Nick's Personal Self-Defense Decisions...

Putting CS161 in Context:
Nick's Self Defense Strategies...

- ***How*** and ***why*** do I protect myself online and in person...
  - ***How*** I decide what to prepare for (and what not to prepare for)
  - ***Why*** I've drunk the Apple Kool-Aid™
  - ***Why*** I use my credit card everywhere but not a debit card
- And then my slides on Nukes and Tor that I didn't get to...

My Personal Threats: The Generic Opportunist

- There are a ***lot*** of crooks out there
  - And they are rather organized...
- But at the same time, these criminals are generally economically rational
  - So ***this*** is a bear race: I don't need perfect security, I just need ***good enough*** security
- I use this to determine security/convenience tradeoffs all the time
  - So no password reuse (use a password manager instead)
  - Full disk encryption & passwords on devices:
    Mitigates the damage from theft
  - Find my iPhone turned on:
    Increases probability of theft recovery

My Personal Threats:
The ***Lazy*** Nation State

- OK, I'm a high ***enough*** profile to have to worry about the "Advanced Persistent Threats"...
  - Trying for a reasonably high profile on computer policy issues
  - A fair amount of stuff studying the NSA's toys and other nation-state tools
  - But only at the Annoying Pestilent Teenager level:
    I'm worth some effort but not an extraordinary amount
- So its only ***slightly*** more advanced than the everyday attackers...
  With one ***huge*** exception: Crossing borders
  - Every nation maintains the right to conduct searches of all electronic contents at a border checkpoint

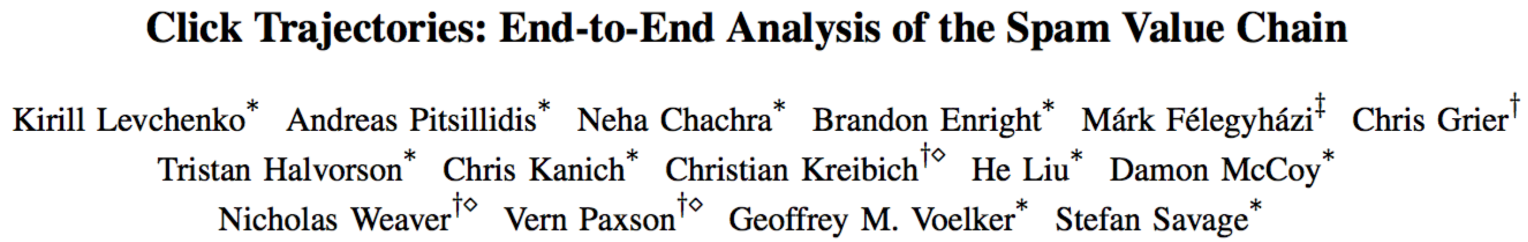
My Border Crossing Policy: Low Risk Borders

- Not very sensitive borders: Canada, Europe, US, etc...
  - I use full disk encryption with strong passwords on all devices
    - Primary use is to prevent theft from also losing data
  - I have a ***very robust*** backup strategy
    - Time machine, archived backups in a safe deposit box, working sets under version control backed up to remote systems...
- So, as the plane lands:
  - Power off my devices
    - Device encryption is only ***robust*** when you aren't logged in
  - Go through the border
- If my devices get siezed...
  - "Keep it, we'll let the lawyers sort it out"

High Risk Borders

- Middle East or, if, god forbid, I visit China or Russia...
  - Need something that doesn't just resist compromise but can also ***tolerate compromise***
- A "burner" iPhone SE with a Bluetooth keyboard
  - The cheapest secure device available
  - Set it up with ***independent*** computer accounts for both Google and Apple
    - Temporarily forward my main email to a temporary gmail account
    - All workflow accessible through Google apps on that device
  - Bluetooth keyboard does leak keystrokes, so don't use it for passwords but its safe for everything else
- Not only is this device very hard to compromise...
  - But there is very low value in ***successfully compromising it***: The attacker would only gain access to dummy accounts that have no additional privileges
- And bonus, I'm not stuck dragging a computer to the ski slopes in Dubai...
  - Since the other unique threat in those environments is the "Evil maid" attack

My Personal Threats:
The Russians... Perhaps



Click Trajectories: End-to-End Analysis of the Spam Value Chain

Kirill Levchenko[*] Andreas Pitsillidis[*] Neha Chachra[*] Brandon Enright[*] Márk Félegyházi[‡] Chris Grier[†]
Tristan Halvorson[*] Chris Kanich[*] Christian Kreibich[†⋄] He Liu[*] Damon McCoy[*]
Nicholas Weaver[†⋄] Vern Paxson[†⋄] Geoffrey M. Voelker[*] Stefan Savage[*]

- This is the paper that killed the Viagra® Spam business
  - A $100M a year set of organized criminal enterprises in Russia...
    And they put the ***organized*** in organized crime...
- I've adopted a ***detection and response*** strategy:
  - The Russians have higher priority targets: The first authors, the last authors, and Brian Krebs
  - If anything suspicious happens to Brian, Kirill, or Stefan, ***then*** I will start sleeping with a rifle under my bed

Excluded Threats: Sorta…

- Intimate Partner Threats…
  - But I've had at least one colleague caught up with that:
    If you have that, iPhone, period
- Aggressive Nation States…
  - $50M will buy the latest version of Pegasus malcode
- The US government…
  - The surveillance powers of the US government are awesome and terrifying to behold…

Passwords and 2-factor...

- Whenever possible, I always use the security key for 2-factor
  - Instead of me having to interrogate the site to determine phishing...
  - The site has to prove to the key it is legitimate!
- For passwords I always use a password manager
  - Yes, if an attacker compromises my computer, they can steal all my passwords...
  - But the same attacker can get all the passwords I actually use when I type them in (a 'keylogger').

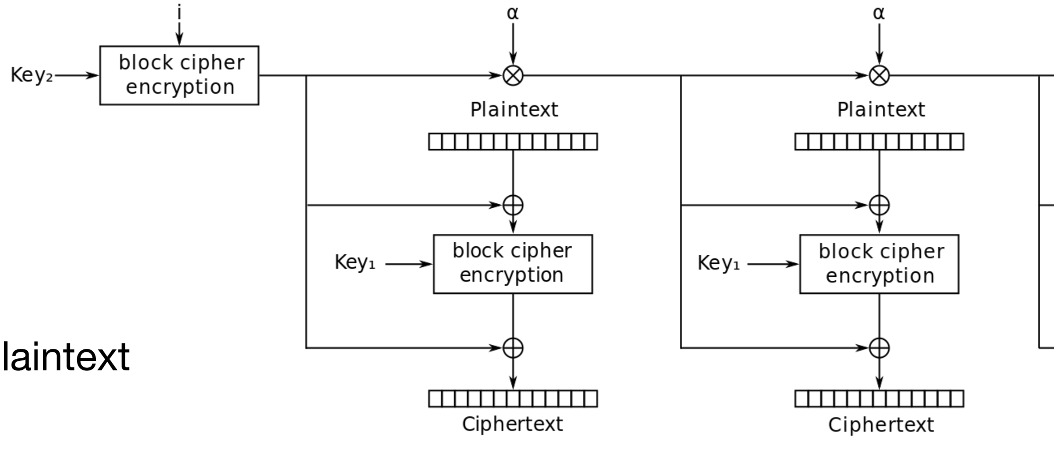
The Apple Kool-Aid...

- The iPhone is perhaps the most secure commodity device available...
  - Not only does it receive patches but since the 5S it gained a dedicated cryptographic coprocessor
- The ***Secure Enclave Processor*** is the trusted base for the phone
  - Even the main operating system isn't fully trusted by the phone!
- A dedicated ARM v7 coprocessor
  - Small amount of memory, a true RNG, cryptographic engine, etc...
  - Important: A collection of ***randomly*** set fuses
    - Should not be able to extract these bits without taking the CPU apart or compromising the Secure Enclave's software
  - But bulk of the memory is shared with the main CPU
- GOOD documentation:
  - The iOS security guide is something you should at least skim….
    I find that the design decisions behind how iOS does things make ***great*** final exam questions.

The Roll of the SEP...
Things ***too important*** to allow the OS to handle

- Key management for the encrypted data store
  - The CPU has to ask for access to data!
- Managing the user's passphrase and related information
- User authentication:
  - ***Encrypted*** channel to the fingerprint reader/face recognition camera
- Storing credit cards
  - ApplePay is cheap for merchants ***because it is secure***:
    Designed to have very low probability of fraud!
- Secure boot chain
  - Authenticates that the code is signed by Apple

AES-256-XEX mode



XEX mode encryption

- An ***confidentality-only*** mode developed by Phil Rogaway...
  - Designed for encrypting data within a filesystem block ***i***
    - Known plaintext, when encrypted, can't be replaced to produce known output, only "random" output
  - Within a block: Same cypher text implies different plaintext
  - Between blocks: Same cypher text implies nothing!
  - α is a galios multiplication and is very quick:

    In practice this enables parallel encryption/decryption
- Used by the SEP to encrypt its own memory...
  - Since it has to share main memory with the main processor
- Opens a limited attack surface from the main processor:
  - Main processor can replace 128b blocks with ***random*** corruption

User Passwords...

- Data is encrypted with the user's password
  - When you power on the phone, most data is completely encrypted
- The master key is PBKDF2(password || on-chip-secret)
  - So you need ***both*** to generate the master key
  - Some other data has the key as F(on-chip-secret) for stuff that is always available from boot
- The master keys encrypt a block in the flash that holds all the other keys
  - So if the system can erase this block effectively it can erase the phone by erasing just one block of information
- Apple implemented ***effaceable storage***:
  - After x failures, OS command, whatever...
    Overwrite that master block in the flash securely
  - Destroy the keys == erase everything!

Background: FBI v Apple

- A "terrorist" went on a rampage with a rifle in San Bernardino...
  - Killed several people before being killed in a battle with police
- He left behind a work-owned, passcode-locked iPhone 5 in his other car...
- The FBI ***knew*** there was no valuable information on this phone
  - But never one to refuse a good test case, they tried to compel Apple in court to force Apple to unlock the phone...
- Apple has serious security on the phone
  - Effectively everything is encrypted with PBKDF2(PW||on-chip-secret): >128b of randomly set microscopic fuses
    - Requires that ***any*** brute force attack either be done on the phone or take apart the CPU
  - Multiple timeouts:
    - 5 incorrect passwords -> starts to slow down
    - 10 incorrect passwords -> optional (opt-in) erase-the-phone

What the FBI wanted...

- Apple provides a ***modified*** version of the operating system which...
  - Removes the timeout on all password attempts
  - Enables password attempts through the USB connection
- Apple ***cryptographically signs the rogue OS version!***
  - A horrific precedent:
    This is ***requiring*** that Apple both create a malicious version of the OS and sign it
    - If the FBI could compel Apple to do this, the NSA could too...
      It would make it ***impossible*** to trust software updates!

Updating the SEP To Prevent This Possibility...

- The SEP will only accept updates ***signed by Apple***
  - But an updated SEP could exfiltrate the secret to enable an offline attack
- The FBI previously asked for this capability against a non-SEP equipped phone
  - "Hey Apple, cryptographically sign a corrupted version of the OS so that we can brute-force a password"
- How to prevent the FBI from asking again?
- Now, an OS update (either to the base OS and/or the SEP) requires the user to be logged in ***and input the password***
  - "To rekey the lock, you must first unlock the lock"
  - The FBI can only even ***attempt*** to ask before they have possession of the phone since once they have the phone they must also have the passcode
  - So when offered the chance to try again with a "Lone Wolf's" iPhone in the Texas church shooting, they haven't bothered
- At this point, Apple has now gone back and allows auto-updates for the base OS
  - (but probably not the SEP, and you may need to have the phone unlocked before the auto-update can trigger)

The Limits of the SEP... The host O/S

- The SEP can keep the host OS from accessing things it shouldn't...
  - Credit cards stored for ApplePay, your fingerprint, etc...
- But it can't keep the host OS from things it is supposed to access
  - All the user data when the user is logged in...
- So do have to rely on the host OS as part of ***my*** TCB
  - Fortunately it is updated continuously when vulnerabilities are found
    - Apple has responded to the discovery of very targeted zero-days in <30 days
  - And Apple has both good sandboxing of user applications and a history of decent vetting
    - So the random apps are ***not*** in the Trusted Base.

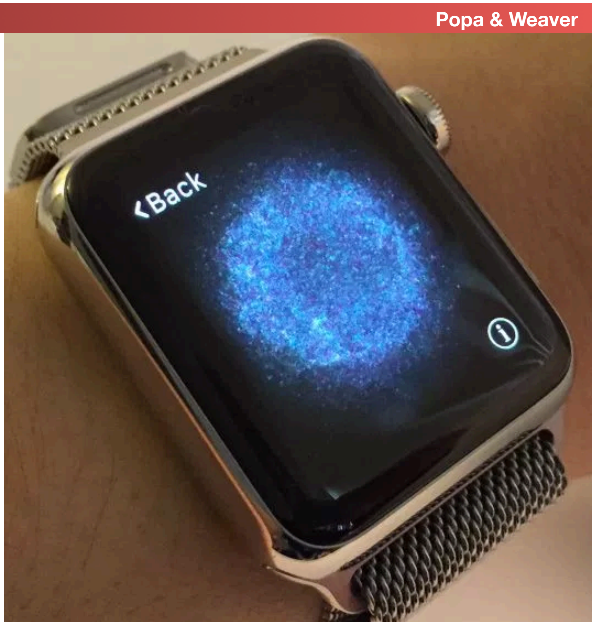
The SEP and Apple Pay

- The SEP is what makes ApplePay possible
  - It handles the authentication to the user with the fingerprint reader/face reader
    - Verifies that it is the user not somebody random
  - It handles the emulation of the credit card
    - A "tokenized" Near Field Communication (NFC) wireless protocol
    - And a tokenized public key protocol for payments through the app
- ***Very hard*** to conduct a fraudulent transaction
  - Designed to enforce user consent at the SEP
- ***Disadvantage***: The fingerprint reader is part of the trust domain
  - Which means you need special permission from Apple to replace the fingerprint reader when replacing a broken screen

I ***love*** ApplePay...

- It is a ***faster*** protocol than the chip-and-signature
  - NFC protocol is designed to do the same operation in less time because the protocol is newer
- It is a ***more secure*** protocol than NFC on the credit card
  - Since it actually enforces user-consent
- It is more ***privacy sensitive*** than standard credit card payments
  - Generates a unique token for each transaction:
    Merchant is not supposed to link your transactions
- Result is its low cost:
  - Very hard to commit fraud -> less cost to transact
- I use it on my watch all the time
- Useful product idea: Enable enrolling credit cards to enable "tap to open" door locks!

Transitive Trust in the Apple Ecosystem...



- The most trusted item is the iPhone SEP
  - Assumed to be rock-solid
  - Fingerprint reader/face reader allows it to be convenient
- The watch trusts the phone
  - The pairing process includes a cryptographic key exchange mediated by close proximity and the camera
  - So Unlock the phone -> Unlock the watch
- My computer trusts my watch
  - Distance-bounded cryptographic protocol
  - So my watch unlocks my computer
- Result? I don't have to keep retyping my password
  - Allows the use of ***strong passwords everywhere*** without driving myself crazy!

Credit Card Fraud

- Under US law we have very good protections against fraud
  - Theoretical $50 limit if we catch it quickly
  - $0 limit in practice
- So cost of credit card fraud for me is the cost of recovery from fraud
  - Because fraud ***will happen***:
  - The mag stripe is all that is needed to duplicate a swipe-card
    - And you can still use swipe-only at gas pumps and other such locations
  - The numbers front and back is all that is needed for card-not-present fraud
    - And how many systems
- What are the recovery costs?
  - Being without the card for a couple of days...
    - Have a second back-up card
  - Having to change all my autopay items...
    - Grrrr....

But What About "Debit" Cards?

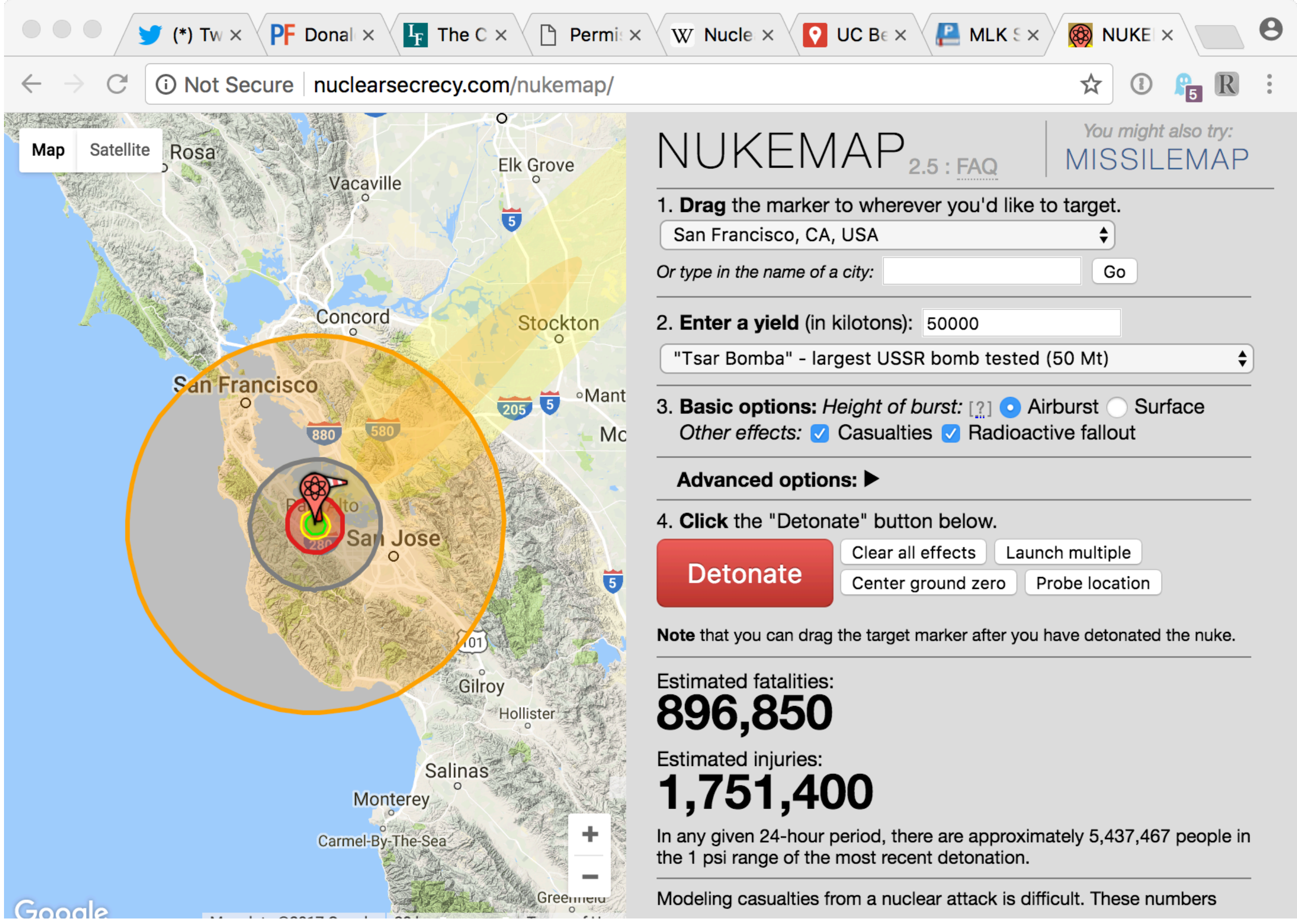
- Theoretically the fraud protection is the same...
- But two caveats...
  - It is easier to not pay your credit card company than to claw money back from your bank...
  - Until the situation is resolved:
    - Credit card? It is the credit card company's money that is missing
    - Debit card? It is ***your*** money that is missing
- Result is debit card fraud is more transient disruptions...

So Two Different Policies...

- Credit card: Hakunna Matata!
  - I use it without reservation, just with a spare in case something happens
  - Probably 2-3 compromise events have happened, and its annoying but ah well
    - The most interesting was $1 to Tsunami relief in 2004...
      was a way for the attacker to test that the stolen card was valid
- Debit card: Paranoia-city...
  - It is an ATM-ONLY card (no Visa/Mastercard logo!)
  - It is used ONLY in ATMs belonging to my bank
    - Reduce the risk of "skimmers": rogue ATMs that record cards and keystrokes
- It is about the cost of mitigating an attack:
  Which has a greater cost in terms of hassle?

Why talk about nukes?

- Nukes are big and scary and in the news...
  - But have interesting security and safety properties
- Lots of material ~~stolen~~ borrowed from Steve Bellovin's excellent talk on PALs



NUKEMAP 2.5 : FAQ

You might also try: MISSILEMAP

1. **Drag** the marker to wherever you'd like to target.

San Francisco, CA, USA

Or type in the name of a city: Go

2. **Enter a yield** (in kilotons): 50000

"Tsar Bomba" - largest USSR bomb tested (50 Mt)

3. **Basic options:** *Height of burst:* [?] ● Airburst ○ Surface

Other effects: ☑ Casualties ☑ Radioactive fallout

Advanced options: ▶

4. **Click** the "Detonate" button below.

Detonate | Clear all effects | Launch multiple | Center ground zero | Probe location

Note that you can drag the target marker after you have detonated the nuke.

Estimated fatalities:

896,850

Estimated injuries:

1,751,400

In any given 24-hour period, there are approximately 5,437,467 people in the 1 psi range of the most recent detonation.

Modeling casualties from a nuclear attack is difficult. These numbers

How a Nuclear Weapon Works...

- 1960s-level technology...
  - A hollow sphere of fissile material
    - Plutonium and/or Plutonium + Uranium
  - Use this as a primary to ignite a Teller/Ulam secondary to make it a hydrogen bomb...

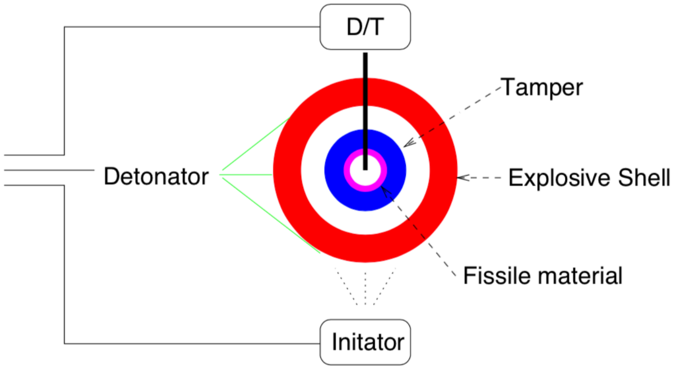
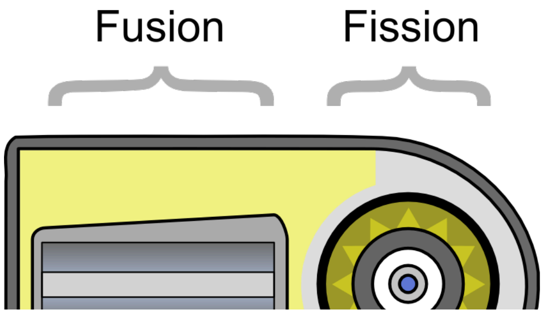


Diagram by Steve Bellovin

- Very careful sequencing needed
  - D/T pump to fill the hollow with Deuterium & Tritium ("Boost gas")
  - Initiator sprays neutrons to start the chain reaction
  - Detonator needs to trigger multiple points on the explosive shell
    - Squiggly-traces of explosive so that all around the shell everything detonates at once

And H-Bombs...

- A "Tellar/Ulam" 2-stage device: A A-bomb ignites a fusion stage
  - Fusion stage has Lithium Deuteride...
    - Neutrons and pressure from the A-bomb convert the Lithium to Tritium
    - Then Deuterium/Tritium fusion makes it go boom!
      - And sprays a crap-ton of neutrons around that increase the fissions as well
- Still 1960s technology!
  - Biggest issue overall is materials: 6 or 7 countries have built H-Bombs



And How To Deliver Them...

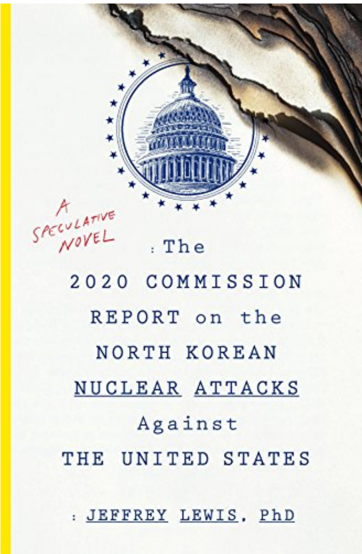
- Stick em on a rocket
  - This ***is*** rocket science: It is probably easier to build the nuke than build the ICBM...
  - Alternatively, stick it on an unmanned miniature airplane ("Cruise Missile") or just hang it under a plane as a old-fashioned bomb
- Then stick the rocket on something
  - In a hardened silo
    - But the other side can drop a nuke on it...
  - On a truck
  - In a sub
  - On a plane...

The Problem:
When To Use Nukes...

- Nuclear weapon systems can fail in two ways:
  - Launch the nukes when you shouldn't...
  - Fail to launch the nukes when you should...
- The latter is (badly) addressed by how our nuclear decision making happens
  - "Launch on warning": If we ***think*** we are under attack, the President has a couple minutes to decide to order a nuclear strike before the attacker hits our ICBMs!
    - This is often regarded as ***insanely*** stupid: We have both nuclear bombers with long-range cruise missiles and nuclear armed submarines, both of which ***will*** be able to launch enough retaliatory hellfire
  - Far better is the "French model" (cite @armscontrolwonk):
    "We have subs. You nuke us ***or*** attack our strategic weapons and we nuke you":
    - This removes the time pressure which can cause errors

"Launch on Warning" and North Korea...

- Let us assume that North Korea's leadership are ***rational*** actors
  - They act in what they perceive as their self interest: survival!
- North Korean leadership ***will eventually lose*** a war with South Korea and the US
  - So they may be provocative, but they want to make ***sure*** the US and South Korea won't start a war
- Nukes are a critical deterrent for them
  - Especially since Donald Trump doesn't seem to care that a war would kill hundreds of thousands in South Korea
- IRBMs and ICBMs are as important as the nukes themselves!
  - Need to be able to hit the US bases in Okinawa and Guam as military targets
  - And Mar-a-lago and Washington DC to dissuade Trump personally: The Hwasong-15 ICBM can just barely range South Florida.
- "***Empathy*** for the devil"
  - Computer security is adversarial, think about your adversary's needs, wants, and desires

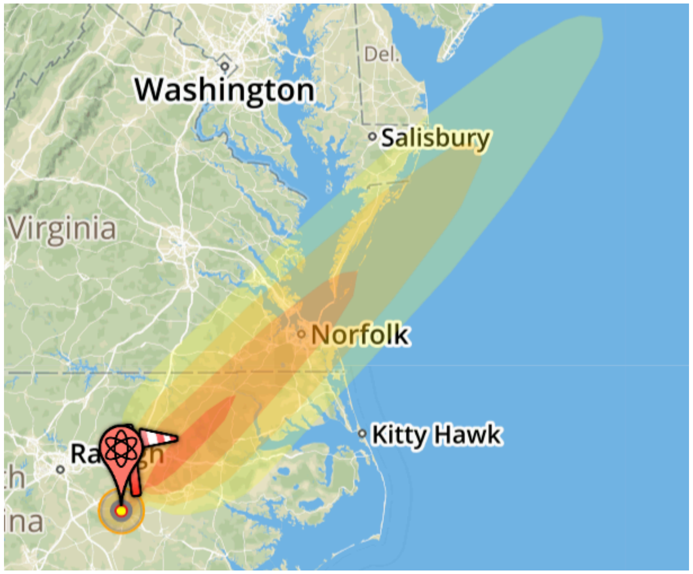


The Interesting Problem: Limiting Use

- Who might use a nuke without authorization?
  - Our "allies" where we station our nukes
    - Original motivation: Nukes stored in Turkey and Greece
  - Someone who can capture a nuke
    - This is what sold the military on the need for the problem:
      We had nukes in Germany which ***would*** be overrun in case of a war with the USSR
  - Our own military
    - General Jack D Ripper scenario
- The ***mandated*** solution:
  - Permissive Access Link (PAL)

Nuke Safety Features

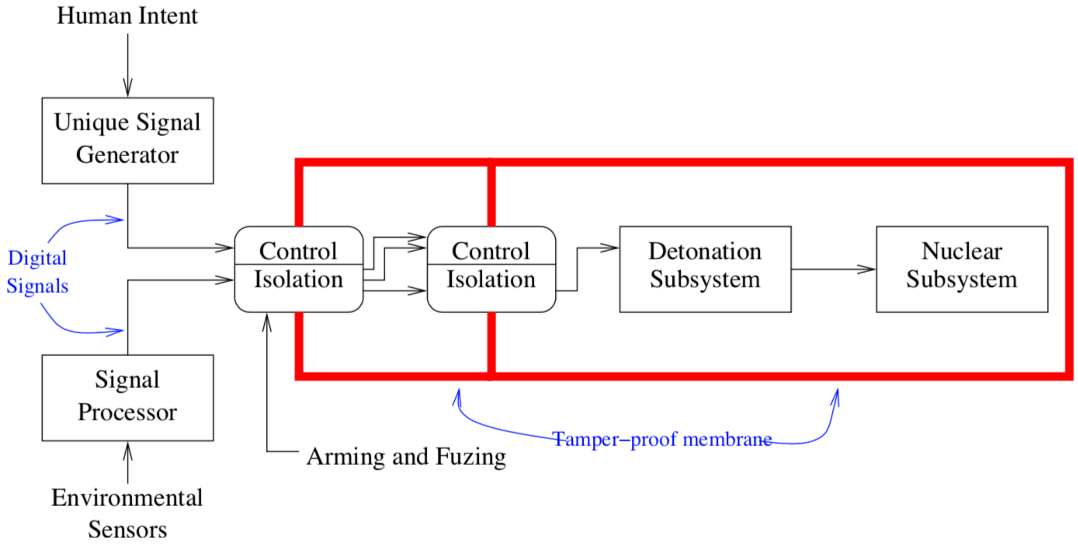
- One-point safety – no nuclear yield from detonation of one explosive charge.
- Strong link/weak link –
    - strong link provides electrical isolation;
    - weak link fails early under stress (heat, etc.)
- Environmental sensors – detect flight trajectory.
- Unique signal generator – digital signal used for coupling between stages.
- Insulation of the detonators from electrical energy.
- “Human intent” input.
- Tamper-resistant skin
- Use Control Systems



- Not always the case: In 1961 in South Carolina a B52 broke up
    - One of the two 4MT bombs ***almost*** detonated on impact, since it thought it was being dropped!

Bomb Safety Systems

- We have a "trusted base"
  - Isolated inside a tamper-detecting membrane
    - Breach the membrane -> disable the bomb
- We have human input
  - Used to generate a signal saying "its OK to go boom"



    - The user interface to the PAL can follow the same path/concepts
- We have critical paths that we can block
  - Complete mediation of the signal to go boom!

Unique Signal Generator

- Part of the strong link
  - Prevent any detonation without clear, unambiguous showing of “human intent”
- A ***safety*** system, not a security system
- Looks for a 24-bit signal that is extremely unlikely to happen during any conceivable accident. (Format of input bits not safety-critical)
  - Accidents can generate random or non-random data streams
  - Desired signal pattern is unclassified!
- Unique signal discriminator locks up on a ***single*** erroneous bit
- At least partially mechanical

PALs

- Originally electromechanical. (Some weapons used combination locks!)
- Newest model is microprocessor-based. There may still be a mechanical component.
  - Recent PAL codes are 6 or 12 digits.
- The weapon will permanently disable itself if too many wrong codes are entered.
- PALs respond to a variety of codes – several different arming codes for different groups of weapons, disarm, test, rekey, etc.
- It was possible, though difficult, to bypass early PALs.
  - Some even used false markings to deceive folks who didn't have the manual.
- It does not appear to be possible to bypass the newest "CAT F" PAL.

How are PALs built?

- We don't know, but some informed speculation from Steve...
- It is ***most likely*** based around the same basic mechanism as the unique signal generator
  - Gives a single point of control already in the system
  - Reports about it indicate that it was successfully evaluated in isolation
  - Take advantage of the existing trusted base of the tamper-resistant barrier around the warhead to protect the device

Deployment History

- Despite Kennedy's order, PALs were not deployed that quickly.
  - In 1974, there were still some unprotected nukes in Greece or Turkey
- PALs and use control systems were deployed on US-based strategic missiles by then
  - But the launch code was set to 00000000
  - Rational: the Air Force was more worried about failure to launch!
- A use control system was added to submarine-based missiles by 1997
- In 1981, half of the PALs were still mechanical combination locks

Steve Bellovin's Lessons Learned

- Understand what problem you’re solving
- Understand ***exactly*** what problem you’re solving
- If your abstraction is right:
  you can solve the key piece of the overall puzzle
- For access control, find the One True Mandatory Path — and block it.
  - And if there is more than one, you're doing it wrong!
- What is the real TCB of our systems?

Resuming Tor Discussion:
Real use for ***true hidden*** hidden services

- "Non-arbitrageable criminal activity"
  - Some crime which is universally attacked and targeted
    - So can't use "bulletproof hosting", CDNs like CloudFlare, or suitable “foreign” machine rooms:
      And since CloudFlare will service the anti-Semitic shitheads like gab.ai, terrorist breeding grounds like 8chan, and the actual nazis at Stormfront...
- Dark Markets
  - Marketplaces based on Bitcoin or other alternate currency
- Cybercrime Forums
  - Hoping to protect users/administrators from the fate of earlier markets
- Child Exploitation

The Dark Market Concept

- Four innovations:
- A censorship-resistant payment (Bitcoin)
  - Needed because illegal goods are not supported by Paypal etc
    - Bitcoin/cryptocurrency is the ***only game in town*** for US/Western Europe after the Feds smacked down Liberty Reserve and eGold
- An eBay-style ratings system with mandatory feedback
  - Vendors gain positive reputation through continued transactions
- An escrow service to handle disputes
  - Result is the user (should) only need to trust the market, not the vendors
- Accessable ***only*** as a Tor hidden service
  - Hiding the market from law enforcement

The Dark Markets: History

- All pretty much follow the template of the original “Silk Road”
    - Founded in 2011, Ross Ulbricht busted in October 2013
- The original Silk Road actually (mostly) lived up to its libertarian ideals
    - Including the libertarian ideal that if someone rips you off you should be able to call up the Hell’s Angels and put a hit on them
        - And the libertarian idea if someone is foolish enough to THINK you are a member of the Hell’s Angels you can rip them off for a large fortune for a fake hit
- Since then, markets come and go
    - But you can generally find the latest gossip on “deepdotweb"

The Dark Markets:
Not So Big, and ***Not Growing!***

- Kyle Soska and Nicolas Christin of CMU have crawled the dark markets for years
  - These markets ***deliberately*** leak sales rate information from mandatory reviews
- So simply crawl the markets, see the prices, see the volume, voila…
- Takeaways:
  - Market size has been relatively steady for years, about $300-500k a day sales
    - Latest peak got close to $1M a day
  - Dominated by Pot, MDMA, and stimulants, with secondary significance with opioids and psychedelics
  - A few sellers and a few markets dominate the revenue: A fair bit of “Winner take all”
    - But knock down any “winner” and another one takes its place

The Scams…

- You need a reputation for honesty to be a good crook
  - But you can burn that reputation for short-term profit
- The “Exit Scam” (e.g. pioneered by Tony76 on Silk Road)
  - Built up a positive reputation
  - Then have a big 4/20 sale
  - Require buyers to “Finalize Early”
    - Bypass escrow because of “problems”
  - Take the money and run!
- Can also do this on an entire ***market*** basis
  - The “Sheep Marketplace” being the most famous

And then the Child Exploitation types

- This is ***why*** I’m quite happy to see Tor Hidden Services ***burn!!!***
  - Because these do represent a serious problem:
    The success against “PlayPen” shows just how major these are
- A far bigger systemic problem than the dark markets:
  - Dark markets are low volume, and not getting worse
    - Plus the libertarian attitude of “drug users are mostly harming themselves, its the drug-associated crime that is the problem”
      - No indication of any ***successful*** murder resulting from dark market activity
  - But these are harming others
- They are also harming Tor:
  Tor itself is a very valuable tool for many legitimate uses, but the presence of the child exploitation sites on hidden services is a stain on Tor itself

Deanonymizing Hidden Services: Hacking...

- Most dark-net services are not very well run...
  - Either common off-the-shelf drek or custom drek
- And most have now learned ***don't ask questions on StackOverflow***
  - Here's looking at you, frosty…
- So they don't have a great deal of IT support services
  - A few hardening guides but nothing really robust

Onionscan…

- A tool written by Sarah Jamie Lewis
    - Available at https://github.com/s-rah/onionscan
- Idea is to look for very common weaknesses in Tor Hidden services
    - Default apache information screens
    - Web fingerprints
    - I believe a future version will check for common ssh keys elsewhere on the Internet
- Its really "dual use"
    - .onion site operators should use to make sure they aren't making rookie mistakes
    - Those investigation .onion sites should use to see if the target site made a rookie mistake!

Deanonymizing Visitors To Your Site FBI Style

- Start with a Tor Browser Bundle vulnerability…
  - Requires paying for a decent vulnerability:
    Firefox lacks sandboxing-type protections but you have to limit yourself to JavaScript
- Then take over the site you want to deanonymize visitors to…
- And simply hack the visitors to the site!
  - With a limited bit of malcode that just sends a “this is me” record back to an FBI-controlled computer

A History of NITs

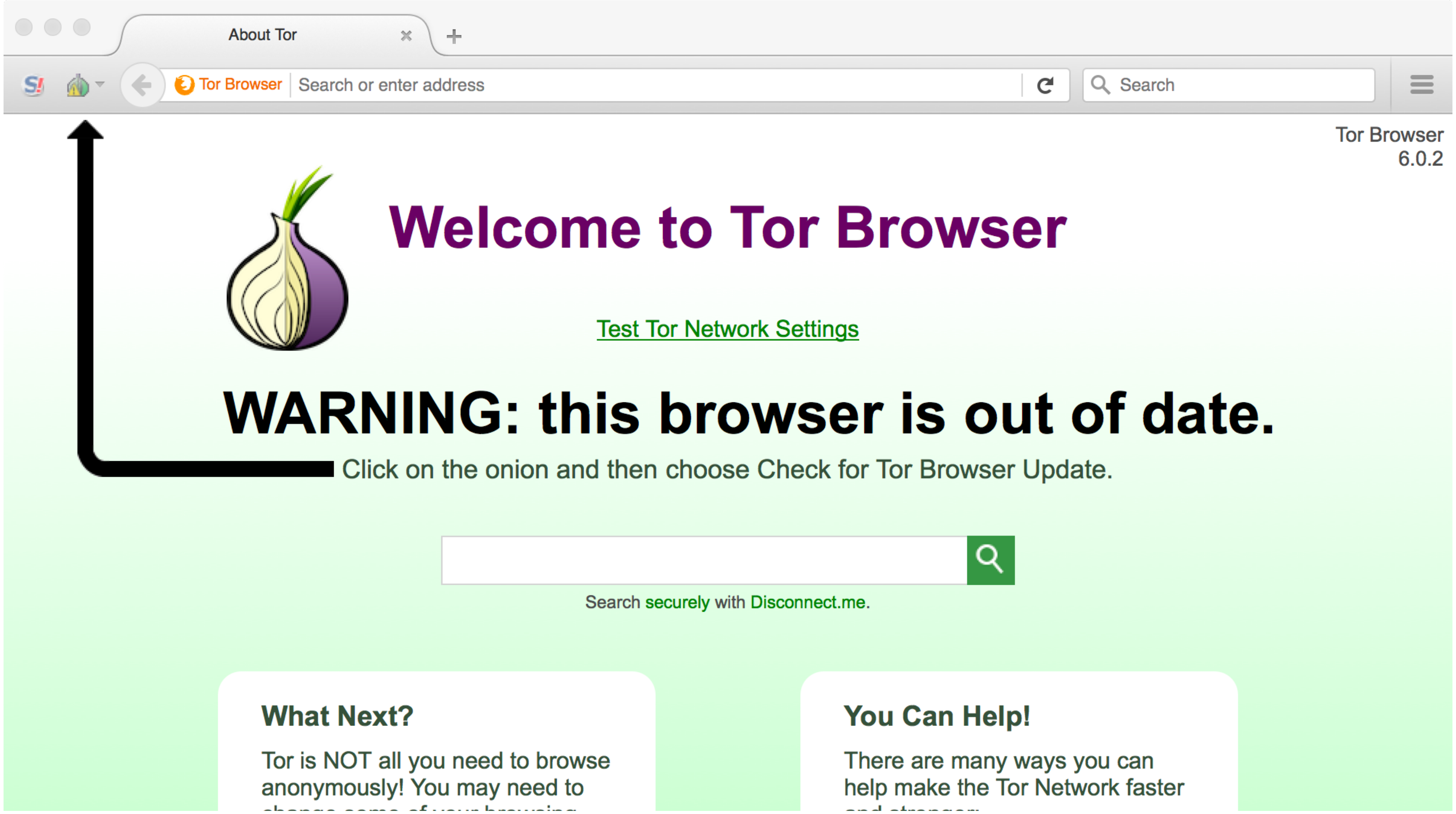
- The FBI calls their malicious code a NIT or Network Investigatory Technique
    - Because it sounds better to a magistrate judge than saying "we're gonna go hacking"
- The exploit attempts to take over the visitor's browser
- But the payload is small: just a "I'm this computer" sent over the Internet to an FBI controlled Internet address

A History of NITs: PedoBook

- The first known NIT targeting a hidden service was "PedoBook" back in 2012
  - Back then, many people used other web browsers to interact with Tor hidden services
- The NIT actually didn't even qualify as malcode
  - And a ***defense*** expert actually argued that it isn't hacking and probably didn't actually need a warrant
- Instead it was the "Metasploit Decloaking" flash applet:
  - A small bit of Flash which contacts the server directly, revealing the visitor's IP address

A History of NITs: Freedom Hosting

- The second big NIT targeted FreedomHosting
  - A hosting provider for Tor Hidden services with an, umm, generous policy towards abuse
    - Hosted services included TorMail (a mail service through Tor) and child porn sites
- FBI replaced the entire service with a NIT-serving page
- Fallout:
  - Very quickly noticed because there are multiple legit users of TorMail
  - Targeted an older Firefox vulnerability in Tor Browser
- Tor browser switched to much more aggressive autoupdates:
Now you ***must*** have a zero-day for a NIT payload to work



Tor Browser 6.0.2

Welcome to Tor Browser

Test Tor Network Settings

WARNING: this browser is out of date.

Click on the onion and then choose Check for Tor Browser Update.

Search securely with Disconnect.me.

What Next?

Tor is NOT all you need to browse anonymously! You may need to change some of your browsing

You Can Help!

There are many ways you can help make the Tor Network faster and stronger:

A History of NITs: Playpen

- The big one: PlayPen was a hidden service for child pornographers
  - In February 2015, the FBI captured the server and got a warrant to deploy a NIT to logged in visitors
    - The NIT warrant is public, but the malcode itself is still secret: >100,000 logins!
- What we do know:
  - This was big: hundreds of arrests, many abuse victims rescued
  - It almost certainly used a zero-day exploit for Tor Browser
- Courts are still hashing this out over two big questions
  - Is it valid under Rule 41?
    - ***Most*** have conclude "no, but a technical not constitutional flaw":
Good faith says that previous violations are OK, but not future violations
  - Does the defense have a right to examine the exploit?
    - I'll argue no, but some defense attorneys have successfully used a graymail technique

A History of NITs:
Two Years Ago

- Someone (probably the French police) captured a child porn site called the "GiftBox"
  - They modified it to serve up a NIT
- The NIT payload was almost identical to the one in the Freedom Hosting case
  - Suggesting assistance from either the FBI or the FBI's contractor
- The exploit was a ***new*** zero-day exploit targeting Firefox
  - Patch released within ***hours***
    - And yes, it was a C-related memory corruption (naturally)

NITs won't work well in the future against Tor!

- The current Tor browser hardened branch is just that, ***hardened***
  - And it will become mainstream in a future version:
    it uses a technique, ***selfrando***, with ***no currently known workaround!***
- Hardening will require that breaking Tor browser, even to just send a "I'm here" message, will require a chain of exploits
  - An information leakage to determine the address of a function and enough content in that function to enable an attack
    - Or the leakage of a lot of functions
  - PLUS a conventional vulnerability
  - And just wait until the Firefox rendering engine gets sandboxed too…
  - And ad in darknet users who are running without JavaScript
- Upshot: the current FBI exploit will need a massive upgrade if it will work at all!
  - And future exploits will be ***vastly*** more expensive and rarer
  - We should thank the FBI for their very valuable contributions to software hardening

Relevant classes

- CS 194: Undergrad cryptography
  - Nick may also have a 194 in a year if he gets drone funding...
- CS 276: Graduate crypto
- CS 261: Graduate security
- CS 261N: Graduate network security
- CS 294: Miscellaneous
  - In the Fall: decentralized security

email instructor for permission to enroll as undergraduate

Please fill in course evaluations

https://course-evaluations.berkeley.edu

- Very helpful to the department and to us, the staff
- Department-wide effort to increase responsiveness
- +1% points on the final exam
  - After filling it in, submit a screenshot of the confirmation
  - Instructions are posted on Piazza

Thanks to our staff… Nick, the TAs and the readers!

Keyhan Vakil

Nicholas Weaver

Weikeng Chen

Rafael Dutra

Rishabh Poddar

Ruta Jawale

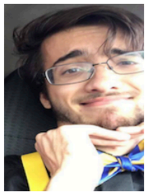
Arvind Iyengar

Austin Murdock

Grant Ho

Ruta Joshi

Seung Jin Yang


Spencer Mccall

Harley Patton

Nathan Malkin

Pratyush Mishra

Victor Chan

Thanks to our random facts "victims"

Most importantly,