# Worms, Tor, and Abuse

CS 161 Spring 2022 - Lecture 24

# Reminder:
# Malcode Wars and the Halting Problem...



- Cyberwars are not won by solving the halting problem...
  Cyberwars are won by making some other poor sod solve the halting problem!!!
  - In the limit, it is ***undecidable*** to know "is this code bad?"
- Modern focus is instead "is this code ***new?***"
  - Use a secure cryptographic hash (so sha-256 not md5)
  - Check hash with central repository:
    If ***not*** seen before, treat binary as inherently more suspicious
- Creates a bind for attackers:
  - Don't make your code *morphic:
    Known bad signature detectors find it
  - Make your code *morphic:
    It always appears as new and therefore ***inherently*** suspicious

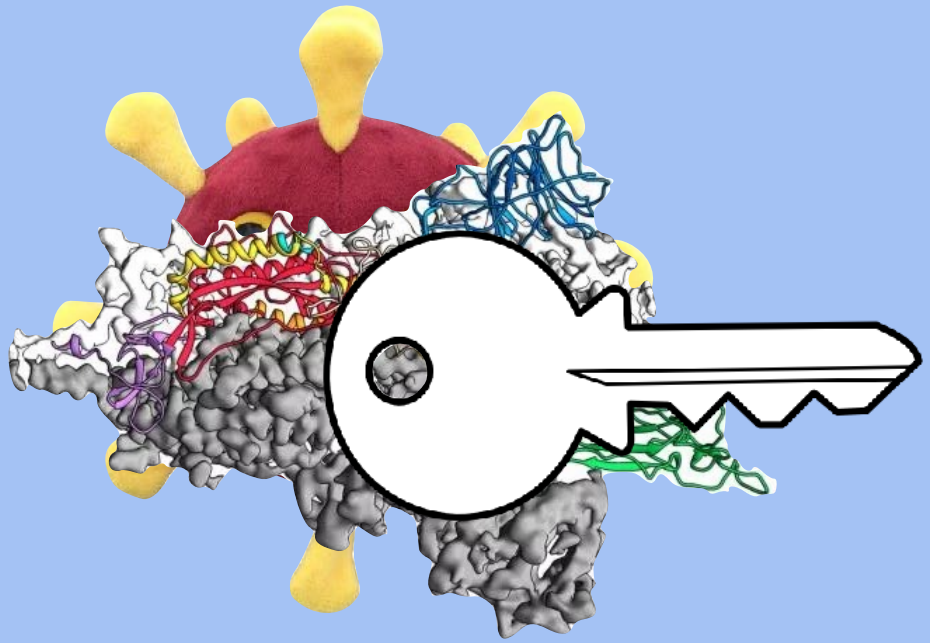
# Creating binds is very powerful...



- You have a detector D for some bad behavior...
  - So bad-guys come up with a way of avoiding the detector D
- So come up with a detection strategy for ***avoiding detector D***
  - So to avoid ***this*** detector, the attacker ***must not*** try to avoid D
- When you can do it, it is very powerful!

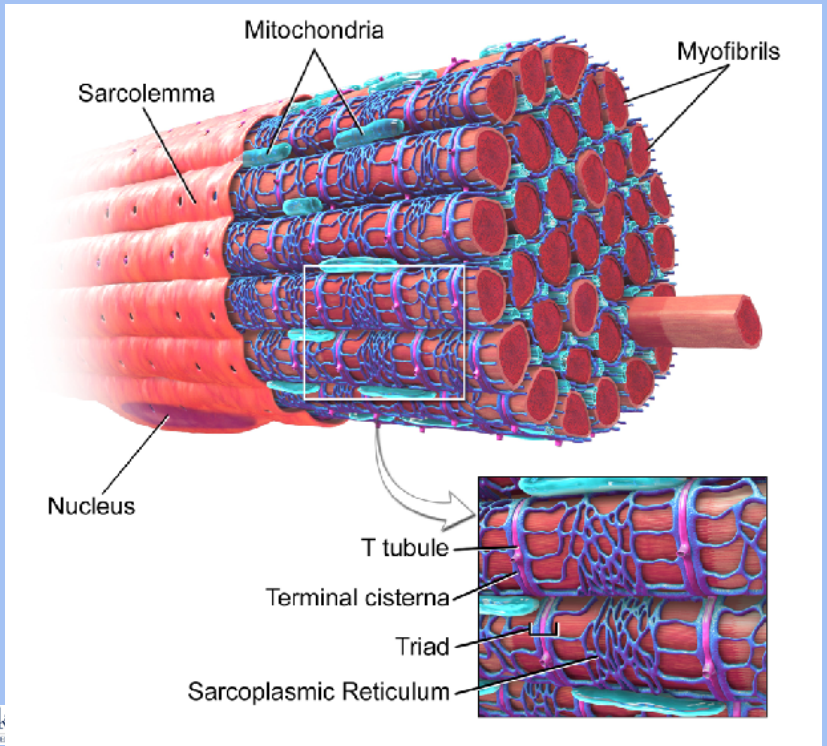
# A Similar Bind for SARS-CoV-2: Our Enemy the Spike

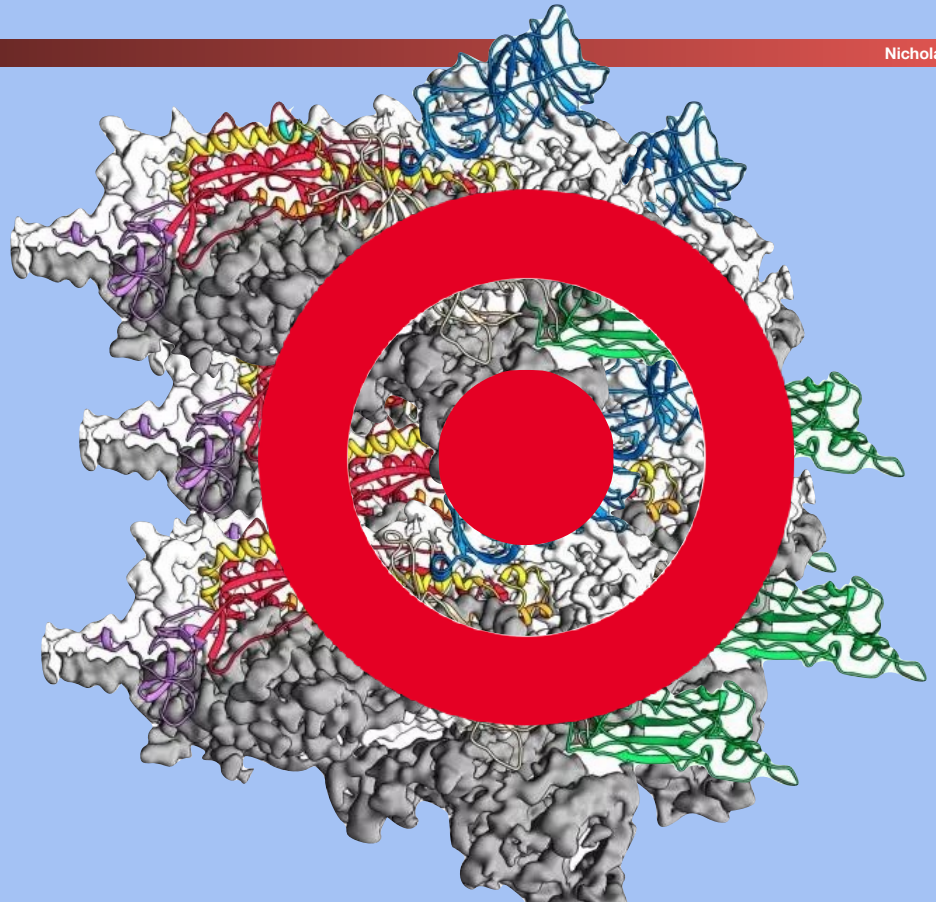




# The Vaccine...
# Invade and reprogram some muscle

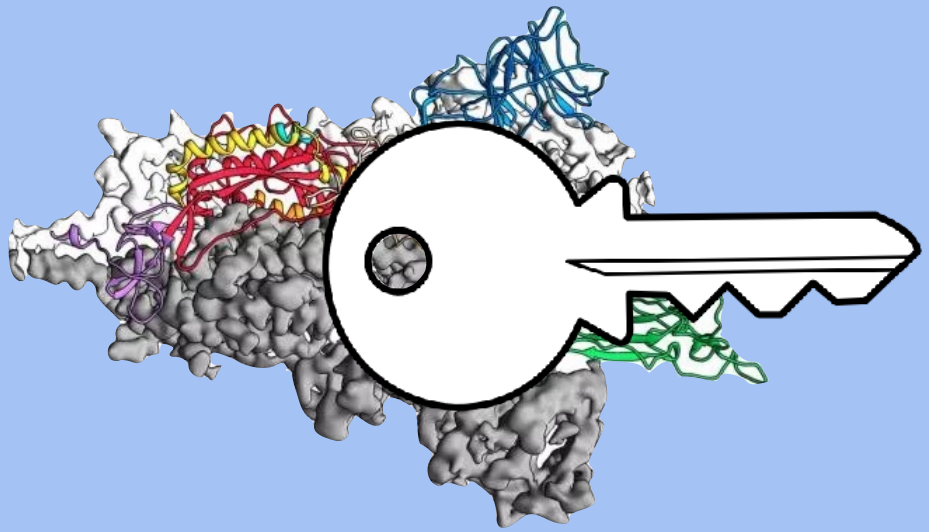




# So The Spike's Bind





# Infection Cleanup

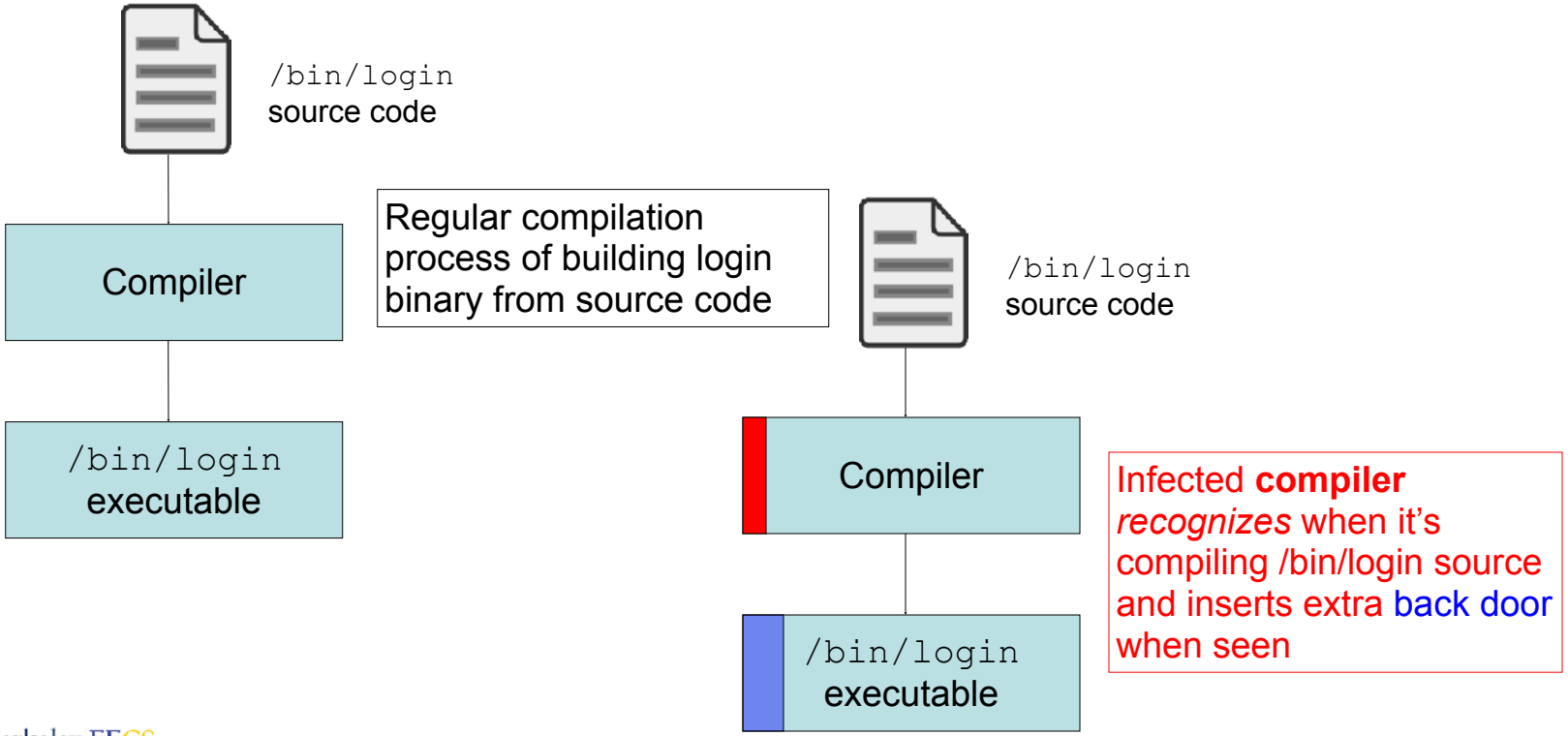


- Once malware detected on a system, how do we get rid of it?
- May require restoring/repairing many files
  - This is part of what AV companies sell: per-specimen disinfection procedures
- What about if malware executed with administrator privileges?
  - "Game over man, Game Over!"
  - “Dust off and nuke the entire site from orbit. It’s the only way to be sure” ALIENS
  - i.e., rebuild system from original media + data backups
- Malware may include a rootkit: kernel patches to hide its presence (its existence on disk, processes)

# Infection Cleanup, con't

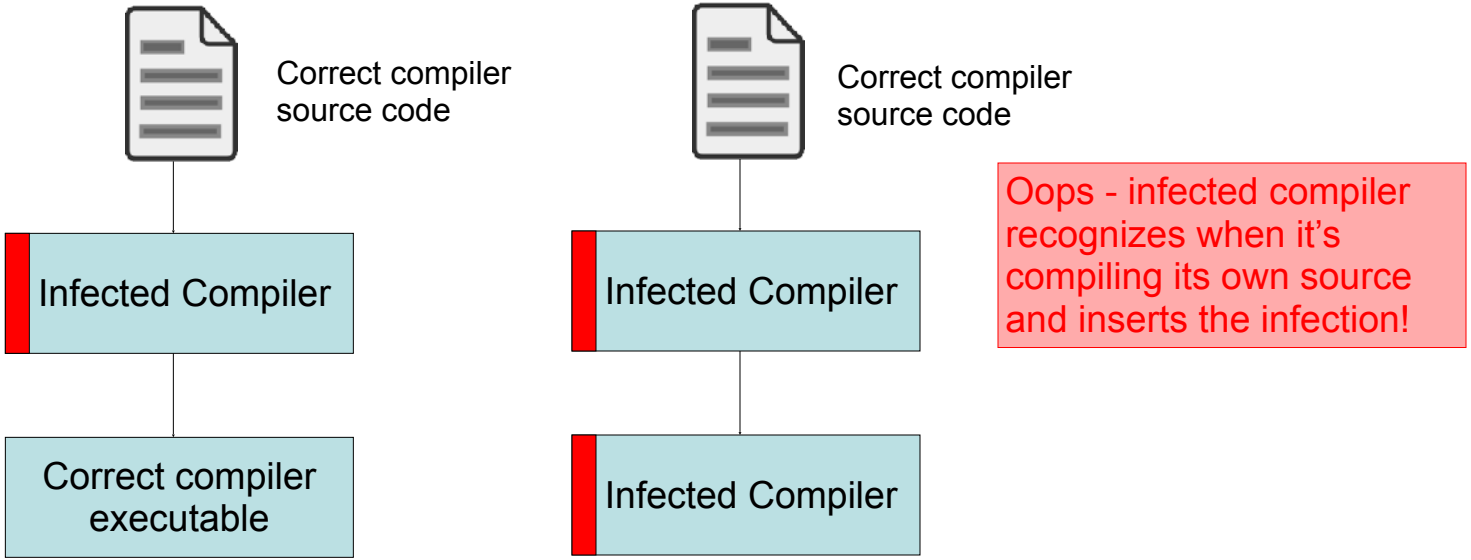


- If we have complete source code for system, we could rebuild from that instead, couldn't we?
- No!
- Suppose forensic analysis shows that virus introduced a backdoor in /bin/login executable
  - (Note: this threat isn't specific to viruses; applies to any malware)
- Cleanup procedure: rebuild /bin/login from source …


/bin/login
source code
Compiler
/bin/login
executable
Regular compilation process of building login binary from source code
/bin/login
source code
Compiler
/bin/login
executable
Infected **compiler** *recognizes* when it's compiling /bin/login source and inserts extra back door when seen

No problem, first step, rebuild the compiler so it's uninfected






Oops - infected compiler recognizes when it's compiling its own source and inserts the infection!

**No** amount of careful source-code scrutiny can prevent this problem.
And if the *hardware* has a back door …

*Reflections on Trusting Trust*
Turing-Award Lecture, Ken Thompson, 1983

# More On "Rootkits"



- If you control the operating system...
  - You can hide extremely well
- EG, your malcode is on disk...
  - So it will persist across reboots
- But if you try to ***read the disk***...
  - The operating system just says "Uhh, this doesn't exist!"

# Even More Places To Hide!



- In the BIOS/EFI Firmware!
  - So you corrupt the BIOS which corrupts the OS...
  - Really hard to find:
    Defense, ***only*** run cryptographically signed BIOS code as part of the Trusted Base
- In the disk controller firmware!
  - So the master boot record, when read on boot up corrupts the OS...
  - But when you try to read the MBR later... It is just "normal"
  - Again, defense is ***signed code:*** The Firmware will only load a signed operating system
    - Make sure the disk itself is ***not trusted!***

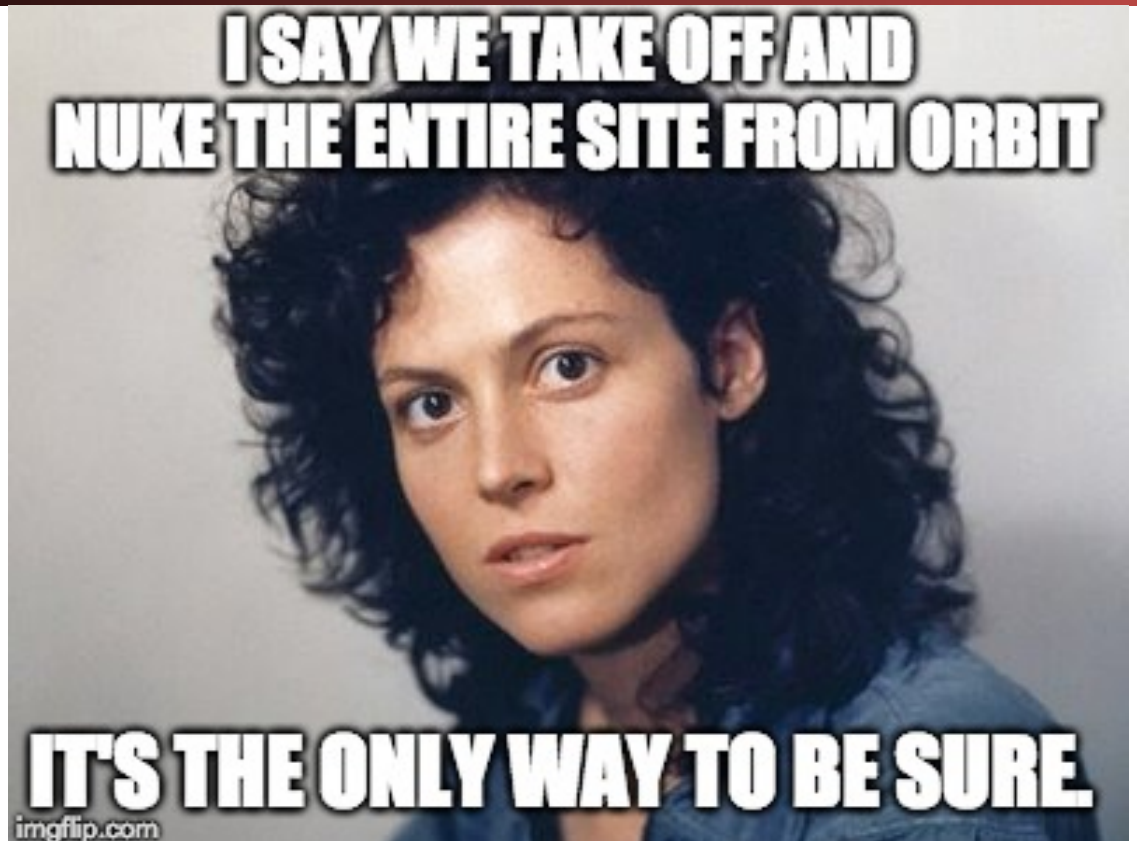
# Robust Rootkit Detection: Detect the act of hiding...



- Do an "in-system" scan of the disk...
  - Record it to a USB drive
- Reboot the system with trusted media
  - So a known good operating system
- Do the same scan!
  - If the scans are different, you found the rootkit!
- For windows, you can also do a "high/low scan" on the Registry:
  - Forces the bad guy to understand the registry as well as Mark Russinovich (the guy behind Sysinternals who's company Microsoft bought because he understood the Registry better than Microsoft's own employees!)
- Forces a bind on the attacker:
  - Hide and persist? You can be detected
  - Hide but don't persist? You can't survive reboots!

# Which Means ***Proper*** Malcode Cleanup...





# Worms

# Worms



- **Worm**: Code that does not require user action to propagate
  - Usually infects a computer by altering some already-running code
  - Unlike malware, no user interaction is required for the worm to spread to other users

# Propagation Strategies

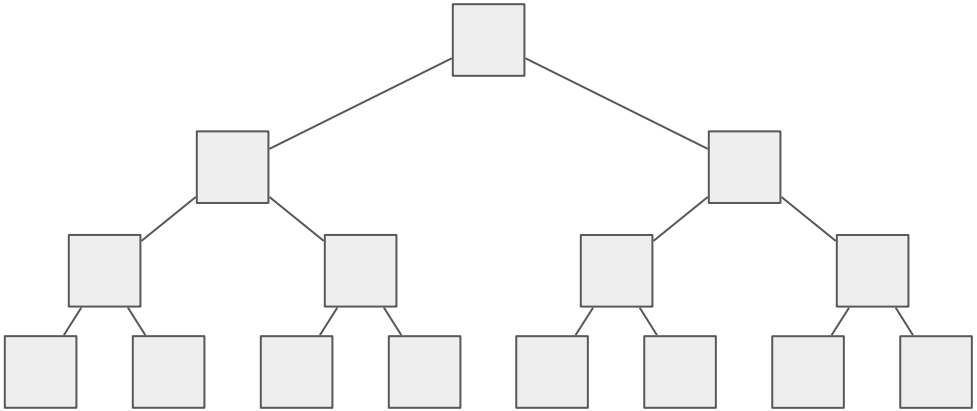


- How does the worm find new users to infect?
  - Randomly choose machines: generate a random 32-bit IP address and try connecting to it
  - Search worms: Use Google searches to find victims
  - Scanning: Look for targets (can be limited by bandwidth)
  - Target lists
    - Pre-generated lists (hit lists)
    - Lists of users stored on infected hosts
    - Query a third-party server that lists other servers
  - Passive: Wait for another user to contact you, and reply with the infection
- How does the worm force code to run?
  - Buffer overflows for code injection
  - A web worm might propagate with an XSS vulnerability

# Modeling Worm Propagation



- Worms can potentially spread extremely quickly because they parallelize the process of propagating/replicating
- More computers infected = more computers to spread the worm further
- Viruses have the same property, but usually spread more slowly, since user action is needed to activate the virus



If each infected computer can infect two more computers, we get exponential growth!

# Modeling Worm Propagation

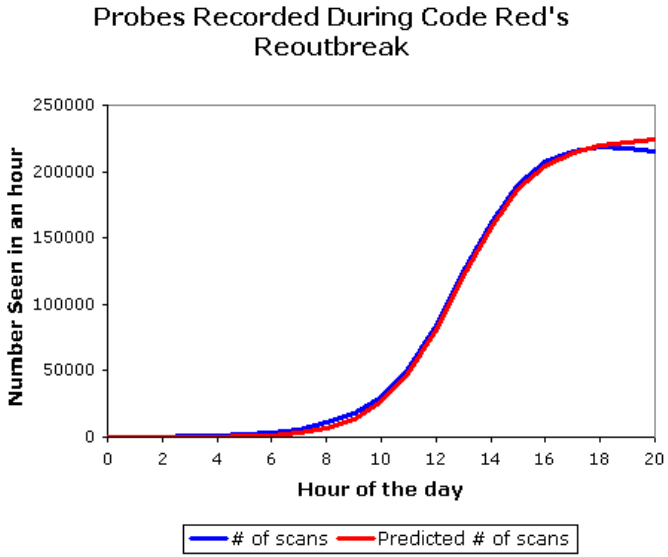


- Worm propagation can be modeled as an infectious epidemic
  - We can use the same models that biologists use to model the spread of infectious diseases
- The spread of the worm depends on:
  - The size of the population
  - The proportion of the population vulnerable to infection
  - The number of infected hosts
  - The contact rate (how often an infected host communicates with other hosts)

# Modeling Worm Propagation



- The number of infected hosts grows **logistically**
  - Initial growth is exponential:
    More infected hosts = more opportunities to infect
  - Later growth slows down: Harder to find new non-infected hosts to infect
- Logistic growth is a good model for worm propagation



# Note: Some worms are not blue slides...



- Why? Not because I will test you on them...
  - I ***won't*** require you to know the name of the Morris worm etc...
- But because these were seminal events in computer security
  - So you should know them by name

# History of Worms: Morris Worm

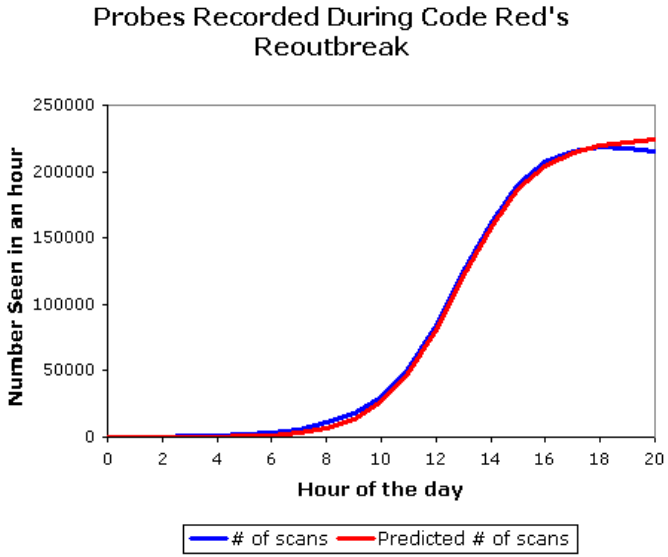


- **Morris Worm**: November 2, 1988
  - Considered the first Internet worm and influenced generations of future worms (and malware)
- Strategies to infect systems
  - Exploit multiple buffer overflows
  - Guess common passwords
  - Activate a “debug” configuration option that provided shell access
  - Exploit common user accounts across different machines
- Strategies to find users to infect
  - Scan local subnet
  - Machines listed in the system’s network configuration
  - Look through user files for mention of remote hosts
- Had a bug!
  - “Is a copy running on this computer already” check didn’t work…
    Resulted in exponential growth of instances on each victim
- The author (Robert Morris Jr) pled guilty to a felony
- **Takeaway**: Worms are hard to get ***right***.
- **Takeaway**: ***Do not experiment with self propagating code!***

# History of Worms: Code Red



- Code Red: July 13, 2001
  - Generally considered the start of the “modern era” of worms
- Payload: Defacing vulnerable websites
  - Add a “hacked” message on English-language websites
- Payload: DoS attack against the US White House
  - For the first 20 days of every month, focus on spreading to other computers
  - For the rest of the month, flood the White House’s website’s IP address with packets
  - Forced the White House to change its website’s IP address
- Strategies to infect systems
  - Exploit buffer overflow in Microsoft IIS web servers
  - Vulnerable by default in many systems
  - The vulnerability and fix were announced one month earlier



# History of Worms: Code Red

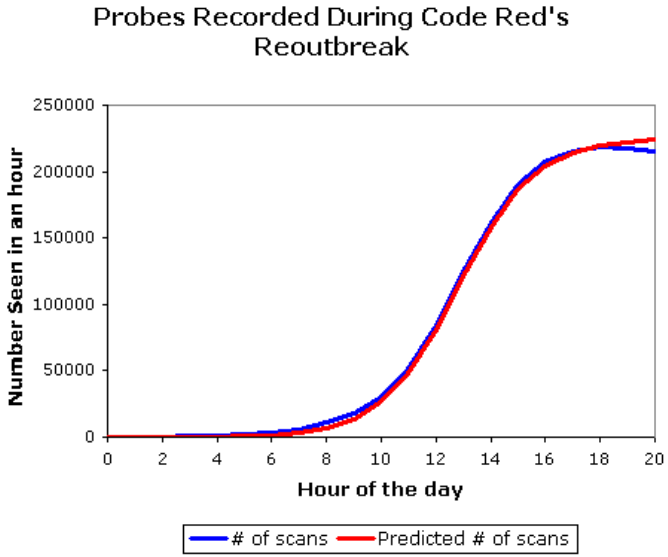


- Strategies to find users to infect
  - Random scanning of 32-bit IP address space
    - Use a PRNG to generate a (pseudo)random 32-bit IP address
    - Try connecting to it
    - If connection successful, try infecting it
    - If not, generate another IP address and repeat
  - First release (July 13, 2001): Every instance used the same PRNG seed
    - Worm spread was linear: every infected machine tried to infect the same computers
  - Revision (July 19, 2001): PRNG is seeded differently for every machine
    - Worm spread is now logistic!

# History of Worms: Code Red



- Code Red took 13 hours to reach peak infection rate
  - Corollary on SARS-CoV-19:
    The only time to react to an exponential growth is when, in retrospect, people will complain you acted too soon!
- **Takeaway**: Exponential growth may be fast but they can take a while to get going



# History of Worms: Warhol Worm

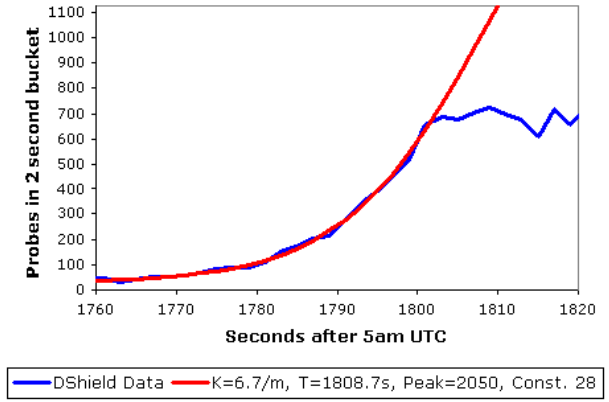


- Nick's reaction to Code Red: "13 hours? That is slow"
- Ideas for speeding up the infection rate of Code Red
  - Preseed to skip the initial ramp-up
  - Scan faster: 100 times per second instead of 10 times per second
  - Scan smarter: Self-coordinated scanning techniques with shutoff strategies
  - Ideas were validated in simulation that matched Code Red's behavior with Code Red's parameters
- Could spread globally in 15 minutes
- Became part of the paper "How to 0wn the Internet in Your Spare Time"
- **Takeaway**: Any robust worm defense needs to be automatic
- **Takeaway**: See, ***you don't need to experiment with self propagating code***!

# History of Worms: Slammer



- Strategies to infect systems
  - Use UDP instead of TCP to infect other computers (faster, avoid a three-way handshake)
  - Entire worm fits in a single UDP packet
  - Stateless spreading: Sending one packet is enough to infect a new computer ("fire and forget")
- Result: Extremely quick spread
  - 75,000+ hosts infected in under 10 minutes
  - Number of infected hosts doubled every 8.5 seconds




Slammer was so fast that it overwhelmed the Internet: No more packets could be sent, slowing the exponential growth

# Witty...



- A worm like Slammer but with a twist...
  - Targeted network intrusion detection sensors!
  - Released ~36 hours after vulnerability disclosure and patch availability!
- Payload wasn't just spreading, however...
  - while true {
    for i := range(20000){
      send self to random target;
    }
    select random disk (0-7)
    if disk exists {
      select random block, erase it;
    }}

# Stuxnet



- Discovered July 2010. (Released: Mar 2010?)
- Multi-mode spreading:
  - Initially spreads via USB (virus-like)
  - Once inside a network, quickly spreads internally using Windows RPC scanning
- Kill switch: programmed to die June 24, 2012
- Targeted SCADA systems
  - Used for industrial control systems, like manufacturing, power plants
- Symantec: infections geographically clustered
  - Iran: 59%; Indonesia: 18%; India: 8%

# Stuxnet, con't



- Used four Zero Days
  - Unprecedented expense on the part of the author
- "Rootkit" for hiding infection based on installing Windows drivers with valid digital signatures
  - Attacker stole private keys for certificates from two companies in Taiwan
- Payload: do nothing …
  - … unless attached to particular models of frequency converter drives operating at 807-1210Hz
  - … like those made in Iran (and Finland) …
  - … and used to operate centrifuges for producing enriched uranium for nuclear weapons

# Stuxnet, con't



- Payload: do nothing …
  - … unless attached to particular models of frequency converter drives operating at 807-1210Hz
  - … like those made in Iran (and Finland) …
  - … and used to operate centrifuges for producing enriched uranium for nuclear weapons
- For these, worm would slowly increase drive frequency to 1410Hz
  - … enough to cause centrifuge to fly apart …
  - … while sending out fake readings from control system indicating everything was okay …
- … and then drop it back to normal range

# Israel Tests on Worm Called Crucial in Iran Nuclear Delay

By WILLIAM J. BROAD, JOHN MARKOFF and DAVID E. SANGER
Published: January 15, 2011

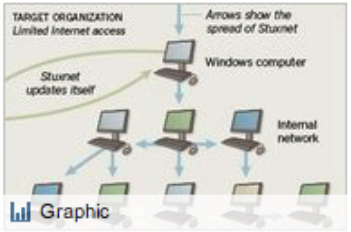
*This article is by **William J. Broad**, **John Markoff** and **David E. Sanger**.*

Enlarge This Image

Nicholas Roberts for The New York Times

Ralph Langner, an independent computer security expert, solved Stuxnet.

## Multimedia




How Stuxnet Spreads

The Dimona complex in the Negev desert is famous as the heavily guarded heart of Israel's never-acknowledged nuclear arms program, where neat rows of factories make atomic fuel for the arsenal.

Over the past two years, according to intelligence and military experts familiar with its operations, Dimona has taken on a new, equally secret role — as a critical testing ground in a joint American and Israeli effort to undermine Iran's efforts to make a bomb of its own.

Behind Dimona's barbed wire, the experts say, Israel has spun nuclear centrifuges virtually identical to Iran's at Natanz, where Iranian scientists are struggling to enrich uranium. They say Dimona tested the effectiveness of the Stuxnet computer worm, a destructive program that appears to have wiped out roughly a fifth of Iran's nuclear

# The "Toddler" Attack Payload...



- Stuxnet was very carefully engineered...
  - Designed to only go off under ***very specific*** circumstances
- But industrial control systems are inherently vulnerable
  - They consist of sensors and actuators
  - And safety is a ***global*** property
- Generic Boom:
  - At zero hour, the payload sees that it is on control system:
    map the sensors and actuators, see which ones are low speed vs high speed
  - T+30 minutes: Start replaying sensor data, switch actuators in low-speed system
  - T+60 minutes: Switch all actuators at high speed...
- This ***has been done***:
  A presumably Russian test attack on the Ukranian power grid! ("CrashOverride" attack)

# Then who "WannaCry"?



- The modern way of making profit from computer crime: Ransomware!
  - "Give us X Bitcoin or you'll never see your data again!"
  - The North Koreans apparently are doing this as a matter of government policy?!?!?
- So lets combine a ransomware payload with a self-spreading worm...
  - Then sit back and PROFIT!!!!
- Oh, wait...
  - The worm escaped early and the ransomware payload wasn't fully tested!
  - A ton of work for absolutely no profit:

    🇰🇵 -> 😭

    Everyone else -> 🤦🤣 if it didn't happen to disrupt a lot of businesses and destroy a lot of data.

# And NotPetya...



- NotPetya was a worm deliberately launched by Russia against Ukraine
  - Initial spread: A corrupted update to MeDoc Ukranian Tax Software
  - Then spread within an institution using "Eternal Blue" (Windows vulnerability) and "Mimikatz"
    - Mimikatz is way ***way more*** powerful:
      Takes advantage of windows transitive authorization...
    - IF you are running on the admin's machine, you can take over the domain controller
    - IF you are running on the domain controller, you can take over ***every computer***!!!
- Then wiped machines as fake ransomware
  - Give a veneer of deniability...
  - Shut down Mersk and many other global companies!

# And Overall Taxonomy of Spread



- Scanning
  - Look for targets
  - Can be bandwidth limited
- "Target Lists"
  - Pregenerated (Hitlist)
  - On-the-host (Topological)
  - Query a third party server that lists servers (Metaserver)
- Passive
  - Wait for a contact: Infect with the counter-response
- More detailed taxonomy here:
  - http://www.icir.org/vern/papers/taxonomy.pdf

# Tor

# 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

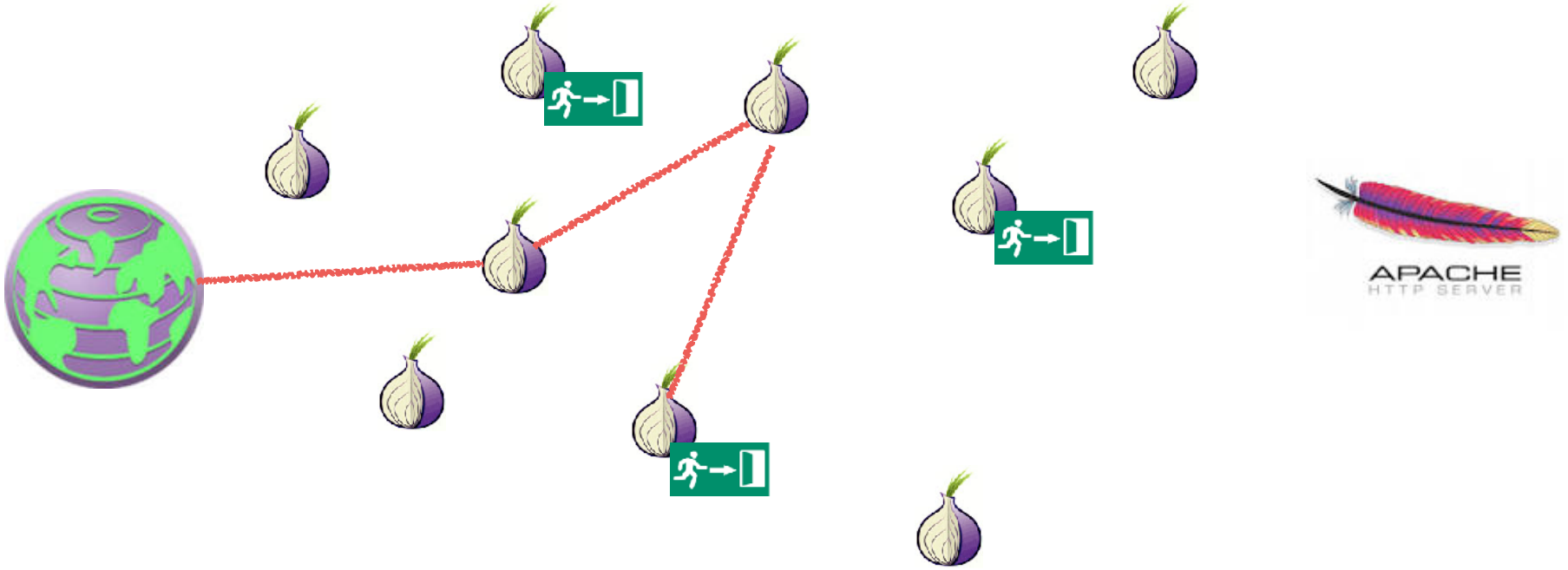
# The High Level Approach: Onion Routing



- 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***

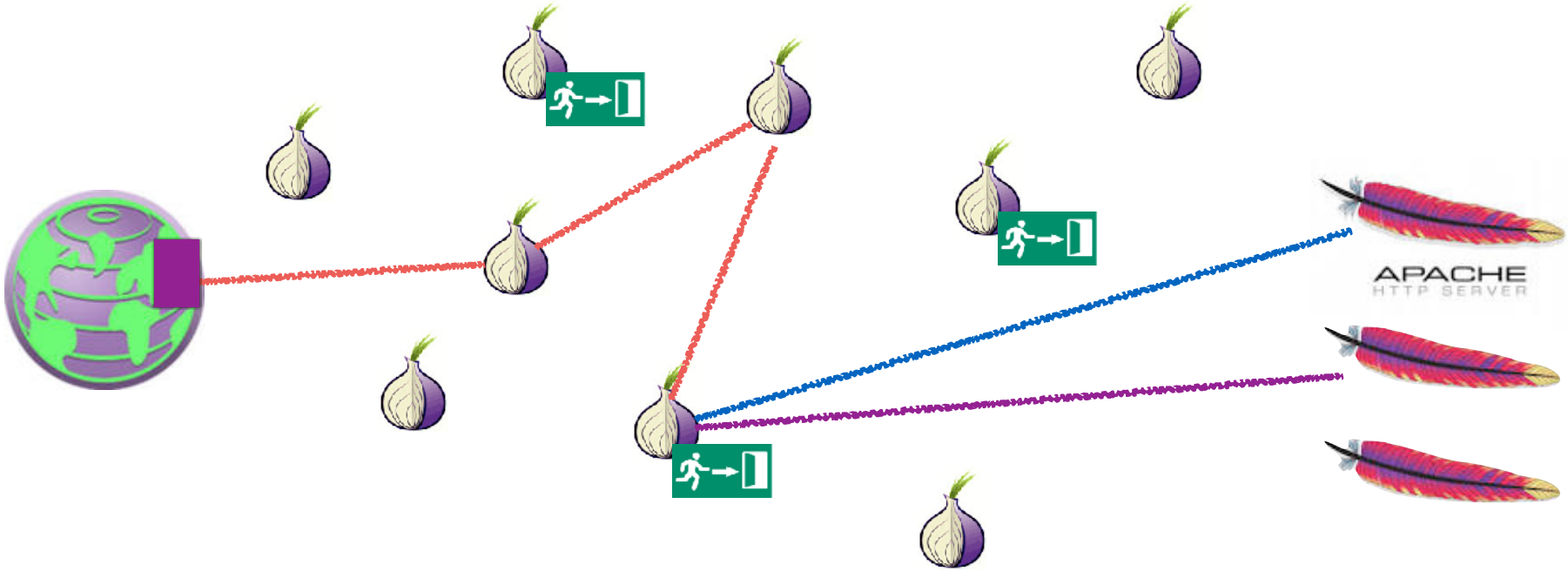
# Tor Routing
# In Action





# Tor Routing
# In Action





# Creating the Circuit Layers…



- The client starts out by using an authenticated DHE key exchange with the first node…
  - OR1 creates $g^a$, signs it with its private key, sends $g^a$, **Sign**($K_{priv\_or1}$, $g^a$) to client
    Client creates $g^b$, sends it to OR1
    Client does **Verify**($K_{pub\_or1}$, $g^a$)
  - Creating a session key $K_{OR1}$ as $H(g^{ab})$
    - 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 ***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…

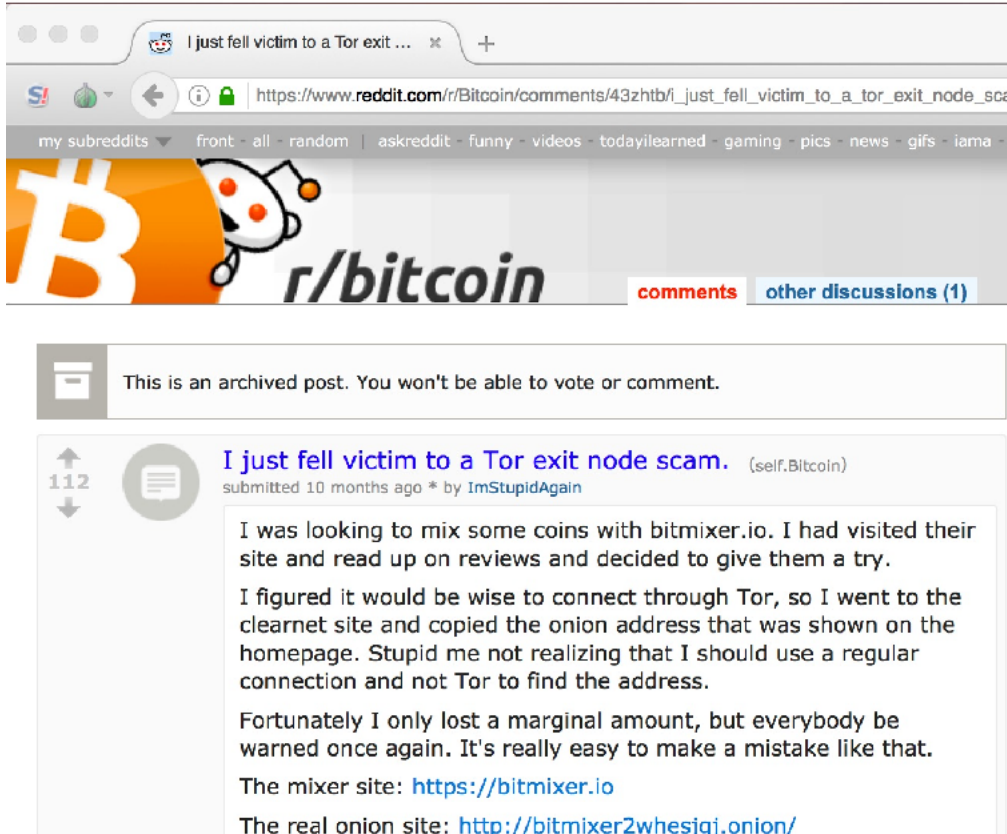


- Surfing “anonymously” doesn’t simply depend on hiding your connection…
- But also configuring the browser to make sure it resists tracking
  - No persistent cookies or other data stores
  - ***No deviations from other people*** running the same browser
- Anonymity ***only works in a crowd***…
  - So 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…

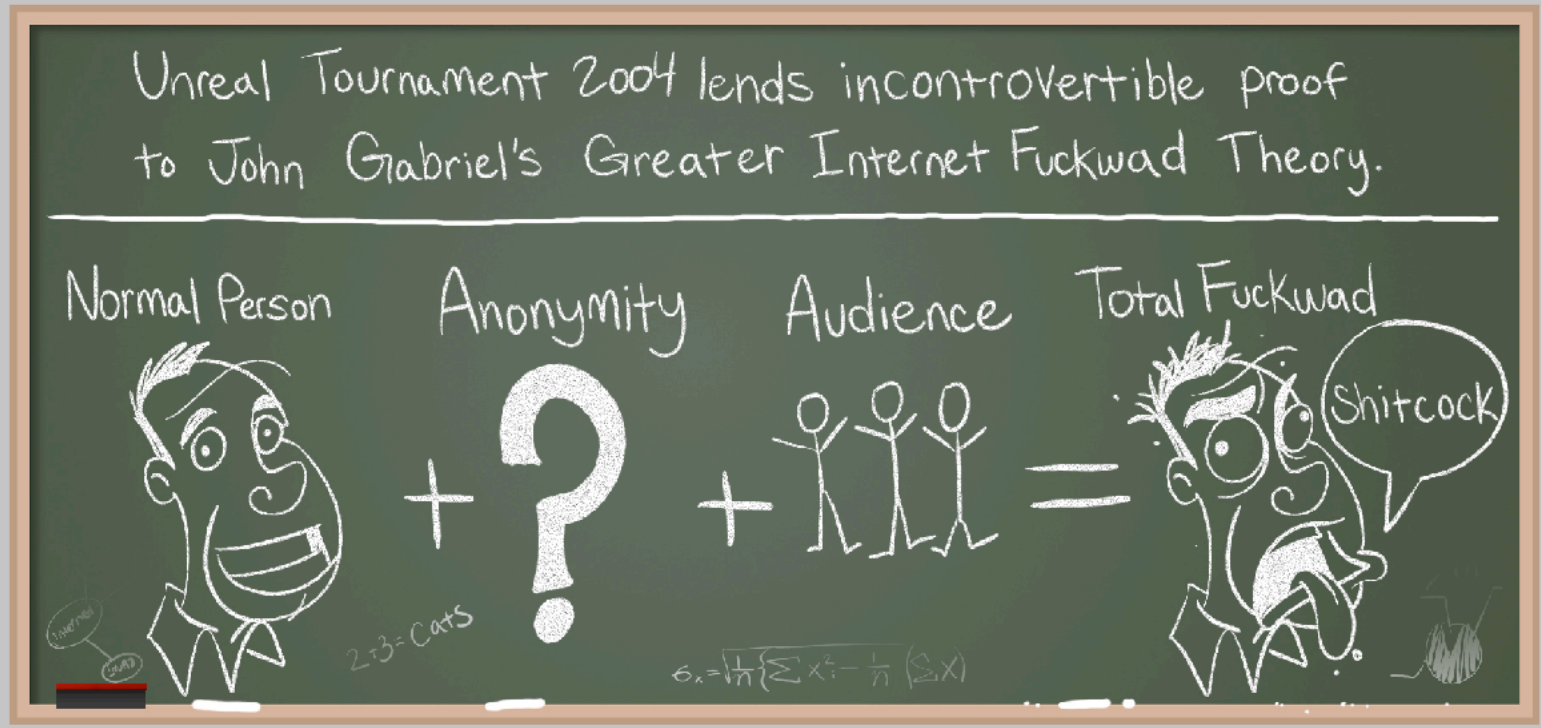


- The exit node, where your traffic goes to the general Internet, is a man-in-the-middle…
  - Who can see and modify all non-encrypted traffic
  - The exit node also does the DNS lookups
- Exit nodes have not always been honest…



# 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***!

# Censorship Evasion...



- Tor is actually really ***bad*** for evading censorship
  - It is trivial to tell that someone on the network is running Tor
- There are ***optional*** pluggable transports that attempt to hide the traffic
  - The problem is you have to learn about these...
    Yet if the censor does, it won't work!
- And then the user has all the bad of Tor...
  - **Fate sharing** with the exit nodes
  - Significantly worse latency
  - Oh, and Tor Browser's not saving history is not necessarily nice!
- Only good thing is it is "free"
  - Tor project gets paid largely for counter-censorship
  - Users are "paying" by providing traffic for those who want anonymity to hide in

# Abuse:
# Content Warning



- Serious content warnings ahead
  - Sexual Harassment
  - Domestic Violence
  - Child Abuse
- Stalkerware and Tor Hidden Services are not
- Implicit blue background on all slides
  - But I think it is important to be aware of these hard problems

# Here Be Blue Slides...



- The rest of this lecture is "blue slides"
  - I won't have explicit "**takeaway**" portions either that you are responsible for
- But there are many important real-world takeaways
  - So I'm including them:
    There is a serious moral dimension to computing and we neglect them at our peril

# Stalkerware:
# The Intimiate Partner Threat in Action



- The "Intimate Partner Threat" is one of the most powerful adversaries
  - They have physical access to your devices
  - They have intimate knowledge
  - They are integrated into your social circle
  - They are highly motivated: no "bear race" is possible
- IPT is often associated within the larger context of stalking & domestic violence
  - This is an area of computer security where there are lives at stake
  - It is also an insanely hard problem because of the attacker's resources

# Example IPT Attack: Compromise Facebook



- I have a colleague who was going through a divorce
  - It was not the friendliest of divorces
  - One day their (ex) partner broke into their facebook account!
- How?
  - Facebook password recovery option: Have 3 friends help out
  - So select three Facebook friends of the target:
    - The family dog
    - A member of their wedding party on the attacker's side
    - The attacker's best friend
  - Attack relied on knowledge of the social circle

# The Stalkerware Problem



- Generally refers to surreptitious monitoring software installed on the target's computer
  - https://stopstalkerware.org
- Method of installation usually takes advantage of physical access
  - Unlocked computer
  - Unlocked phone
- Once installed it enables surreptitious monitoring without notifying the victim
  - Common features include monitoring all messages and location tracking

# The Problem:
# Reuse of functionality...

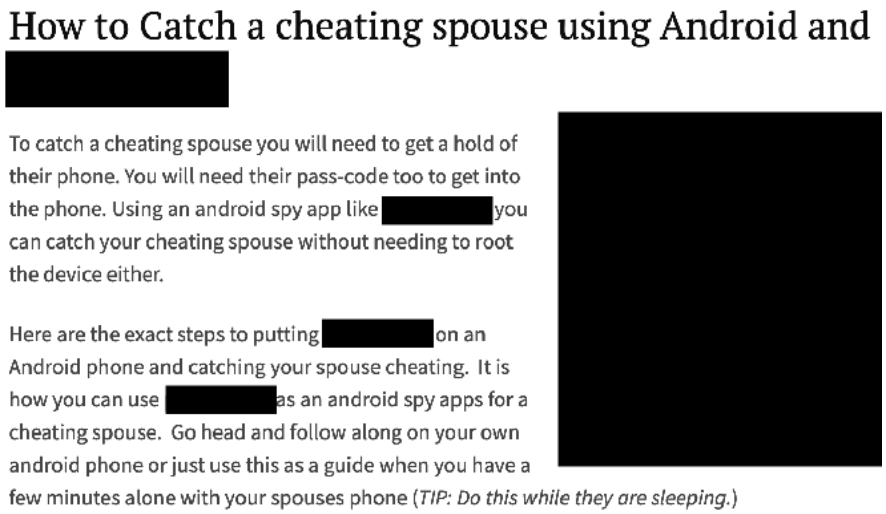


- Cellular phones support "Mobile Device Management"
  - A ***business*** phone belongs to the business, not the user
  - So the phones have hooks to monitor components
- Family Sharing/Parental Controls
  - EG, Apple allows sharing location with others in the family...
    The IPT can surreptitiously enable this
- Android supports "side-loading" of applications with physical access and the password
  - Allows bypassing security checks and vetting that occurs in application stores

# Just One Example: Installation of Stalkerware...



- Abuser does a Google search
  - Google doesn't allow ads on a lot of these searches, but...
  - There is still plenty of SEO optimized content!
- Abuser gets victim's passcode
  - Watches victim input the passcode
  - Just asks
  - Knows victim uses 123456
- And now when the victim is asleep...



How to Catch a cheating spouse using Android and [redacted]

To catch a cheating spouse you will need to get a hold of their phone. You will need their pass-code too to get into the phone. Using an android spy app like [redacted] you can catch your cheating spouse without needing to root the device either.

Here are the exact steps to putting [redacted] on an Android phone and catching your spouse cheating. It is how you can use [redacted] as an android spy apps for a cheating spouse. Go head and follow along on your own android phone or just use this as a guide when you have a few minutes alone with your spouses phone (*TIP: Do this while they are sleeping.*)

# Recovery is hard!



- It can be very hard to recover
  - These programs can be hard to detect:
    And on Android it often starts by rooting the phone
  - If you think you are a victim of stalkerware, trust your instincts!
  - A good guide here:
    https://stopstalkerware.org/information-for-survivors/
- Personal recommendation: Prevention is easier!
  - My phone uses a 5 word random passphrase which nobody else knows!
  - But I also enable fingerprint/face unlock
    - So the passphrase is seldom used but it is necessary for any of the abusive installations or configuration changes

# Another IPT Threat: Apple AirTags



- Stalkers/partners have long used trackers
  - $100 GPS tracker on a car which may last a week
- AirTags don't give a greater capability except for cost
  - But security is economics!
  - Works by using ***all*** iPhones as a reporting network:
    Anytime any phone records the beacon of an AirTag it reports to Apple where it is!
    - But a lot of cool cryptographic protections to keep this part of it from being a privacy nightmare
- Really really useful service
  - I have one now so I won't lose my keychain again!
- But at the same time, an easy-to-use-tracker
  - Stick one on the spouse's car and know where she is

# Some More on Tor



- Picking up from earlier...
- Why Tor Hidden Services are a plague

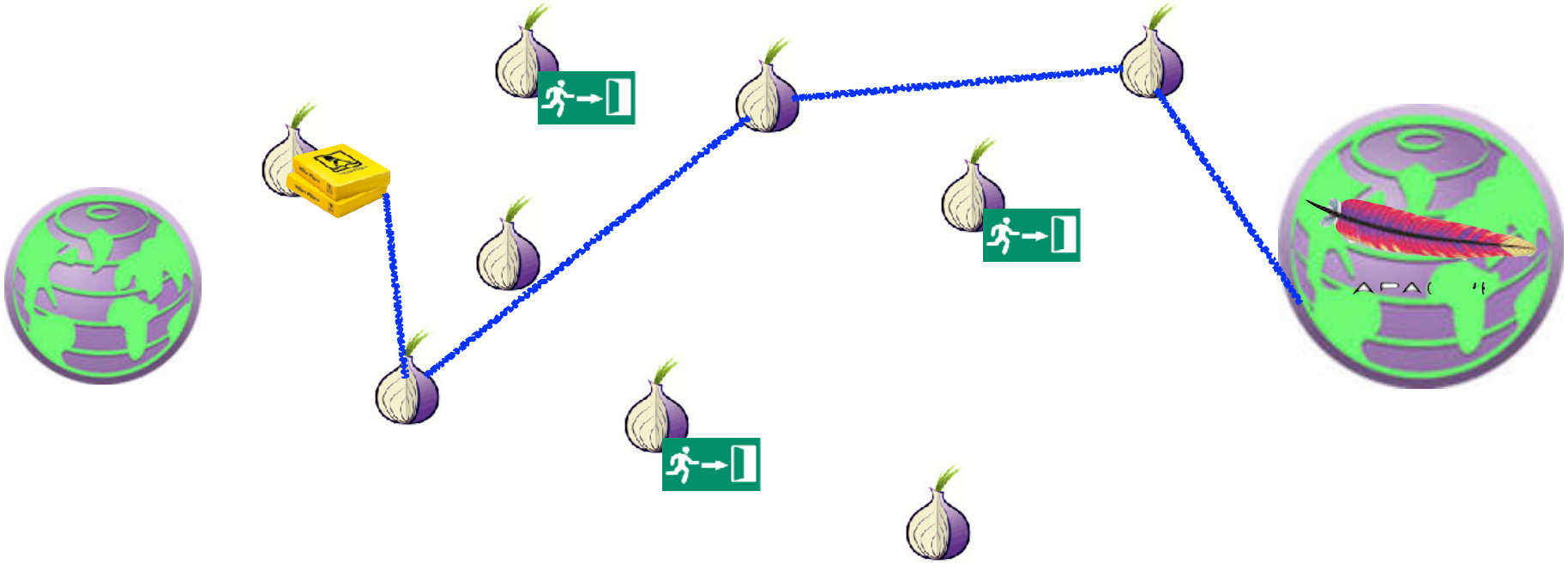
# Tor Browser is also used to access Tor Hidden Services aka .onion sites



- Services that only exist in the Tor network (the "Dark Web")
  - So the service, not just the client, has possible anonymity protection
- A hidden service name is a hash of the hidden service's public key
  - Used to be smaller: https://facebookcorewwwi.onion
    - In this case, Facebook spent a lot of CPU time to create something distinctive
  - Now larger: https://facebookwkhpilnemxj7asaniu7vnjjbiltxjqhye3mhbshg7kx5tfyd.onion
    - Change was designed to prevent one attack at the cost of forcing records of .onion domains to be unmemorizeable
- 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 when we finally create a final connection

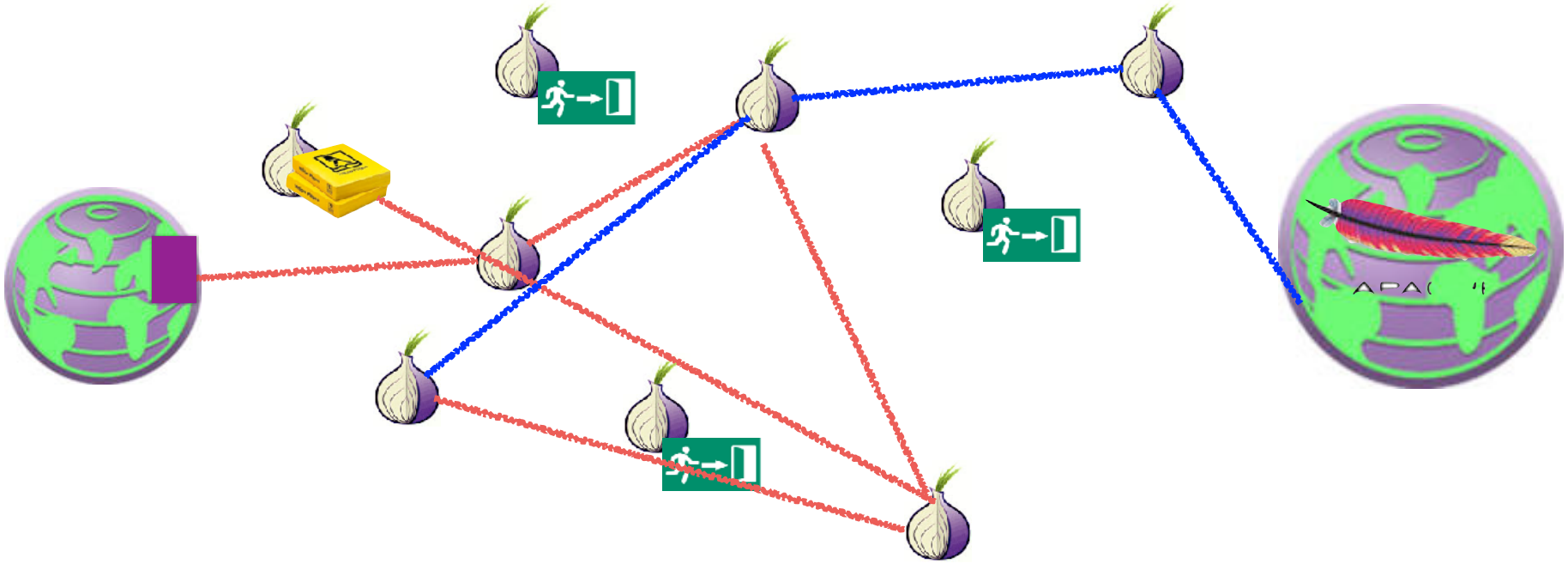
# Tor Hidden Service:
# Setting Up Introduction Point





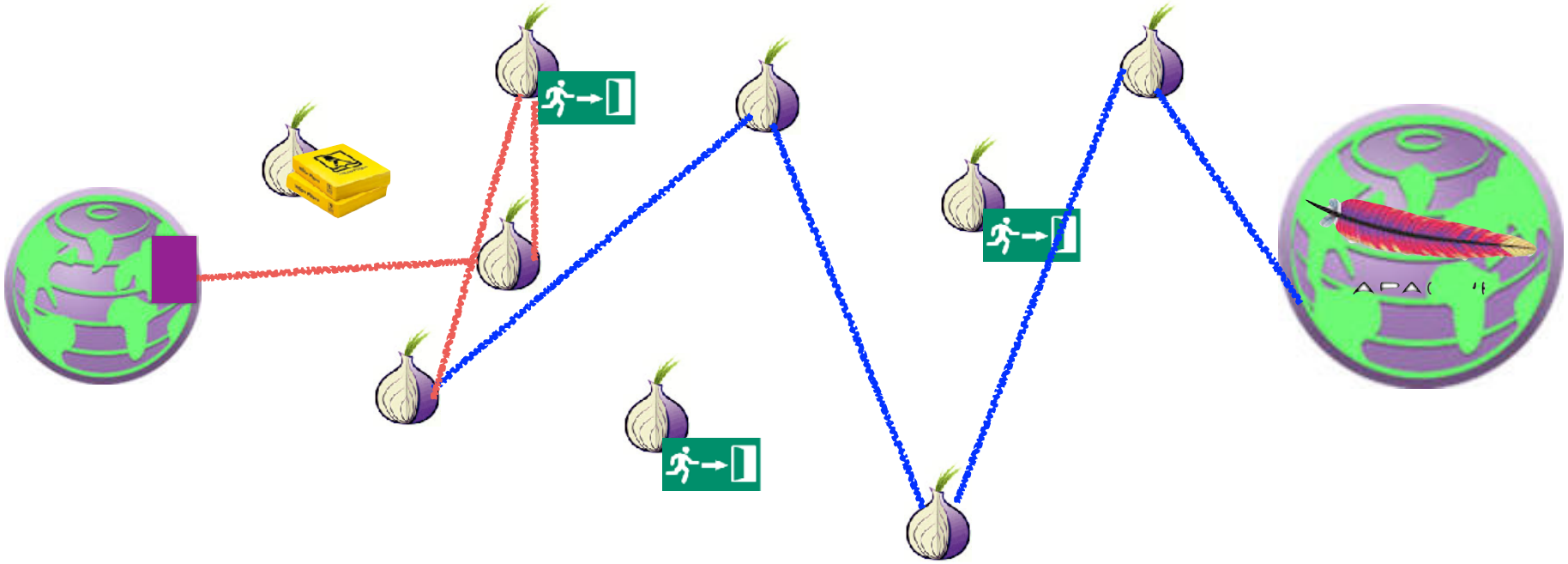
# Tor Hidden Service:
# Query for Introduction, Arrange Rendevous





# Tor Hidden Service: Rendevous and Data





Home | Alphabay Market | About Tor

pwoah7foa6au2pul.onion/index.php

Search

AlphaBay Market

Logged in as seanbridges
Balance: BTC 0.0000 / XMR 0.0000
Autoshop Logout

USD 573.53 CAD 735.76 EUR 508.38 AUD 753.03 GBP 437.84

HOME SALES MESSAGES ORDERS LISTINGS BALANCE FEEDBACK FORUMS API SUPPORT

Home

seanbridges
Joined: Aug 30, 2016
Trust level: Level 1
Total sales: USD 0.00
Total orders: USD 0.00

Search: Search

We **highly recommend** that you disable Javascript when viewing the marketplace for better security.

CC / ACCOUNT AUTOSHOP

Access the CC autoshop

Access the account autoshop

BROWSE CATEGORIES

| | |
|---|---|
| Fraud | 25438 |
| Drugs & Chemicals | 136335 |
| Guides & Tutorials | 10029 |

Featured Listings

[FE 100%]

▶ FRESH CC/CVV USA VISA/MASTERCARD /DISCOVER/AMEX (OLD MAGIC QUALITY/VALIDITY) - (New Stock OF CC +10K) - (Delivery Instantly) - (Always

[Bulk] USA HIGH LEVEL CC - VISA RANDOM CREDIT - BUSINESS/SIGNATURE /PLATINUM [AUTO FULFILL ON - DAILY SUPPORT] Browse store for more types and levels CCs!

**# 6329** - CVV & Cards - st0n3d

[MS] EDITABLE HQ TEMPLATES OF DOCUMENTS WORLDWIDE - GET VERIFIED EVERYWHERE INSTANTLY! - OVER 250 TEMPLATES TO CHOOSE FROM, SAMPLES ON ymhulceusuzrj3i5.onion

Double Your Bitcoins in ONE Day ! GUARANTEED! (2 in 1) $7000+ in 20 TWENTY MINUTES (50 + COPIES SOLD 100% POSITIVE FEEDBACK!)

**# 183848** - Other - BitcoinThief

Buy: USD 600.00

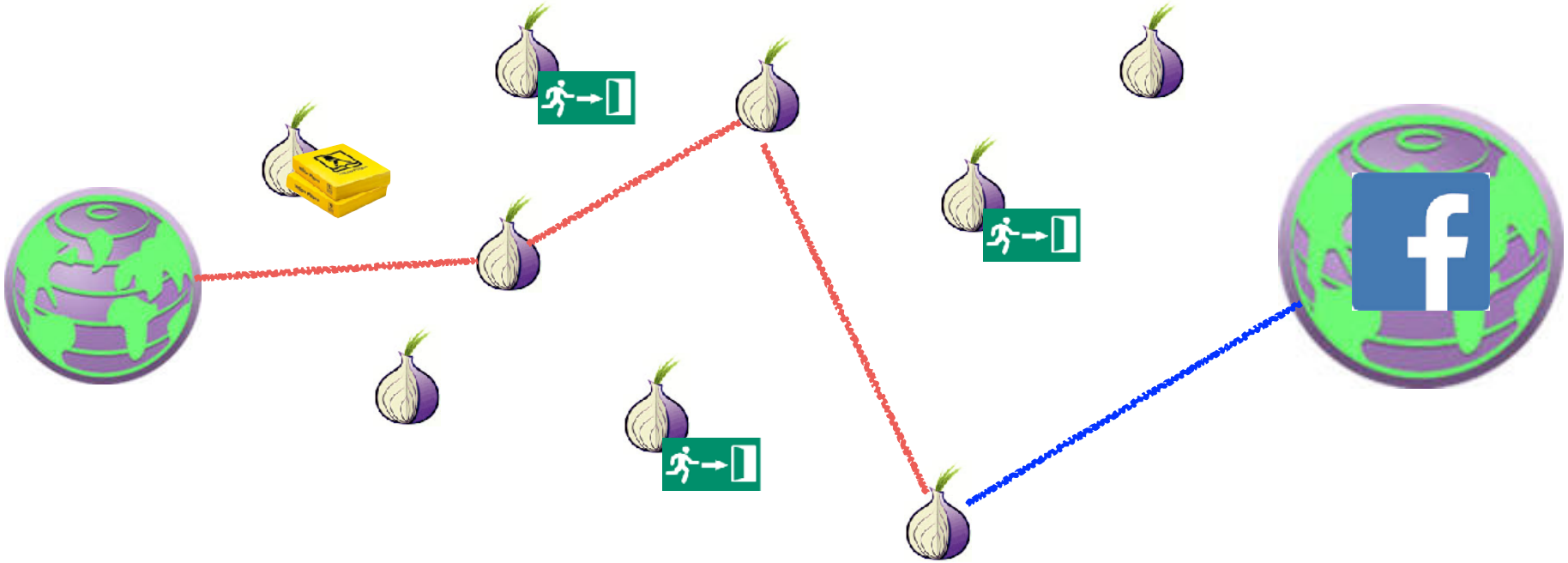
# Remarks…



- A hidden service wants to keep the 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
  - Facebook does this
- 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", CloudFlare, or suitable "foreign" machine rooms
- Dark Markets
  - Marketplaces based on Bitcoin or other alternate currency
- Cybercrime Forums
  - Hoping to protect users/administrators from the fate of earlier markets
- And worse...

# The Dark Market Concept



- Four innovations:
- A censorship-resistant payment (Bitcoin)
  - Needed because illegal goods are not supported by Paypal etc
- 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
  - The market is *shifting* trust, not eliminating it
- 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...
  - And even information about them is harder:
    Reddit no longer supports them, deepdotweb got busted...
    Leaving "Dread": Reddit as a Tor Hidden Service

# 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” (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 Now Once Again a ***SERIOUS*** Content Warning...



- The rest of the lecture is going to talk about the Elephant in the Room with Tor...
  Tor hidden services facilitate child abuse on an industrial scale
  - And the Tor project ***DOES NOT CARE***!
- I will be talking about actual cases and the scope of the problem
  - I studied these cases because they touched on significant policy issues surrounding searches and government hacking
- This will not be on the test beyond the following:
  "Nick hates Tor's Hidden Services with the fires of a thousand suns" and this is why...
  - And for the love of everything do not ever build something that has proved as loathsome as Tor

# February 2, 2020, Sunrise, Florida



- A team of FBI agents in the Violent Crimes Against Children division, including special agents Daniel Alfin and Laura Schwartzenberger, attempted to serve a search warrant as part of a CSAM (Child Sexual Abuse Material) investigation
  - Agents Alfin and Schwartzenberger were murdered by the suspect and three other agents injured
- I knew Dan professionally from his previous work involving CSAM and Tor...

# The "Playpen" Investigation



- In 2015 the FBI managed to identify and capture the server hosting the "Playpen" child exploitation site:
  Daniel Alfin was one of the lead investigators
- Playpen operated as a hidden service image board for posting CSAM
  - 250,000+ registered users, 20,000+ images
  - This represents thousands of abused children!
- But the site operator's are not the only problem...
  The site users are a problem
  - A significant number are "hands-on" abusers:
    Both because of their predilections and because creating new "content" is currency in these communities

# To Deanonymize the Users...



- The FBI took over Playpen and ran the site for 2 weeks
- During those two weeks...
  - Disabled posting of new content, but continued to serve old content...
  - And added a post-login bonus: A zero-day attack on the Tor Browser Bundle
- Exploit payload: "phone home"
  - Not a general purpose shellcode, instead collect Ethernet Addresses, current user, and similar identifying information and contact an FBI server
- FBI calls this a NIT: "Network Investigation Technique"
- They had a warrant:
  - It described with particularity what it would search for, how it would work conceptually, etc...

# Significant Impact



- 25 producers prosecuted, 350 arrests in the US alone
- Nearly 300 children identified or rescued from abusive situations worldwide, over 50 in the US
- But also two significant controversies:
- Was the warrant actually valid?
  - Answer ended up being "No, but 'good faith'....":
    At the time there was no way to write a warrant that says "I want to search these computers, but we don't know where they are!"
- What should defendants be able to examine with regard to the exploit?
  - Answer largely ended up being "No, not actually relevant"
  - An in the weeds discussion by Susan Hennesey and myself is available here:
    https://www.lawfareblog.com/judicial-framework-evaluating-network-investigative-techniques

# The Problem:
# These are communities of abusers



- There have been others both before and since
  - Before Playpen there was "Freedom Hosting": hosted close to 50 CSAM sites.
    If you want to be nauseated read the Freedom Hosting NIT warrant application
  - In 2017 an FBI style NIT was deployed on "GiftBox" (probably by the French):
    But it was captured by a site user and posted to Reddit...
  - In 2018 "Welcome to Video" was busted: Pay for CSAM with Bitcoin!
    Again, if you want to vomit read the indictments or the recent article online in Wired
- Communities create dangerous cycles of normalization
  - And larger communities are more dangerous:
    See more mild versions that happened on Reddit with TheDonald, jailbait, creepshots, etc...
    - Self reinforcement behavior: "Its normal because others in the community do it" and the community becomes self justifying
    - See the "Jailbait" analysis in ***Twitter and Tear Gas***
  - Drives to extremes: Over the past decade, the age of CSAM victims has basically gotten younger...
    To the point where average age really can't get much lower

# The Problem #2:
# The Tor Project ***JUST DOES NOT CARE!***



- They treat this as "collateral damage" with a series of excuses. Here are actual justifications by Roger Dingledine (Founder):
- "But hidden services are in their infancy"
  - And in the same presentation talk about it being a 10 year old idea...
- "But hidden services are end-to-end authenticated"
  - Yeah, there is this thing call TLS...
- "But hidden services work through NATs"
  - Yeah, there is this thing called uPNP: You ask the NAT to allow inbound connections
  - Oh, or just use EC2...
- "But dissidents..."
  - Well, running Tor is very noticeable...
  - Plus you can "arbitrage host": Want to piss off China? Host in the US. Piss off the US? Host in Russia...
- "But Facebook/SecureDrop/Etc... has an onion service"
  - Uh, they don't actually need to be hidden! And work better when they aren't!

# And A Different Problem: Grooming



- I never encountered Agent Schwartzenberger, but this was her specialty...
  people who use electronic chat to groom child victims for exploitation
- In unencrypted chats, the chat-provider can ***theoretically*** try to detect this behavior
  - A case where classic Machine Learning tends to work pretty well if the results are human-reviewed for false-positives
- The problem grows even harder when dealing with encrypted chats
  - Since there is no longer a central server that can try to detect the behavior...
  - And the developers would probably resist adding an AI-snitch to the client

# So Remember:
# Child Abuse IS REAL



- Too often those in favor of security/privacy view claims about child abuse and CSAM as disingenuous...
  - It isn't helped that those wanting encryption backdoors will use claims about child abuse and CSAM in a disingenuous manner!
- But these problems are real
  - Grooming over messenger systems is a serious problem
    - Usually starting over some open system where children frequent...
    - Before moving onto encrypted messengers like iMessage and Facebook Messenger
  - Tor hidden services have created a CSAM "industry"