Low-Resource Routing Attacks Against Tor

Client

![Onion icon]

Server

Kevin Bauer ¹  Damon McCoy ¹  Dirk Grunwald ¹  Tadayoshi Kohno ²  Douglas Sicker ¹

University of Colorado ¹  University of Washington ²

Workshop on Privacy in the Electronic Society
Alexandria, VA, USA  October 29, 2007
Striking a Balance: Performance vs. Anonymity

Two observations:
1. Perfect anonymity is desirable, but not practical
2. Users demand certain performance standards from any anonymous network

Tor attempts to balance the traffic load to ensure a relatively low latency and high throughput service, but:

What impact does this attempt at load balancing have on Tor’s ability to provide strong anonymity?
Talk Outline

- Background
- Compromising Anonymity
- Experimental Setup and Results*
- Attack Extensions
- Proposed Solutions
- Future Work and Conclusion

* Using stable Tor release in August 2006
To easily correlate client requests to a server’s response, an adversary must control the entry and exit routers for a circuit.
Traditionally, it has been assumed that an attacker can control an entry/exit router combination for a target circuit with a very low probability.

Assuming that routers are chosen uniformly at random, this probability is \((c/n)^2\), where the attacker controls \(c\) routers in a network of \(n\) total routers.
Related Work: Previous Attacks Against Tor

Two classes of attacks against Tor:

Traffic Analysis

- Low-Cost Traffic Analysis of Tor
  [Murdoch and Danezis] S&P ‘05

- Sampled Traffic Analysis by Internet-Exchange-level Adversaries
  [Murdoch and Zielinski] PET ‘07

Locating Hidden Services

- Locating Hidden Servers
  [Øverlier and Syverson] S&P ‘06

- Hot or Not: Revealing Hidden Services by their Clock Skew
  [Murdoch] CCS ‘06

Our work
Ọverlier and Syverson demonstrated that an adversary can force new circuits to be built until it controls an entry router.

*Entry guards* were introduced in May 2006 to limit the likelihood of a malicious Tor router existing at the entry position.
Routing in Tor
Part 1: Entry Guard Selection

Tor routers advertise their own bandwidth capabilities and uptimes to the trusted directory servers.

Directory servers mark routers as a potential “Guard” if they advertise uptime and bandwidth at or above the median for all routers; only these routers may be entry guards.
Routing in Tor

Part 2: Non-Entry Node Selection

Routers for the middle/exit positions are chosen to balance the traffic according to each node’s bandwidth history; this allows for relatively low latency and high throughput service.

Key feature: Nodes **claiming** high bandwidth are chosen most frequently.
Adversary: Our attacks assume a low-bandwidth (i.e., residential cable modem) non-global passive adversary controlling $n >> c > 1$ malicious Tor routers

Client: The clients run only as Tor proxies (default setting) and join the Tor network for the first time; the default .torrc file is used
Compromising Anonymity: Basic Attack

Malicious Tor Routers

Real bandwidth: 1.5MB/s
Real uptime: 100 days

Trusted Directory Server

Very simple: An adversary deploys c high-bandwidth ("fast") and high-uptime ("stable") Tor routers; under routing model, increase likelihood of controlling an entry/exit pair
Compromising Anonymity: Resource Reduction

Any router can claim up to 1.5MB/s and high uptimes

Currently, Tor directory servers do not verify bandwidth and uptime claims

Focus malicious router’s real bandwidth on accepting new clients and/or targeting specific clients or destinations
New Circuit Linking Technique

Tor circuits are built deterministically; steps 1-3 must occur in order.

This allows circuits to be correlated before any payload data is sent.

Also allows for circuits to be quickly disrupted, if only one node is malicious on the circuit.
Experimental Methodology

We evaluated the resource-reduced attack on two isolated Tor deployments with 40 and 60 benign routers using Tor 0.1.1.23 (August 2006)

<table>
<thead>
<tr>
<th>Tier</th>
<th>Real Tor</th>
<th>40 Node</th>
<th>60 Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>996 KB</td>
<td>38</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>621 KB</td>
<td>43</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>362 KB</td>
<td>55</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>111 KB</td>
<td>140</td>
<td>13</td>
<td>20</td>
</tr>
<tr>
<td>29 KB</td>
<td>123</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>20 KB</td>
<td>21</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>103.9 MB</td>
<td>10.4 MB</td>
<td>15.7 MB</td>
</tr>
</tbody>
</table>

Evaluating the bandwidth distribution from the real network provided a realistic model for the experimental deployments.

We generated traffic using HTTP requests from 60 (40 node network) and 90 (60 node network) clients for two hours.
Malicious Router Configurations

Varied the number of malicious routers (2/42, 4/44 and 3/63 and 6/66); malicious nodes advertised 1.5MB/s and large uptimes and were limited to 20KB/s

Malicious routers contributed 0.3-0.8% of network’s bandwidth, but advertised 22-36% of the total bandwidth

Modified malicious Tor router code to prioritize circuit-building requests over data packets
Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Experiments</th>
<th></th>
<th>Experiments</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2/42</td>
<td>4/44</td>
<td>3/63</td>
<td>6/66</td>
</tr>
<tr>
<td>Random Selection</td>
<td>0.12%</td>
<td>0.63%</td>
<td>0.15%</td>
<td>0.70%</td>
</tr>
<tr>
<td>Experimental</td>
<td>8.90%</td>
<td>33.55%</td>
<td>11.06%</td>
<td>46.36%</td>
</tr>
<tr>
<td>Improvement</td>
<td>7.565%</td>
<td>5.190%</td>
<td>7.097%</td>
<td>6.530%</td>
</tr>
</tbody>
</table>

“Random Selection” expectation based on $(c/n)^2$

The experimental results show a significant increase over the expectation if routers were chosen uniformly at random.

Tor’s load balancing optimizations introduce opportunities to reduce the system’s anonymity.
Attack Extensions: Quick Overview

1. Compromise existing clients (i.e., entry guards already chosen)

2. Target specific clients or destination servers

3. If only the entry guard is malicious, embed a watermark that can be detected at specific destination servers

4. Replace all honest entry guards with malicious routers
Attacking Entry Guards

There is currently **no limit** to how many routers may run on a single host

1. Run several routers on a local machine, advertising high bandwidths and uptimes

2. The *global* median uptime value will increase to that of the malicious routers

3. Now only malicious nodes can be entry guards

**Resource claims must be trusted**
Solutions: Raising the Bar

1. Resource advertisement verification
   Idea: The directory servers should monitor uptime (easy) and bandwidth (more difficult)

   Tor specification proposal #107: Uptime sanity checking
   Tor specification proposal #108: Base ‘Stable’ flag on mean time between failures

2. Mitigating Sybil attacks
   Idea: Limit the number of Tor routers that may be run at an IP address

   Tor specification proposal #109: Two routers per IP

Each proposal will be implemented in a future Tor release
Open Questions and Future Work

- Evaluate the attacks when scaled to larger and more realistic Tor networks
- Derive a more realistic analytical expectation of the attack’s success
- Investigate resource verification techniques
  - Centralized and distributed verification
  - Reputation systems
- Explore alternative routing strategies
  - Proximity awareness (i.e., DHTs)
  - Loose routing (i.e., Crowds)
  - Local reputation-based routing
Summary and Conclusion

Problem: Load balancing has left Tor open to end-to-end traffic correlation attacks by a very weak adversary.

Traffic can be linked using only circuit-building packets, allowing anonymity compromise before receiving payload data.

Entry guards, added to protect circuits from these attacks, are themselves vulnerable to attack.

Verifying resource claims and mitigating Sybil attacks can help reduce the risk posed by the resource-constrained attacker.