Security is solved and we've done everything right

A modern fairy tale

Intro

- Various OSS SW & HW
- Kismet wireless sniffer
- Previously worked on enterprise Wi-Fi systems and designing secure Android devices

We’ve been here before

- If you’ve been to Sharkfest before you may have heard me rant a bit on security
- Here we are again
- Did we learn any lessons since last year?
- Hint: If there’s a question in the headline, the answer is “no.”

This matters for *everyone*

- Your attack surface now isn’t what it was a decade ago
- Attackers today aren’t who they were a decade ago
- Your data may not even be where you think it is
- The vulnerable areas and people of your organization may not be what you expect

Old attackers

- Deface web pages
- DDoS services to prove a point
- A little light credit card theft

Kiddies

- From the beginning of computing through the late 2000s, exploits / bugs were often freely published
- No monetary value
- Update cycles were slow
- Seen as normal bugs
- “Script kiddies” would hack systems using public bugs w/out knowing how they worked

Griefers & Defacement

- Attacks into a system were often to deface the website
- Anonymous routinely targeted groups and individuals for harassment
- DDOS was relatively rare, but effective when used
- Toolkits like “Low-Orbit Ion Cannon” commoditized attacks that were publicly known but commonly unpatched, “no-skill” hacking

Exploratory strange things

- First hints of larger botnets
- Code Red and SQL Slammer super-fast spreading worms started hitting networks
- Bank / Fortune 500 / Defense still faced different threats

New attackers

- Some element of the old groups are still around
- But when was the last time you saw a web page get defaced after a break-in?
- There’s a lot more money and new attackers out there now...

Hacking for profit

- Exploits have significant value now
- Browser / Java / Flash / etc exploits are worth hundreds of thousands of dollars
- They won’t get burned just to deface a website…
- But it’s not like development has stopped on them, either, so what’s going on?

Payment processing intrusion

- Major financial incentive
- Target, Home Depot, other major national stores compromised
- Heartland credit card center
- Both the front-end (cash register) and back end (centralized payment processing systems)

Identity compromise

- Ever applied for, or held, US Clearance? (Don’t raise your hand)
- Ever used Ashley Madison? (Again, don’t raise your hand)
- Ever had a credit rating?
- Your PII is already gone - All SF86 data exfilled from OPM, all accounts from Ashley Madison

Why steal identities?

- What’s the value of an identity?
- Financial opportunities to open credit, mortgages, etc
- Burner card, max it, open a new one under a new ID
- The US uses permanent ID numbers (SSN) that can’t be changed after theft so stolen data holds value over time
- There’s a more sinister option...

Extortion

- That was an interesting 3 examples, wasn’t it?
- Hundreds of thousands of people with access to secure/secret information
- Who may have used the same email to sign up for a cheating website
- And might have financial troubles?

Let that sink in...

So who’s doing these hacks?

- Get your tinfoil hats ready!
- Intelligence agencies ***are*** collecting information on targets via hacking; contractors and those w/ clearance *are* targets
- This isn’t just on the national security level, either
- Personal and work lives are intimately mixed now and present excellent extortion targets

I’m sorry...

Nationalistic hackers

- Rise in nation-state actors
- “Pariah” states such as North Korea and Iran have little to lose
- You can’t prosecute someone when you can’t extradite them
- US consistently files warrants against groups from Russia, Iran, etc - just means they can’t travel as easily now

War and manipulation in the cybers

- Attribution is *really hard*.
- No standard for what elicits a “kinetic” response
- No standard for how to deal with a nation already under sanctions
- Nation-state attacks today are the s’kiddy attacks of tomorrow, techniques spread

State exploits

- *Eternal Blue* leaked as part of Shadow Brokers dump
- Possibly originated with NSA
- Quickly adapted into a highly wormable ransomware, WannaCry
- Possibly further adapted by Russia and North Korea for Petya, nPetya, and other malware frameworks

Iran counter-hacks

- Documented rise in hacks since US withdrawal from the Iran nuclear deal
- Combination of nationalistic retaliation and a declaration of capability
- Often targeting industrial control systems, electrical grids, municipal systems
- “Bomb us and you don’t have clean water”

North Korean hacks

- Used to make political statements (such as of Sony for instance to interfere with release “The Interview” about NK)
- Used to raise capital - bitcoin mining, ransomware, and hacking/theft all attributed to NK to bypass currency sanctions
- Espionage, info gathering

Kaspersky

- Well known antivirus company
- Recently banned from US government sales
- Compelling evidence they may be used by the FSB to mine top secret documents and corporate internal information, via AV sensors
- Unknown if complicit or unwitting

Kaspersky + “Slingshot”

- Kaspersky recently released detailed information on “Slingshot” malware
- Advanced router malware found in thousands of coffee shop networks and SOHO routers in the Middle East and North Africa
- Kernel-mode, malware insertion, monitoring extensions

Unfortunately...

- “Slingshot” was an *active* US JSOC/SOCOM operation
- Targeting Islamic State
- Where did the Kaspersky samples come from?
- Possibly retaliation for banning sales in US?

So where’s the line

- When nations are willing to sling exploits and publicly spoil intelligence ops...
- What’s the line before “kinetic response”?
- Where does that leave all of us caught in the middle?
  - Home router botnets
  - Industrial / water / etc control
  - Business data
  - Personal / Social media data

Leveling the playing field

- Strong resemblance to terrorism
- Allows weaker actors to inflict significant damage on stronger actors
- The more industrialized the target, the more pain they feel
- Muddies the attribution waters - Was it China, pretending to be Russia, pretending to be North Korea?

Threats you'll actually see

"In the real world, threat models are much simpler. Basically, you're either dealing with Mossad or not-Mossad. If your adversary is not-Mossad, then you'll probably be fine if you pick a good password and don't respond to emails from ChEaPestPAiNPi11s@virus-basket.biz.ru. If your adversary is the Mossad, YOU'RE GONNA DIE AND THERE'S NOTHING THAT YOU CAN DO ABOUT IT. The Mossad is not intimidated by the fact that you employ https://."

- James Mickens

Zero-Day? Probably not today...

- In the “good old days” new exploits dropped on mailing lists with regularity
- Chances of you getting hit with a new attack were definitely non-zero

But now...

- Serious value to new exploits
- Chances are *you will not get hit with an unknown attack*
- **Caveat:** If you do business with government agencies, are a large bank, or a Fortune 50, your risk profile is totally different.
- However, once an exploit is public...

Very short lifecycle

- Software doesn’t care how much money it took to write
- Once someone gets a copy of a new exploit, commoditizing it becomes a very short race
- Major recent events have been *known* and *patched* vulnerabilities being re-used

Wannacry

- Wannacry infected over 400,000 systems
- Derived from stolen NSA payload, “Eternal Blue”
- *It was patched months before Wannacry hit*
- Every system hit by Wannacry *could have been protected* with just the standard Patch Tuesday updates!

Petya

- Additional variant of Eternal Blue
- Happened significantly *after* Wannacry (which, remember, was *after* official patches!)
- 17,000+ systems, most in Ukraine
- Possibly politically driven

Not Petya (aka nPetya)

- Significantly modified and more aggressive than the original, spreading using additional exploits
- *Looks* like Petya ransomware
- But isn’t - now just scrambles infected systems
- Again in eastern Europe, hitting major shipping company Maersk and other Ukraine businesses

Office, Flash, Java

- All have been hit with high-profile vulnerabilities
- All have been patched
- Most have been patched *before significant spread of the exploits*
- Millions of systems still hit

System libraries

- Equifax got owned
- Ever had credit in the US? You might be affected
- Point of entry was a bug in Apache Struts
- Owned within *days* of the bug becoming public
- Didn’t discover they were owned for *months*

Worse after public?

- Why are things worse *after* the patch is available?
- The attack now has a limited lifetime
- Attack is publicly known
- Patches can be reverse engineered to generate exploit
- … or the exploit was already public, resulting in the patch
- No reason *not* to burn an exploit now that it’s going away!

So what needs to get patched?

- Well… everything?
- But what is everything?
- Do you even know your attack surface?

“Where to attack”

- Public facing resources
- Poorly secured parts of the network
- Poorly updated devices
- High-profile employees

The most secure device… is the one you can’t talk to...

But that’s not super useful

- We generate, consume, sell, and buy information
- Everything is connected now
- Everything has attack surfaces
- Everyone has an online presence
- Everyone brings their own devices to work, too

Until about 2009...

- Corporate supplied the laptop
- Corporate maybe supplied a pager
- Centrally provisioned and managed laptops
- WPA-TLS, Radius, Active Directory are all easy with a central corp provisioned system

But now...

- BYOD rules
- Even company laptops used at coffee shops, airport networks, and so on, far outside your security perimeter
- Even if you don’t allow BYOD…
- … You probably give email, etc to phones
- Now your threat perimeter includes iOS, Android, home Wi-Fi...

Problems with BYOD

- Can’t enforce software (usually)
- Can’t control what else a user does
- May not be able to enforce updates
- Users may not apply updates
- Updates might not *even exist* (Looking at you, Android)

We don’t need no stinking administrator

- With a BYOD or a corporate laptop, administrator is bad…
- But the normal user owns all the files your company *really* cares about
- Anything which gets code execution that lets it read the users email, saved documents, etc...

Ingress to network

- Users are excellent ways to get ingress into the corporate network
- Where all the really juicy stuff is, assuming it’s not in an inbox and dropbox and google drive, of course
- Lots of poorly defended stuff *inside* the network

Lots of ways to get access to users

- Phishing (generic or targeted)
- Social engineering (Facebook, Linked In, etc)
- Re-used credentials (Did an employee use the same login on Ashley Madison as their corporate network?)
- Devices outside your network perimeter are juicy targets

Users have...

- SSH keys to log into production systems
- Commit access to code repositories
- Confidential documents
- Payroll and other PII, depending on their department

So now they’re in...

What happens next depends...

- Ransomware has crippled hospitals, shipping companies, and even entire city infrastructures (Atlanta)
- If you handle banking or other finance, you have a *big* problem now
- If you're lucky they'll just install cryptocurrency miners
- What else is on the inside of your network?

Industrial Control

- Industrial control systems (or “IOT before IOT was ‘cool’ “) can control lights, heating, manufacturing, power generation, etc
- If you’re in the manufacturing business, you care
- If you’re in the chemicals business, you care
- Industrial systems very rapidly get into state actors

2010, Stuxnet

- No official admission, likely USA + Israel
- Spread via infected USB drives and internal networks throughout the middle east
- Ultimate payload was to physically damage a specific type of centrifuge
- … Used by Iran for nuclear materials enrichment

Possibly the first

- Possibly the first, or at least, publicly known, “cyber weapon”
- Goal: Disabling a weapons program without provoking a “kinetic” response and a war
- Also opened the door for more international hacking...

2012, Saudi Aramco

- Shamoon virus
- Attributed to Iran
- Data destruction; wiped tens of thousands of systems
- Retaliation for Stuxnet / Flame
- Oil production network separate from office network prevented spread into industrial systems

2018, Tasnee Petrochemical

- August, 2017 - systems at Tasnee and related petro refineries experience problems
- Initial attack wipes drives and leaves political messages
- Actual payload targets Schneider Electric Triconex devices

Boom

- If successful, attack would have disabled safety systems
- Goal was physical destruction of plant
- *Advanced* malware could map internal control network and report back to select targets
- A bug in the code prevented the exploit from working - instead caused crashes

What else uses systems like this?

- Oil refineries
- Water treatment plants
- Nuclear reactors
- “You must reboot your reactor to complete the upgrade…”

The “S” in IOT is for “Security”

- Why do I like to pick on IOT & SCADA so much?
- Multiple industries who traditionally do not make online devices, suddenly connecting things online
- Years of security best practices go unknown
- Motley selection of hardware
- *Everything* works *against* having a secure platform

What’s in a controller/IOT system

- What’s inside these things anyhow?
- Controller / Processor
- Network interface or radio
- Some sort of network stack
- Some sort of RTOS or traditional OS

Microcontroller

- Lots of basic microcontrollers from TI, Microchip
- Typically found in low-functionality or low-power devices
- Typically running some sort of RTOS
- Very limited processing, ram, and complexity

ESP

- The darling of Wi-Fi connected devices
- Competent processor
- Does all basic Wi-Fi modes
- $2.50 on E-Bay for hackers/makers, even cheaper in bulk for device makers
- Found in light switches, outlets, and battery devices

Or it’s just Linux

- Linux runs on pretty small systems
- Self-contained system-on-a-chip contains MIPS processor, RAM, Wi-Fi, Ethernet, USB, etc
- Anything that can run OpenWRT runs Linux
- Some “industrial” and “commercial” devices are just a RPi or a Beaglebone shoved in the works

Time travel

- Messing with IOT/SCADA, especially wireless enabled ones, is like having a time machine

But not hippy friendly time travel

- We get to go back to the 1980s and 1990s security model
- But with all the tools and knowledge we have about security today
- And the commoditization of tools is *amazing*

We don't learn

- We have serious problems retaining knowledge
- Even more problems sharing that knowledge across industries
- Every time a group decides to implement from scratch they’re liable to make the same mistakes made a dozen times before

Crossing industry

- Does a car manufacturer know about hardening networks from intrusion?
- Does a valve manufacturer know how to prevent hardware glitching attacks?
- Does a company who shoves a raspberry pi inside an industrial controller know how to issue updates for kernel vulnerabilities?

Mitigations we've made get ignored

- We’ve gotten pretty good at mitigating a lot of things
- As bad as it is today, it *was* worse
- Major operating systems get regular security fixes
- We have mitigations like ASLR and non-executable pages
- … but simple controller devices don’t have *any* of those.

Buffer overflows

- One of the most well-known and often simplest attacks
- If you don’t check the length of a buffer and write too much data into it, you overwrite other things
- Do it right and you can control how that function works and call other code

ASLR

- Address space randomization
- Basically moves parts of the OS, system libraries, and kernel functions around randomly
- Makes it much, much harder for exploit code to run when it doesn’t know where functions it needs are

Non-exec pages

- Pages of memory can be flagged as “code” or “data”
- “Data” can’t be executed
- Buffers should typically live in “data”, so even if you manage to overflow, you can’t execute it

But we don’t have these

- But these don’t exist on small RTOS systems
- Or microcontrollers without memory security, virtual memory, etc
- No MMU means no protected memory
- Finding an overflow on these can mean full execution, trivially

Long term known vulnerabilities

- I said before the biggest risk comes from *known* vulnerabilities
- Do you have any Wi-Fi controlled light switches, etc?
- Have you ever gotten an OS update for them?
- How would you even apply an update?

Flashing is hard

- Upgrading microcontroller devices is hard
- It’s not like you can just boot into safe mode if it goes wrong
- Assuming whoever made it is the same person who sold it
- And even cares about supporting it anymore
- And if there’s even a way to flash it w/out a programmer!

Attacking the hardware

- The great thing about hardware…
- Is you can take it apart
- It’s very, very hard to secure something against someone physically tampering with it

If all the hardware is the same...

- The great thing about mass-produced hardware…
- Is it’s all the same
- Figure out how to crack into one, you can probably crack into them all
- Especially if there are hardcoded credentials!

Security enclaves

- Some hardware has enhanced security options
- Can be considered a TPM, a “TEE”, or a “Secure Enclave”
- Even making its way into modern Intel processors
- Additional tech costs money tho
- Rarely found on cheap hardware
- Commonly found on phones

External threats

- Embedded devices are most likely on your network
- They might *also* present an unencrypted configuration network
- They might punch through your firewall via uPNP, etc
- Insecure, un-updatable systems bridging into your private network?

Internal threats

- Even if they don’t communicate with the outside world at all…
- Once an attacker is inside your network, everything is fair game
- Mixing office and control networks is a *real* bad idea

“Infinite Time” attacks

- If an attacker can buy the same model hw
- And spend an “infinite” (or at least, effectively unlimited) amount of time with it…
- They’ve got as much time as they want to spend trying to break in, completely unobserved

Hardware vulnerabilities

- When you control the hardware there’s a whole host of attacks that can be performed
- Even if attempts are made to thwart them
- … rarely are attempts made.

Timing attacks

- Being able to finely measure time while talking to the hardware opens lots of possibilities
- For example, comparing a password
- Lets look at some pseudocode...

A hypothetical

```
For (x in length(password)):
   If (password[x] != saved_password[x]):
   Return
```

- How broken is this?
- Knowing we’re talking about timing attacks...

Timing, timing

- Naive password compare lets you know how much of the password you got right
- Same can happen with crypto
- And certificate validation
- Any secure compare MUST BE constant time
- Hacking like the movies - spin through knowing when you guessed the right character!

Power analysis

- Different CPU instructions take different amounts of power
- Different loops of crypto functions take different amounts of power
- Provable methods for deriving keys via power analysis

Even complex systems

- Power analysis can be used even against complex systems
- Modified power adapters can monitor phone power use
- Private keys derived
- Still want to plug into random chargers while travelling?

Power glitching

- If you control the power to the chip you can cause “things” to happen
- Processors expect consistent, stable, clean power feeds
- Combines power analysis and timing analysis
- Knowing when the processor does things, you can then attack it

BadFET and firmware locks

- Many microcontrollers prevent reading the firmware
- Processor boots
- Firmware request is set
- Checks if the “prevent read” flag is set
- Sets register to ‘false’
- Firmware read denied

But that needs power

- Setting registers takes power
- What happens if you undervolt the processor just when the register set happens?
- Register remains at previous value
- Firmware can now be read

Scale up to modern processors

- Not just embedded microcontrollers
- Modern processors have advanced power control
- Some also share execution with the secure environment
- ARM TEE found on Android is a special mode of the primary CPU

Crypto corruption

- Undervolting the CPU via power management APIs causes glitches
- Some crypto algorithms are exceptionally vulnerable to glitches
- Reduces complexity from “quadrillions” to “thousands” of guesses, OR corrupts prime factors and compromises entire keystream!

Attacking the Enclave

- Arm TEE vulnerable for example
- Allows extraction of key data from the secure enclave
- Allows injection of “signed” code by corrupting the signing keystream
- Arm TEE, Intel SGX
- No fix

Rowhammer

- Physical-layer attack against how RAM works at the electrical level
- Basically by flipping one part of ram between 0 and 1 very quickly you cause errors in other parts
- Flip ‘permission denied’ to ‘OK’

OS mitigations

- Operating system tries to prevent setting RAM quickly in ways that cause this effect
- Tries to not put security sensitive data next to pages of RAM user space can control
- How janky is this?

Keeps going deeper

- Rowhammer has been shown to work via ***javascript***
- Via OpenGL and graphics cards
- Via 10GBe network cards

Spectre, Meltdown, and all their friends

- Modern processors want to do a lot of things at once
- And they like to guess what things are right
- Many classes of information leakage derive from this
- Intel is struggling to fix it, but also impacts many other chips doing speculative execution
- Certainly not going to get solved in small, cheap devices!

Crossing attack surfaces

- Just like with user devices
- Once you get a foothold on the network from a vulnerable system
- Lots of other systems likely to fall

Vegas High-Rollers

- Casino in Vegas lost database with details of the richest (or at least, most prolific) clientele
- Network compromised via smart thermometer in the lobby fish tank
- Did *not* get casino gaming network - because it is *super* segregated and regulated!

Unencrypted traffic

- Many sensors either use no encryption
- Or offer an unencrypted side channel
- Unencrypted connections to servers to push data can be used to exploit vulnerabilities in the servers themselves

uPNP

- uPNP is used to map public ports to internal devices
- Can, when not restricted, completely compromise a firewall by allowing direct connections
- Remember how embedded systems like to use hardcoded credentials?
- Mirai botnet among others spread using default logins on tunneled hardware

Attacking hardware

- So you’ve got a physical device handy
- What can we do to it?

Easy: Firmware updates

- Pull the firmware update and dump it
- Run the ‘strings’ command against it
- Many firmware images, especially for Linux systems, contain full filesystem images
- ‘Binwalk’ is your friend!

Some basic firmware...

```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
512           0x200           LZMA compressed data, properties: 0x6D, dictionary size: 8388608
bytes, uncompressed size: 3556396 bytes
1177024       0x11F5C0        Squashfs filesystem, little endian, version 4.0, compression:xz,
size: 2281166 bytes, 1128 inodes, blocksize: 262144 bytes, created: 2016-02-02 11:51:44
```

```
dragorn@drd1812-25:~/Downloads$ binwalk tl-MR3020-webflash.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             TRX firmware header, little endian, image size: 3932188 bytes, CR
C32: 0xAC257ED4, flags: 0x1, version: 1, header size: 28 bytes, loader offset: 0x1C, linux kern
el offset: 0x0, rootfs offset: 0x0
540           0x21C           LZMA compressed data, properties: 0x5D, dictionary size: 8388608
bytes, uncompressed size: 2667760 bytes
983068        0xF001C         Squashfs filesystem, big endian, DD-WRT signature, version 3.0, s
ize: 2882672 bytes, 709 inodes, blocksize: 131072 bytes, created: 2013-03-25 08:35:57
```

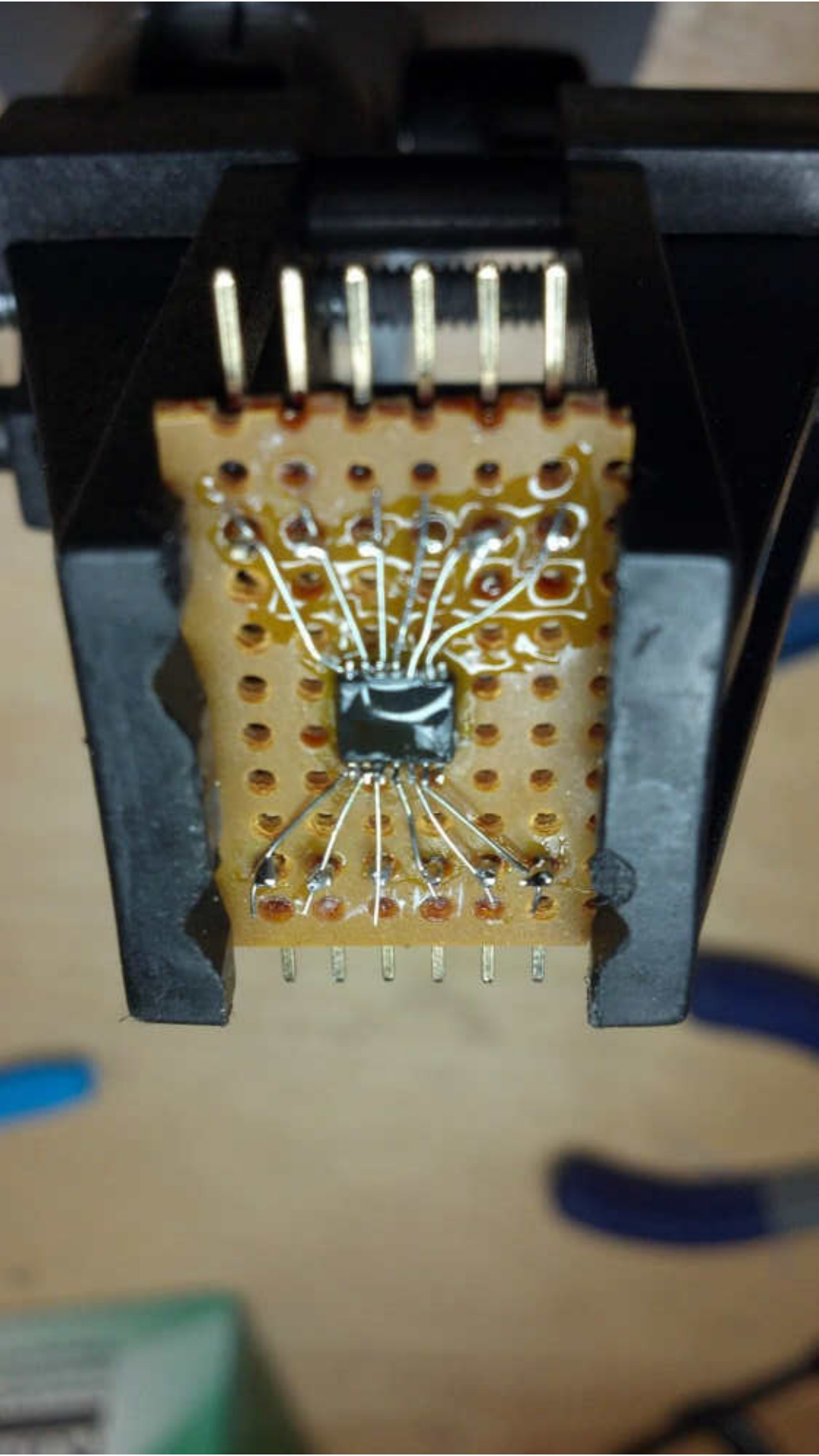
Random security camera

```
dragorn@drd1812-25:~/Downloads$ binwalk DH_IPC-HX1X2X-Themis_EngSpn_N_V2.620.0000002.0.R.170830
.zip

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Zip archive data, at least v2.0 to extract, compressed size: 375,
 uncompressed size: 2130, name: category.txt
445           0x1BD           Zip archive data, at least v2.0 to extract, compressed size: 520,
 uncompressed size: 5180, name: check.img
1032          0x408           Zip archive data, at least v2.0 to extract, compressed size: 392,
 uncompressed size: 904, name: CmdScript.img
1495          0x5D7           Zip archive data, at least v2.0 to extract, compressed size: 8547
0, uncompressed size: 86080, name: custom-x.squashfs.img
87044         0x15404         Zip archive data, at least v1.0 to extract, name: describe.txt
87114         0x1544A         Zip archive data, at least v2.0 to extract, compressed size: 1320
68, uncompressed size: 260160, name: dhboot.bin.img
219254        0x35876         Zip archive data, at least v2.0 to extract, compressed size: 2286
8, uncompressed size: 131136, name: dhboot-min.bin.img
242198        0x3B216         Zip archive data, at least v2.0 to extract, compressed size: 1374
4899, uncompressed size: 13743964, name: DH_IPC-HX1X2X-Themis_EngSpn_N_V2.620.0000002.0.R.17083
0.bin
13987214      0xD56D8E        Zip archive data, at least v2.0 to extract, compressed size: 1556
449, uncompressed size: 1558664, name: kernel.img
```

More difficult: Sniffing HW busses

- If you can open the device, you can tap communications busses
- Common protocols include I2C and SPI
- Very low power devices often communicate the encryption key in plaintext over the SPI bus to the radio!



Making sense of signals

- Logic analyzers are DIRT cheap now
- Built-in protocol decoding
- Saleae, SUMP, OpenLogic
- $100-$200

Expert level: FIB

- Insane pro level
- Focused Ion Beam
- Allows laying new traces *directly onto the silicon die*
- Has been used to hack TPM modules on gaming systems
- And the TPM/Smartcard module

Realistic threat?

- So is someone with a FIB a realistic threat?
- Not to 99% of the people here
- … but if you're in the defence contracting business
- … or you make millions of gaming systems which all lock the secret keys in a TPM
- *NOW* the profit balance changes for attacking your system

Cost to attacker when they fail?

- Profit balance: What are the risks of failure?
- If it’s a one-of-a-kind system, the costs are very high, which makes attacking you much riskier
- Evidentiary devices (terrorist iPhone, for example)
- Custom devices
- Devices that can’t be replaced w/out the victim knowing

But readily available devices...

- If the device is readily and anonymously available
- Even if it’s expensive, the profit for breaking it could be huge
- Gaming consoles are $400, iPhones are $1000, but an iPhone exploit can be worth $2,000,000

Supply chains

- Where does your hardware come from
- Do you use pre-install?
- Do all your hardware vendors protect THEIR supply chains?
- You’ve plugged an appliance in - what if it’s already owned, from the factory?

Android phones

- “RottenSys” found on over 5 million android phones
- Pre-installed from the factory
- Compromised system image process
- Oppo, Honor, ZTE, Archos, and dozens more, over 100 models
- Do you have an appliance using a built-in tablet or similar?

ZTE, Huawei

- Banned from sales to the US Govt
- Any company doing HW sales banned from sales to the US Govt
- Considered an extension of the intelligence branch of the govt of China
- Selling into the 5G mobile space as a backbone vendor

Counterfeit components

- At best they don’t work to spec
- Many don’t work at all
- At worst, they have hidden features
- Crypto counts on high-quality random data
- Compromise that randomness…
- What other HW components are in the supply chain?

Ghost shifts

- I used to work designing secured Android phones
- Heard about some “cheap” sales of our HW
- Turns out the manufacturer made 10,000+ extra phones
- Removed the security checks and pre-installed SW
- Intended to only sell in China below the radar
- Looks exactly like legit HW because it IS - just, also, not

Software supply chain

- How many libraries does your software use?
- Where do they come from?
- Do you update them?
- Do you use Docker?
- Where do your Docker images come from?
- Docker allows anyone to upload a base image for others to base their systems on

Pre-trojaned Docker Images

- Docker recently pulled 10-15 images
- Downloaded over ***1,000,000 times***
- Pre-installed with crypto miner software
- I’m ***sure*** there are more than 10-15 such images out there
- What else is preconfigured in them?

How far down the rabbit hole?

So it’s all pointless, right?

- In an absolutist sense, yeah, probably
- If you’re interesting enough to burn an 0day on, because you do national security, military, or enough money to pay for it, you’re going to lose
- So is it worth even trying?

“Never let the perfect be the enemy of the good”

- *Sun Tzu*

“Don’t trust every quote you read on the Internet”

- *Abraham Lincoln*

Every barrier to an attacker helps

- Every trivial problem you solve raises the cost of breaking into your systems
- Every user you train not to fall for a phishing attack saves you time, money, and public trust
- If you're *not* targeted by serious actors, you *can* get ahead of the smaller ones

Do you already have a security group?

- Do you already have a security group you can work with?
- Generally responsible for the public security stance of the company
- Often involved w/ developers, legal, networking
- If you have a CISO...

You need teeth

- If the management and C levels aren’t on board it’s not going to go well
- The goal should be *minimal* interference and *minimal* downtime for security problems…
- But when something goes wrong you need to be able to *fix* it

If you make a product...

- Define your vulnerability handling process
- Mechanisms to determine how severe a vulnerability is
- “Oh sh*t handles” for stopping development for *super* critical vulnerabilities
- How will you do hotfix releases?
- How will you notify customers?

If you manage systems

- How do you apply updates
- How do you audit that updates have been applied
- Do you know exactly what versions all your servers are deployed with?
- What about containers?
- Do you, and *can* you, force security status onto BYOD?

Proactive and effective security

- A customer will almost never congratulate you on *good* security practices
- But you'll get eviscerated when you fail
- Try to target *effective* solutions
- Perfection is impossible

Useful questions

- Was it discovered outside the company?
  - Anything reported to you
  - Anything in public components
- Does it require authentication to trigger?
  - Still bad, but less urgently bad
  - “Parking lot” / “Drive-by” attacks *super* bad
- What is the impact?
- Who are the most vulnerable users?

Not-so-useful questions

- “How hard is it to write an exploit”
  - Only takes one clever person to write one
  - See: Metasploit, most malware
- “How likely is it”
  - 0% chance of it happening, until it does
- “Is that really a security problem”
  - Crashes and “you can’t exploit this” bugs turn into “now it’s exploited” pretty damn quickly

Culture of security

- You don’t need to be an expert
- But knowing when you’re in dangerous territory lets you research and proactively defend!
- Encouraging everyone to take part in keeping an eye out for weird things happening

Vulnerable users

- Vulnerable users aren’t necessarily the ones with exciting job titles
- Anyone in HR - employee records, billing, access to sensitive systems
- Anyone in payroll - billing
- Anyone involved in finance / stock disclosure - stock scams / futures

CFO scams

- CFOs are, obviously, major targets
- Lots of attacks against small and medium business CFO or billing departments
- Initiate money transfer or grant access to company accounts
- *Protect your C-levels*

Educate without shaming

- Security is hard
- Constant vigilance is hard
- Let’s not pretend it’s all the users fault
- Would you rather spend 10 minutes helping someone determine if an email was legit
- … Or rebuild their entire system and try to roll back a $100k wire transfer?

Usable solutions

- The key is to maintain *usable* security
- If the security group is known for simply stalling all projects, people find ways to avoid the security group
- If the “solutions” make it impossible to work, people will find workarounds that bypass them

Effective… but not *usable*...

Ounce of prevention, etc

- So we know we can’t solve everything
- Solving *anything* is better than ignoring the problem
- What can we do about it?

Defensive sysadmin

- Know what your systems are running
- Know what traffic is normal
- Firewall, firewall, firewall
- RBAC / SELinux / cgroup restrictions
- Multifactor login tokens

Defensive network

- Difficult to maintain w/out tight link to server group
- Prevent unrestricted traffic between office, server, control networks
- Firewall, firewall, firewall

Defensive home network

- Use a reputable router!
- Make sure updates are applied!
- If you’re a Linux nerd, OpenWRT/Lede/DDWrt offer upgrade cycles outside of your router manuf
- *Very* good gear is cheap! Ubiquity Edgerouter is damn near enterprise quality for ~$200 or less

Going it alone

- Enable 2FA
- *Enable. 2. F. A.*
- Enable full disk encryption.
- *Be suspicious* of weird emails
- *Follow up via alternate methods* - does the email seem weird? *Call them* before giving a password or a money transfer

Terrible new scam

- Compromise the O365 accounts of real estate agents
- Set mail forwarding rules
- Man-in-the-middle (literally) communication with clients
- Change the destination of the escrow/downpayment transfer
- Bank isn’t liable because you told them where to send it
- Now you’re out your life savings

2FA for SSH

- Krypton - 2FA for generic OpenSSH
- Acts as a PKCS smartcard library to OpenSSH
- Uses an iPhone or Android app to store the private key in a HW enclave
- Requires authentication on the phone

Defensive programming

- Obviously this could be an entire talk all on its own
- Assume all data is hostile
- Assume anything that declares the length of a field is lying to you
- Triple-check your types
- Separate privileges
- Beware encoding & escapes!

Field lengths suck

- Say the protocol looks like:

  Unsigned int length;

  U_char data[length];
- But what if they only send half of that data?
- Make sure you *always* compare the length of the buffer to the claimed length in the data

Type failures

- Same example:

  Unsigned int length;
- What happens if you cast that to an int? Eventually it goes negative
- What happens when you reference buffers with negatives? Or copy negative-sized data?

Gotta keep ‘em separated

- You need root to read packets
- You don’t need root to decode packets
- You should do as little as possible as root
- Notice how wireshark has a capture daemon that talks to the UI?

Know what you’re talking to

- Every output system has its own set of special characters you have to obey
- Failure to do so leads to:
  - SQL injection
  - XSS cross-site injection
  - Shell command injection
  - Corrupted data/reports

Don’t do it yourself

- Whenever possible, don’t write an interface library yourself
- There are almost certainly nuances you don’t know
- Find a well supported, well used library for SQL/ORM, JSON, etc
- Always filter before you use supplied data!

Some more material

- James Mickens, security researcher - google him
- Risky Business podcast - http://risky.biz
- Brian Krebs - http://krebsonsecurity.com
- Defensive Programming Guide + [language] - google

Contact

- mike@kismetwireless.net
- @KismetWireless
- https://www.kismetwireless.net