Vulnerabilities in Tor: (past,) present, future

Roger Dingledine
The Tor Project
https://www.torproject.org/
Outline

- Crash course on Tor
- Solved/solvable problems
- Tough ongoing issues, practical
- Tough ongoing issues, research
- Future
Tor: Big Picture

- Freely available (Open Source), unencumbered.
- Comes with a spec and full documentation: Dresden and Aachen implemented compatible Java Tor clients; researchers use it to study anonymity.
- 1500 active relays, 200000+ active users, >1Gbit/s.
- Official US 501(c)(3) nonprofit. Eight full-time developers (!), dozens more dedicated volunteers.
- Funding from US DoD, Electronic Frontier Foundation, Voice of America, a French NGO, Google, NLnet, Human Rights Watch, ...you?
Anonymity serves different interests for different user groups.

“'It's privacy!'”
Anonymity serves different interests for different user groups.

- Private citizens: “It's privacy!”
- Businesses: “It's network security!”
Anonymity serves different interests for different user groups.

“It's traffic-analysis resistance!”

Governments

Anonymity

Private citizens

“It's privacy!”

“It's network security!”

Businesses
Anonymity serves different interests for different user groups.

- Blockaded users: "It's reachability!"
- Governments: "It's traffic-analysis resistance!"
- Private citizens: "It's privacy!"
- Businesses: "It's network security!"
The simplest designs use a single relay to hide connections.

(example: some commercial proxy providers)
But a single relay (or eavesdropper!) is a single point of failure.
So, add multiple relays so that no single one can betray Alice.
A corrupt first hop can tell that Alice is talking, but not to whom.
A corrupt final hop can tell that somebody is talking to Bob, but not who.
Alice makes a session key with R1
...And then tunnels to R2...and to R3
The basic Tor design uses a simple centralized directory protocol.

S1

S2

S3

Servers publish self-signed descriptors.

Trusted directory

Trusted directory

 Authorities publish a consensus list of all descriptors

cache

Alice downloads consensus and descriptors from anywhere
Outline

• Crash course on Tor
• *Tough ongoing issues, practical*
• Tough ongoing issues, research
• Future
Snooping on Exit Relays (1)

- Lots of press last year about people watching traffic coming out of Tor. (Ask your lawyer first...)
- Tor hides your location; it doesn't magically encrypt all traffic on the Internet.
- Though Tor *does* protect from your local network.
Snooping on Exit Relays (2)

- https as a “premium” feature
- Should Tor refuse to handle requests to port 23, 109, 110, 143, etc by default?
- Torflow / setting plaintext pop/imap “traps”
- Need to educate users?
- Active attacks on e.g. gmail cookies?
- Some research on exit traffic properties is legitimate and useful. How to balance?
Who runs the relays? (1)

- At the beginning, you needed to know me to have your relay considered “verified”.
- We've automated much of the “is it broken?” checking.
- Still a tension between having lots of relays and knowing all the relay operators
Who runs the relays? (2)

- What if your exit relay is running Windows and uses the latest anti-virus gadget on all the streams it sees?
- What if your exit relay is in China and you're trying to read BBC?
- What if your exit relay is in China and its ISP is doing an SSL MitM attack on it? (What if China owns a CA?)
Who runs the relays? (3)

- What happens if ten Tor relays show up, all on 149.9.0.0/16, which is near DC?
- "EnforceDistinctSubnets" config option to use one node per /16 in your circuit (Tor 0.1.2.1-alpha, 27 August 2006)
- No more than 2 relays on one IP address (Tor 0.2.0.3-alpha, 29 July 2007)
- How about ASes? IXes? Countries?
Tor Browser Bundle traces

- We want to let you use Tor from a USB key without leaving traces on the host
- "WINDOWS/Prefetch" trace
- Windows explorer's "user assist" registry entry
- Vista has many more?
Application-level woes (1)

• Javascript refresh attack
• Cookies, History, browser window size, user-agent, language, http auth, ...
• Mostly problems when you toggle from Tor to non-Tor or back
• Mike Perry's Torbutton 1.2.0 tackles many of these (30 July 2008)
Some Firefox privacy bugs remain

- No way to configure/spoof time zones
- “Livemarks” / “Live bookmarks” does a lookup over Tor when Firefox starts.
- Client-side SSL certs are messy to isolate (Firefox happily sends them to the remote website even via Tor)
- The TLS ClientHello message in FF2 uses uptime for the “time” variable!
Application-level woes (2)

• Some apps are bad at obeying their proxy settings.
• Adobe PDF plugin. Other plugins. Extensions. Especially Windows stuff.
Transparent proxying

- Easy to do in Linux / BSD: iptables/pf, getsockopt()/getsockname(), done.
- Put Tor client in a Linux QEMU running inside Windows. Then intercept outgoing traffic from Windows apps. Or,
- Put Tor client and apps inside a Linux QEMU, and launch it from Windows.
Filtering connections to Tor

- By blocking the directory authorities
- By blocking all the relay IP addresses in the directory
- By filtering based on Tor's network fingerprint
- By preventing users from finding the Tor software
Outline

• Crash course on Tor
• Tough ongoing issues, practical
• **Tough ongoing issues, research**
• Future
Traffic confirmation

- If you can see the flow into Tor and the flow out of Tor, simple math lets you correlate them.
- Defensive dropping (2004)? Adaptive padding (2006)?
Website fingerprinting

- If you can see an SSL-encrypted link, you can guess what web page is inside it based on size.
- Does this attack work on Tor? “maybe”
- Considering multiple pages (e.g. via hidden Markov models) would probably make the attack even more effective.
Clogging / Congestion attacks (1)

- Murdoch-Danezis attack (2005) sent constant traffic through every relay, and when Alice made her connection, looked for a traffic bump in three relays.
- Couldn't identify Alice – just the relays she picked.
Clogging / Congestion attacks (2)

- Hopper et al (2007) extended this to (maybe) locate Alice based on latency.
- Chakravarty et al (2008) extended this to (maybe) locate Alice via bandwidth tests.
- Evans et al (2009?) showed the original attack doesn't work anymore (too many relays, too much noise) – but “infinite length circuit” makes it work again?
Profiling at exit relays

- Tor reuses the same circuit for 10 minutes before rotating to a new one.
- (It used to be 30 seconds, but that put too much CPU load on the relays.)
- If one of your connections identifies you, then the rest lose too.
- What's the right algorithm for allocating connections to circuits safely?
Declining to extend

• Tor's directory system prevents an attacker from spoofing the whole Tor network.
• But your first hop can still say “sorry, that relay isn't up. Try again.”
• Or your local network can restrict connections so you only reach relays they like.
Outline

• Crash course on Tor
• Tough ongoing issues, practical
• Tough ongoing issues, research
• *Future*
Traffic correlation

• It's just going to get better.
• E.g., maybe somebody publishes mrtg graphs or other apparently innocent data, and that turns out to be enough?
• Or smoke ping data for all the relays?
Countries blocking Tor network

- Blocking the website is a great start
- Eventually, they'll block the Tor relays, and bridges will be needed
- Then the arms race for blocking bridge relays will start.
Data retention

• “Service providers” must log “stuff”
• It means major ISPs have to remember which customer had which IP address?
• GPF lawyer says doesn't apply to non-commercial service providers anyway?
• The police I talked to in Stuttgart said they didn't ask for this law.
Data retention

- Some modifications we can make to the Tor design to resist logging at ISPs.
- There will be no logging inside Tor.
- CCC is going to challenge this law. To start, they're going to use the CCC-Tor donations from 2008. They could use a lot more!
Last thoughts

- Pretty much any Tor bug seems to turn into an anonymity attack.
- Many of the hard research problems are attacks against all low-latency anonymity systems. Tor is still the best that we know of -- other than not communicating.
- People find things because of the openness and thoroughness of our design, spec, and code. We'd love to hear from you.
Debian RNG flaw

- [Addressed in Tor 0.2.0.26-rc, 13 May 2008]
- 300 out of ~1500 Tor relay identity keys were bad.
- Logged traffic breakable too--if the client was Debian, or if it used only Debian relays!
- Three out of the six v3 dir authority keys were bad. Four would have really sucked.