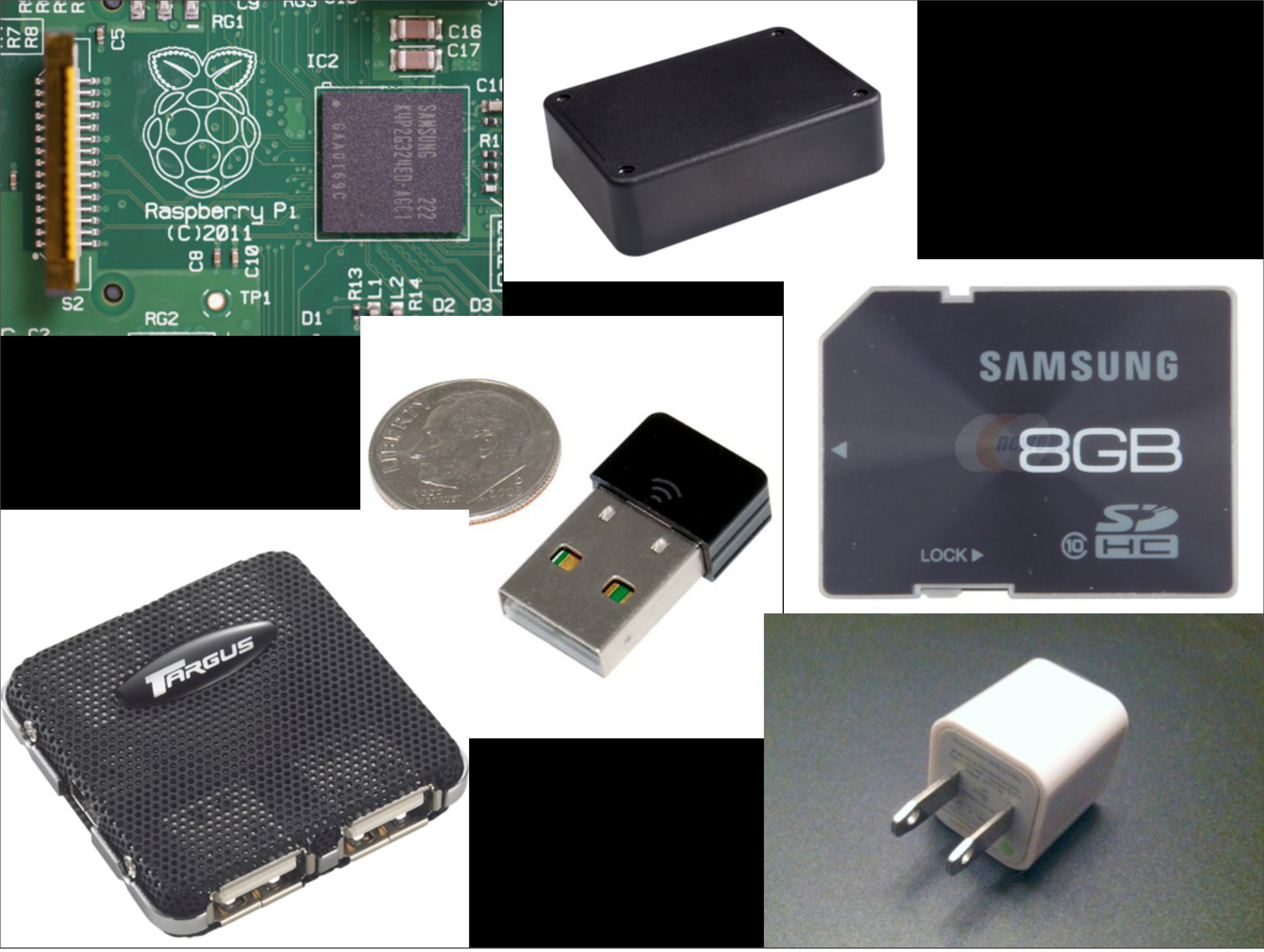


No one ever checks their CO detector to see if it has become a node in a sensor network.

The new one fits much better into this case; much less cutting is necessary.

F-BOMB v.2

2. Now based on the Raspberry Pi Model A, because it's awesome, runs an easier version of Linux (Debian vs. Arch), and I can actually get it for cheaper than the salvage PogoPlugs.



http://www.flickr.com/photos/gijsbertpeijs/7988257583
http://www.polycase.com/lp-51p
http://www.targus.com/us/productdetail.aspx?sku=ACH63US
http://www.amazon.com/JacobsParts-150Mbps-Wireless-Notebook-TP-WF11/dp/B0067NFSE2
http://www.newegg.com/Product/Product.aspx?Item=N82E16820147152
http://www.ebay.com/itm/10x-USB-USA-AC-Wall-Charger-for-Apple-iPhone-3-3G-4G-4S-5-5G-iPod-New-White-/271163372744

Raspberry Pi, Model A:	$25
Case:	$4.61
USB Hub:	$5.99
WiFi:	2x $6.52
SD Card:	$6.99
USB Power:	$1.45
Total:	57.08 per node

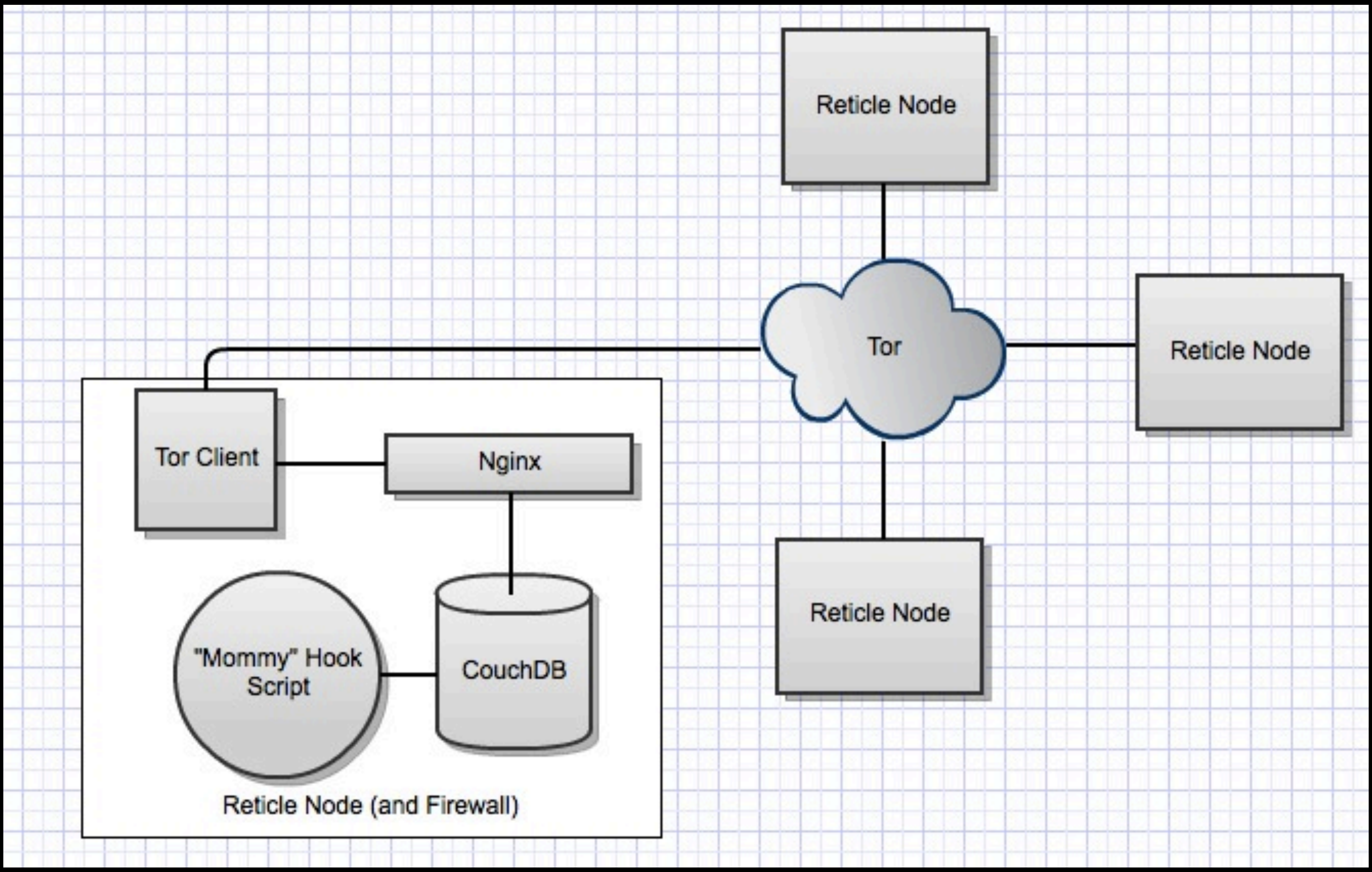
3. Per-Node Cost: $57.08 in 10-node quantities, excluding case.

a. I bought cheap wall-wart cases and used a drill saw; you can 3D print them, or even buy disposable GladWare and use that.

C&C Software

- “Reticle: Leaderless Command and Control”
    - This was the first of the two DARPA CFT contracts I mentioned
    - Whole presentation at B-Sides Vegas 2012---but I will summarize

Reticle



B. C&C Software: Reticle, Leaderless C&C

1. Developed under DARPA Cyber Fast Track, Spring 2012

2. Original work presented at BSidesLV 2012, but massive improvements, and a complete rewrite, since then.

3. Tor, CouchDB, Client-Side TLS; these combine for both encrypted communications and, critically, obfuscated communications gateways (via Tor Hidden Service---there's no indication of where the command node is). (Details on Reticle architecture.)

4. "Contagion Network"---all nodes know all things, commands can be inserted by any node. PKI, commands must be signed.

5. Disk encryption using grenade methodology to prevent tampering; "Pull Pin, Throw at Enemy" for disk encryption UX.

6. CreepyDOL, on the node side, is just a mission Reticle runs; it can be retasked at any time.

PortalSmash

It clicks on buttons, so you don't have to
https://github.com/ussjoin/portalsmash

This is how we get connectivity in every place: use local wifi. Reticle had some scripts to take care of this, but during its rewrite, I pulled them out so that this would work in more of a general case---because weirdly, I couldn't find any examples of doing this cleanly.

CreepyDOL

So as I mentioned, Creepy

Distributed Computation for Distributed Systems, and Centralized Computation For High-Level Data

http://www.flickr.com/photos/lara604/3164622774
http://www.flickr.com/photos/jenniferboyer/52474490/
http://www.flickr.com/photos/gaelx/1858599144

A. Distributed Querying for Distributed Data

1. Since we don't have independent, high-bandwidth channels for sending data home, it's not a good idea (and may not be possible) to send raw packets home. Nodes should send home data that's already been digested.

2. So: we run any queries on the nodes that can be effectively run on the nodes, *given data that node has collected*.

3. We do not process multi-node data on individual nodes, even though every node has access to all the data (see "contagion network"), because they've got limited processing power---and more importantly, data storage.

B. Centralized Querying for High-Level Data

1. Things that need datapoints from multiple nodes---tracking, pattern analysis, etc., go on the "backend."

2. The backend is just another node, but with a special mission configuration: rather than just sensing and adding data, it receives data from the contagion network, pushes it into another system (a data warehouse), and then instructs the contagion to delete it to make room.

NOM: Nosiness, Organization, and Mining

http://www.flickr.com/photos/scjody/5345366096

C. Data Query Methodology: NOM

1. O: Observation. Take as much data out of local traffic as possible; this means names, photos, services used, etc. To make this easy, we've created a large number of "filters" that are designed for traffic from specific applications---DropBox, Twitter, Facebook, OKCupid, etc. This is a distributed query (run on the nodes).

2. N: Nosiness. Using data extracted from O queries, there are lots of leveraged queries we can make; for instance, given an email address, we can look for accounts on web services, or given a photo, we can look for copies of that photo pointing to other accounts. This can be run either as distributed or centralized.

3. M: Mining. Taking data found by the nodes, build up larger analyzed products. For instance, is the device (person) usually in one area during a certain time of day? Are there three devices that are almost always seen together, if at all? (The latter may indicate that they are all carried by the same user.) This type of query is exclusively run on the backend.

Visualization

- Second DARPA CFT Contract
- Used the Unity Game Engine
  - Side note: wow, that's a fun toy
  - Side note: wow, I hate writing JavaScript that's interpreted by C#, then compiled into .NET CLR
- Runs on an iPad! Or OSX/Windows/Linux/Android
  - I think I could make it run on an XBox360, actually (Unity is Very Nice)

Want to see what it looks like?

(Say yes)
(Please)

Test Parameters

- To prevent badness, we programmed the NOM system to look only for traffic from devices we owned; **no “random stranger” data was collected at any time.**

Results

Scaling Up

- Sharding Contagion Networks
- Scaling backend --- luckily, this isn't hard
- Scaling limits of visualization
  - Frame rate...
- Hey, aren't there nice $20 SDRs? I wonder what I could sniff with that....
  - (Yes, those work on a Raspberry Pi)

Other Applications

Counter-Infiltration

http://www.flickr.com/photos/igalko/6341182132/

A. Counter-Infiltration

1. There is a persistent rumor, in cases of exceptional police brutality (Occupy Anything, or more protests in Britain) that the police are sending in agents provocateur to cause the disruption that gives them an excuse to crack down. (This rumor is at least 300 years old, by the way.)

2. CreepyDOL would let you set up "known devices" with alarms for new ones, watch as new people come in, or even simply set off a klaxon if a Blackberry shows up (obviously a cop).

C. OPSEC Training

1. The ROE for my tests demonstrate limiting data capture to one or several known devices. Use that to test your agents' OPSEC capabilities: set up a wide-ranging capture network (but tied to their stuff) and see what they leak.

2. The advantage is that you don't need to control every network an agent accesses. This lets you test "in the real world," which is much more realistic.

Evidence Logging

http://www.flickr.com/photos/decade_null/142235888

B. Evidence Logging

1. Again in fast-moving scenarios like protests and rallies: there's a real problem with destruction of evidence, electronic or physical, during crackdowns. In addition, it's very, very difficult to know who was *in* a kettle in the first few hours afterward; a way to know that could be very comforting and/or helpful to those outside.

2. Since CreepyDOL uses a contagion network, anything it logs will be immediately shipped out of the area to linked nodes anywhere on the planet. If those nodes go offline, the data is preserved.

3. For bonus points, use F-BOMB belt packs (which last a very long time on batteries) to have moving logs---and if you come in range of a WiFi AP somewhere (say, at a stop light), they'll offload their data without any additional interaction.

4. The encryption, and the fact that the nodes don't persist their keys, mean that unless an adversary *already knows what it is and how to cold boot it*, they don't get data. If people on the outside are concerned about the nodes, revoke their device certificates and they'll be cut off immediately.

Mitigation: A Sacrifice

VIII. Mitigation

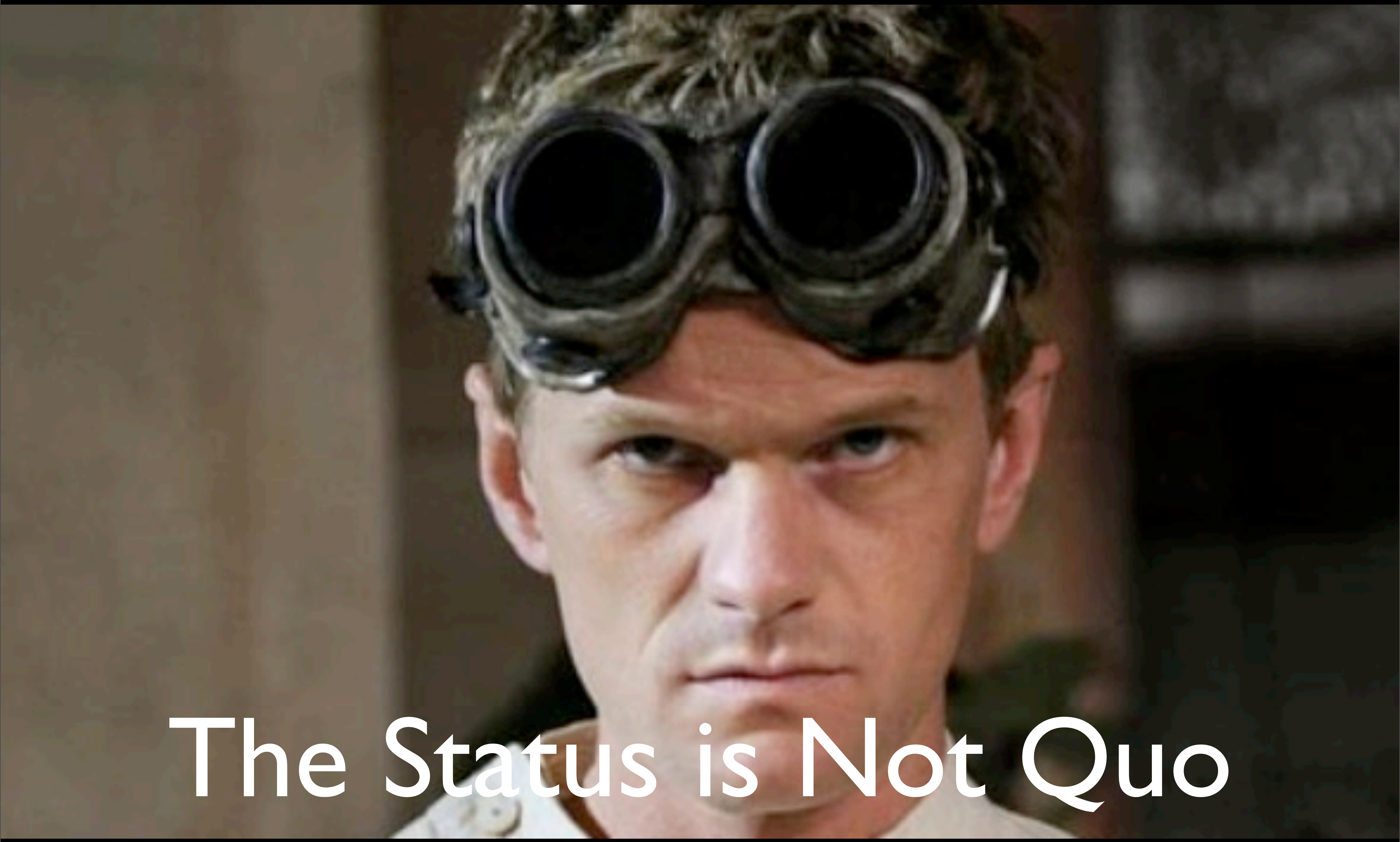
A. Much of the functionality CreepyDOL exploits is the 802.11 protocol itself; it's not as simple as "patch this," because the networks rely on it.

1. SSID Beaconing
2. MAC Address signing

B. What do you want to give up?

1. Always-on---turning things off would help.
2. Open WiFi---this would somewhat mitigate (harder to steal app data), but we can still see your MAC, and from this get your location. If we can otherwise associate a MAC with a name, we can still correlate multiple devices per person, and track patterns.
3. It isn't helpful to just use cellular data; given that SDRs now cost $20, we could cheaply extend CreepyDOL to watch for that (at least for MAC tracking purposes).

We don’t need the extra features of these protocols, but they’re nice... and it’s the status quo.



The Status is Not Quo

Image from Dr. Horrible's Sing-Along Blog, by Joss Whedon

Thanks!

- To the CFT program, without which I couldn't have spent a large part of law school doing something much more interesting
- To all those I've harassed for comments on proposals, work, or slides
- Ping me: http://www.maliceafterthought.com