What is Tor?

• Volunteer run relay network designed for privacy, anonymity, and censorship resistance.

• Tor relays TCP connections ("streams")
  - Multiplexed on encrypted paths ("circuits")

• Nodes connected by TLS/SSL

• Circuits route through 3 nodes
  - "Guard", "relay", "exit"
Tor Routing
Classes of Attack

- **Passive attacks**
  - Packet and connection timing correlation
  - Fingerprinting of traffic/usage patterns
  - “Intersection Attacks” of multiple attributes of users

- **Active attacks**
  - Lying about bandwidth to get more traffic
  - Failing circuits to bias node selection
  - Modifying application layer traffic at exit
Position of Attack

• **Internal**
  - Node operator
  - Can differentiate circuits at guard and relay.
  - Able to differentiate streams per circuit at exit

• **External**
  - ISP or Echelon-style adversary
  - Assumed to be unable to see inside TLS streams
  - Likely frustrated to a large degree by running Tor as both node and client
Attack Points

Surveillance, confiscation

Active Network Attacks
(Blocking, guard node bias)

Active Circuit Failure Attacks

Application Layer Attacks

Client misconfiguration, other user error.

Guard Node

Relay Node

Exit Node

Timing Correlation Attacks
(External or Internal, Active or Passive);
Intersection Attacks
Approaches to Security

- Verify node operators (Ha!)
- Path selection hacks
- Scan nodes for modification/reliability
- “Tor up from the floor up”
- Secure the applications (different threat model)
- Improve network speed and usability
Path Selection Hacks

• /16 hack: No two nodes from same /16 netmask
  - Many ISPs have disjoint IP ranges..

• Guard nodes
  - Essentially a time-tradeoff of risk
    • Difficult to do right. Typically still rotate
  - Avoids long-term fingerprinting
  - Without rotation, can deter intimidation attacks
  - Foil “repetitive fetch” application layer attacks
Tor Routers and LiveCDs

• JanusVM, Anonym.OS, others
  - “Tor up from the floor up”
  - Addresses application-level attacks to bypass Tor
  - Blocks UDP

• Major flaw: Circuit reuse -> app correlation
  - Windows update, other ID-based software updates
  - AIM, ssh, email usage of different “nyms”
  - Media players checking recommended music, etc etc
Centralized Network Scanning

- Tor control port is fun stuff
- Snakes on a Tor and TorFlow
  - Verifies md5 sums of googled URLs
  - Also verifies node reliability+bandwidth
- Works against incompetent+blanket adversaries
  - Actually found some broken+malicious nodes
- Does not work against targeted adversaries
- Vulnerable to detection
Decentralized Network Scanning

- **Client-based:**
  - Use reliability averages from TorFlow
  - Alert user if guard node fails more than X% circuits
  - Measure observed bandwidth/latency of nodes

- **Node-based:**
  - Gather statistics on average capacity and queue lengths to peers, compare to node rankings
  - Report major deviations or use as balancing feedback loop.
Securing the Application Layer

- Tor has a superset of the threat model most applications are written for.
  - No UDP!
  - Unique identifiers are bad
  - Proxy settings must be sacrosanct
  - Location information must not be transmitted
  - Updates are dangerous. Hostile network.
Tor's Web Attack Profile

1. Bypassing proxy settings
2. Correlation of Tor vs Non-Tor
3. History disclosure
4. Location information
5. Misc Anonymity set reduction
6. History records
Solution: Improved TorButton

- Disable plugins while Tor is enabled
- Isolate dynamic content per Tor load state
- Cookie jars/cookie clearing
- Cache management
- History management
- User agent spoofing during Tor
- Javascript Hooking
TorButton Demo

- http://gemal.dk/browserspy/basic.html
- http://gemal.dk/browserspy/css.html
- http://gemal.dk/browserspy/date.html
- http://gemal.dk/browserspy/plugins.html
- http://ha.ckers.org/weird/CSS-history.cgi
- http://www.tjkdesign.com/articles/css%20pop%20ups/5.asp
Interesting Technical Details

- Context issues
- Tab tagging
- XPCOM hooking and XPCOM policies
- Javascript hooking
Improving Speed and Usability

• Key component of Tor security: Large userbase

• Users want speed and ease of use
  - Many do not need as much anonymity
  - Two hop proposal (semi-controversial)
  - Intelligent path selection

• Tor network is unbalanced
  - Guard node issues (bug #440)
  - Exit selection issues
Final Thoughts

- Tor security ≠ Internet security
  - Superset, actually
  - Adversary has different goals
  - Many apps do not consider privacy vulnerabilities as real vulnerabilities
Credits+Contributions

Scott Squires (Original TorButton Author)
Collin Jackson (History blocking+Cookie jars)
Johannes Renner (TorFlow contributions+research)
Nick & Roger (Advice, Tor in general)
Dave, Nitin G, Thom (Advice, moral support)
“What can I do to help Tor?”

• Extra bandwidth? Run a node!
  – See conference CD for Linux 'tc' prioritization script
  – No need to impact your own traffic flows

• Post patches/plugins to your favorite apps to protect against info disclosure.
  – Work to raise awareness that privacy issues should be considered as part of security measures