Tor And Nukes!
Tor and NUKES!

NOT SURE IF SHOULD NUKE IT FROM ORBIT
OR BURN IT WITH FIRE
Tor: The Onion Router
Anonymous Websurfing

• Tor actually encompasses many different components

• The Tor network:
  • Provides a means for anonymous Internet connections with low(ish) latency by relaying connections through multiple Onion Router systems

• The Tor Browser bundle:
  • A copy of FireFox extended release with privacy optimizations, configured to only use the Tor network

• Tor Hidden Services:
  • Services only reachable though the Tor network

• Tor bridges with pluggable transports:
  • Systems to reach the Tor network using encapsulation to evade censorship

• Tor provides three separate capabilities in one package:
  • Client anonymity, censorship resistance, server anonymity
The Tor Threat Model: Anonymity of content against local adversaries

- The goal is to enable users to connect to other systems “anonymously” but with low latency
  - The remote system should have no way of knowing the IP address originating traffic
  - The local network should have no way of knowing the remote IP address the local user is contacting

- Important what is excluded: The global adversary
  - Tor does not even attempt to counter someone who can see all network traffic: It is probably impossible to do so and be low latency & efficient
The High Level Approach: Onion Routing

• The Tor network consists of thousands of independent Tor nodes, or “Onion Routers”
  • Each node has a distinct public key and communicates with other nodes over TLS connections
• A Tor circuit encrypts the data in a series of layers
  • Each hop away from the client removes a layer of encryption
  • Each hop towards the client adds a layer of encryption
• During circuit establishment, the client establishes a session key with the first hop…
  • And then with the second hop through the first hop
• The client has a **global** view of the Tor Network:
The directory servers provide a list of all Tor relays and their public keys
Tor Routing
In Action
Tor Routing
In Action
Creating the Circuit Layers…

- The client starts out by using an authenticated DHE key exchange with the first node...
  - So conceptually like DHE in TLS:
    - OR1 creates $g^a$, signs it with public key in the directory, sends to client
    - Client creates $g^b$, sends it to OR1
  - Creating a session key to talk to OR1
    - This first hop is commonly referred to as the “guard node”
- It then tells OR1 to extend this circuit to OR2
  - Through that, creating a session key for the client to talk to OR2 that OR1 \textit{does not know}
  - And OR2 doesn't know what the client is, just that it is somebody talking to OR1 requesting to extend the connection...
- It then tells OR2 to extend to OR3…
  - And OR1 won’t know where the client is extending the circuit to, only OR2 will
Unwrapping the Onion

- Now the client sends some data…
  - \( E(K_{or1}, E(K_{or2}, E(K_{or3}, Data))) \)
- OR1 decrypts it and passes on to OR2
  - \( E(K_{or2}, E(K_{or3}, Data)) \)
- OR2 then passes it on…
- Generally go through at least 3 hops…
  - Why 3? So that OR1 can’t call up OR2 and link everything trivially
- Messages are a fixed-sized payload
The Tor Browser…

- Surfing “anonymously” doesn’t simply depend on hiding your connection…
- But also configuring the browser to make sure it resists tracking
  - No persistent cookies or other data stores
  - *No deviations from other people* running the same browser
- Anonymity *only works in a crowd*…
  - So it really tries to make it all the same
- But by default it makes it easy to say “this person is using Tor”
But You Are Relying On Honest Exit Nodes…

- The exit node, where your traffic goes to the general Internet, is a man-in-the-middle…
- Who can see and modify all non-encrypted traffic
- The exit node also does the DNS lookups
- Exit nodes have not always been honest…

I just fell victim to a Tor exit node scam. (self.Bitcoin)
submitted 10 months ago * by ImStupidAgain

I was looking to mix some coins with bitmixer.io. I had visited their site and read up on reviews and decided to give them a try.
I figured it would be wise to connect through Tor, so I went to the clearnet site and copied the onion address that was shown on the homepage. Stupid me not realizing that I should use a regular connection and not Tor to find the address.
Fortunately I only lost a marginal amount, but everybody be warned once again. It’s really easy to make a mistake like that.
The mixer site: https://bitmixer.io
The real onion site: http://bitmixer2whesio1.onion/
Anonymity Invites Abuse…
(Stolen from Penny Arcade)
This Makes Using Tor Browser Painful…
And Also Makes Running Exit Nodes Painful…

- If you want to receive abuse complaints…
  - Run a Tor Exit Node

- Assuming your ISP even allows it…
  - Since they don’t like complaints either

- Serves as a large limit on Tor in practice:
  - Internal bandwidth is plentiful, but exit node bandwidth is restricted
  - Know a colleague who ran an exit node for research...
    - And got a *visit from the FBI!*
One Example of Abuse: 
The Harvard Bomb Threat…

- On December 16th, 2013, a Harvard student didn’t want to take his final in “Politics of American Education”…
  - So he emailed a bomb threat using Guerrilla Mail
  - But he was “smart” and used Tor and Tor Browser to access Guerrilla Mail

- Proved easy to track
  - “Hmm, this bomb threat was sent through Tor…”
  - “So who was using Tor on the Harvard campus…” (look in Netflow logs..)
  - “So who is this person…” (look in authentication logs)
  - “Hey FBI agent, wanna go knock on this guy’s door?!”

- There is no magic Operational Security (OPSEC) sauce…
  - And again, anonymity only works if there is a crowd
Censorship Resistance: Pluggable Transports

- Tor is really used by two separate communities
  - Anonymity types who want anonymity in their communication
  - Censorship-resistant types who want to communicate despite government action
    - The price for "free" censorship evasion is that your traffic acts to hide other anonymous users
- Vanilla Tor fails the latter *completely*
- So there is a framework to deploy bridges that encapsulate Tor over some other protocol
  - So if you are in a hostile network...
  - Lots of these, e.g. OBS3 (Obfuscating Protocol 3), OBS4, Meek...
OBS3 Blocking: China Style

• Its pretty easy to recognize something is probably the Tor obs3 obfuscation protocol
  • But there may be false positives...
    • And if you are scanning all internet traffic in China the base rate problem is going to get you

• So they scan all Internet traffic looking for obs3...
  • And then try to connect to any server that looks like obs3...
  • Do a handshake and if successful...

• If it is verified as an obs3 proxy...
  • China then blocks that IP/port for 24 hours
Meek: Collateral Freedom

• Meek is another pluggable transport
  • It uses Google App engine and other cloud services

• Does a TLS connection to the cloud service
  • And then encapsulates the Tor frames in requests laundered through the cloud service

• Goal is "Too important to block"
  • The TLS handshake is to a legitimate, should not be blocked service
  • And traffic analysis to tell the difference between Meek and the TLS service is going to be hard/have false positives
The End Of Collateral Freedom...

- Meek relied on "Domain fronting"
  - A "bug"/"feature" of TLS/HTTPS:
    You tell TLS what host you want to talk to
    You tell the HTTP server what host you want to talk to...

- So you tell TLS one thing
  - Which the censor can see

- And the web server something else
  - Because its a Google server, or a Cloudflare CDN server or...
    Which supports a large number of different hosts

- Recently all the major CDNs stopped supporting it
  - After all, it is a bug!
Tor Browser is also used to access Tor Hidden Services aka .onion sites

- Services that only exist in the Tor network
  - So the service, not just the client, has possible anonymity protection
  - The “Dark Web”
- A hash of the hidden service's public key
  - http://pwoah7foa6au2pul.onion
    - AlphaBay, one of many dark markets
  - https://facebookcorewwwi.onion
    - In this case, Facebook spent a lot of CPU time to create something distinctive
- Using this key hash, can query to set up a circuit to create a hidden service at a rendezvous point
  - And because it is the hash of the key we have end-to-end security
Tor Hidden Service:
Setting Up Introduction Point
Tor Hidden Service:
Query for Introduction, Arrange Rendezvous
Tor Hidden Service: Rendezvous and Data
We highly recommend that you disable Javascript when viewing the marketplace for better security.
Remarks...

- Want to keep your guard node constant for a long period of time...
  - Since the creation of new circuits is far easier to notice than any other activity
- Want to use a different node for the rendezvous point and introduction
  - Don’t want the rendezvous point to know who you are connecting to
- These are slow!
  - Going through 6+ hops in the Tor network!
Non-Hidden Tor Hidden Service: Connect Directly to Rendezvous
Non-Hidden Hidden Services Improve Performance

- No longer rely on exit nodes being honest
  - No longer rely on exit node bandwidth either
- Reduces the number of hops to be the same as a not hidden service
- Result: Huge performance win!
  - Not slow like a hidden service
  - Not limited by exit node bandwidth
- Any legitimate site offering a Tor hidden service should use this technique
  - Since legitimate sites don't need to hide!
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 and the actual nazis at Storefront are still online...

- 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
Welcome to Tor Browser

WARNING: this browser is out of date.
Click on the onion and then choose Check for Tor Browser Update.

Test Tor Network Settings

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
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
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...

- 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 "Tellur/Ulam" 2-stage device: 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!
  - 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 \textit{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?