Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries

Aaron Johnson¹  Chris Wacek²  Rob Jansen¹  Micah Sherr²  Paul Syverson¹

¹ U.S. Naval Research Laboratory, Washington, DC  ² Georgetown University, Washington, DC

MPI-SWS
July 29, 2013
Summary: What is Tor?
Summary: What is Tor?

Tor is a system for anonymous communication.
Summary: What is Tor?

Tor is a system for anonymous communication.

Popular

Over 500,000 daily users and 2.4GiB/s aggregate
Summary: Who uses Tor?
Summary: Who uses Tor?

- Individuals avoiding censorship
- Individuals avoiding surveillance
- Journalists protecting themselves or sources
- Law enforcement during investigations
- Intelligence analysts for gathering data
Summary: Tor’s Big Problem
Summary: Tor’s Big Problem
Summary: Tor’s Big Problem
Summary: Tor’s Big Problem
Summary: Tor’s Big Problem

Traffic Correlation Attack
Summary: Tor’s Big Problem

Traffic Correlation Attack

- Congestion attacks
- Throughput attacks
- Latency leaks

- Website fingerprinting
- Application-layer leaks
- Denial-of-Service attacks
Summary: Our Contributions
Summary: Our Contributions

1. Empirical analysis of traffic correlation threat
2. Develop adversary framework and security metrics
3. Develop analysis methodology and tools
Overview

• Summary
• Tor Background
• Tor Security Analysis
  o Adversary Framework
  o Security Metrics
  o Evaluation Methodology
  o Node Adversary Analysis
  o Link Adversary Analysis
• Future Work
Overview

• Summary
• Tor Background
• Tor Security Analysis
  o Adversary Framework
  o Security Metrics
  o Evaluation Methodology
  o Node Adversary Analysis
  o Link Adversary Analysis
• Future Work
Background: Onion Routing

Users | Onion Routers | Destinations
Background: Onion Routing
Background: Onion Routing

Users → Onion Routers → Destinations
Background: Onion Routing
Background: Onion Routing
Background: Using Circuits
1. Clients begin all circuits with a selected guard.
1. Clients begin all circuits with a selected *guard*.
2. Relays define individual *exit policies*.
1. Clients begin all circuits with a selected *guard*.
2. Relays define individual *exit policies*.
3. Clients multiplex *streams* over a circuit.
1. Clients begin all circuits with a selected guard.
2. Relays define individual exit policies.
3. Clients multiplex streams over a circuit.
4. New circuits replace existing ones periodically.
Overview

• Summary
• Tor Background
• Tor Security Analysis
  o Adversary Framework
  o Security Metrics
  o Evaluation Methodology
  o Node Adversary Analysis
  o Link Adversary Analysis
• Future Work
Adversary Framework
Adversary Framework
Adversary Framework
Adversary Framework
Adversary Framework

Resource Types
- Relays
- Bandwidth
- Autonomous Systems (ASes)
- Internet Exchange Points (IXPs)
- Money
# Adversary Framework

<table>
<thead>
<tr>
<th>Resource Types</th>
<th>Resource Endowment</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Relays</td>
<td>• Destination host</td>
</tr>
<tr>
<td>• Bandwidth</td>
<td>• 5% Tor bandwidth</td>
</tr>
<tr>
<td>• Autonomous Systems (ASes)</td>
<td>• Source AS</td>
</tr>
<tr>
<td>• Internet Exchange Points (IXPs)</td>
<td>• Equinix IXPs</td>
</tr>
<tr>
<td>• Money</td>
<td></td>
</tr>
</tbody>
</table>
## Adversary Framework

<table>
<thead>
<tr>
<th>Resource Types</th>
<th>Resource Endowment</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Relays</td>
<td>• Destination host</td>
<td>• Target a given user’s communication</td>
</tr>
<tr>
<td>• Bandwidth</td>
<td>• 5% Tor bandwidth</td>
<td>• Compromise as much traffic as possible</td>
</tr>
<tr>
<td>• Autonomous Systems (ASes)</td>
<td>• Source AS</td>
<td>• Learn who uses Tor</td>
</tr>
<tr>
<td>• Internet Exchange Points (IXPs)</td>
<td>• Equinix IXPs</td>
<td>• Learn what Tor is used for</td>
</tr>
<tr>
<td>• Money</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overview

• Summary
• Tor Background
• Tor Security Analysis
  o Adversary Framework
  o Security Metrics
  o Evaluation Methodology
  o Node Adversary Analysis
  o Link Adversary Analysis
• Future Work
Security Metrics

Prior metrics
Security Metrics

Prior metrics
1. Probability of choosing bad guard and exit
   a. $c^2 / n^2$: Adversary controls $c$ of $n$ relays
   b. $ge$: $g$ guard and $e$ exit BW fractions are bad
Prior metrics

1. Probability of choosing bad guard and exit
   a. $c^2 / n^2$: Adversary controls $c$ of $n$ relays
   b. $ge$: $g$ guard and $e$ exit BW fractions are bad

2. Probability some AS/IXP exists on both entry and exit paths (i.e. *path independence*)
Security Metrics

Prior metrics
1. Probability of choosing bad guard and exit
   a. $c^2 / n^2$: Adversary controls $c$ of $n$ relays
   b. $ge$: $g$ guard and $e$ exit BW fractions are bad
2. Probability some AS/IXP exists on both entry and exit paths (i.e. path independence)
3. $g_t$: Probability of choosing malicious guard within time $t$
Security Metrics

Principles

1. Probability distribution
2. Measure on human timescales
3. Based on adversaries
Security Metrics

Principles
1. Probability distribution
2. Measure on human timescales
3. Based on adversaries

Metrics
1. Probability distribution of time until first path compromise
2. Probability distribution of number of path compromises for a given user over given time period
Overview

• Background

• Onion Routing Security Analysis
  o Problem: Traffic correlation
  o Adversary Model
  o Security Metrics
  o Evaluation Methodology
  o Node Adversary Analysis
  o Link Adversary Analysis

• Future Work
TorPS: The Tor Path Simulator

Network Model

User Model → Streams → Tor → Client Software Model → Circuit mappings

Relay statuses
TorPS: The Tor Path Simulator

Network Model

User Model

Streams

Client Software Model

Relay statuses

Stream ➔ Circuit mappings
TorPS: User Model

- Gmail/GChat
- Gcal/GDocs
- Facebook
- Web search
- IRC
- BitTorrent

20-minute traces
TorPS: User Model

Gmail/GChat
Gcal/GDocs
Facebook
Web search
IRC
BitTorrent

20-minute traces

Typical
TorPS: User Model

- Gmail/GChat
- Gcal/GDocs
- Facebook
- Web search
- IRC
- BitTorrent

20-minute traces

Typical

Session schedule

- One session at 9:00, 12:00, 15:00, and 18:00 Su-Sa
- Repeated sessions 8:00-17:00, M-F
- Repeated sessions 0:00-6:00, Sa-Su
TorPS: User Model

Gmail/GChat
Gcal/GDocs
Facebook
Web search
IRC
BitTorrent

20-minute traces

Typical
Worst Port (6523)
Best Port (443)

Session schedule

One session at
9:00, 12:00, 15:00, and 18:00
Su-Sa

Repeated sessions
8:00-17:00, M-F

Repeated sessions
0:00-6:00, Sa-Su
### TorPS: User Model

<table>
<thead>
<tr>
<th>Rank</th>
<th>Port #</th>
<th>Exit BW %</th>
<th>Long-Lived</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8300</td>
<td>19.8</td>
<td>Yes</td>
<td>iTunes?</td>
</tr>
<tr>
<td>2</td>
<td>6523</td>
<td>20.1</td>
<td>Yes</td>
<td>Gobby</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>25.3</td>
<td>No</td>
<td>(SMTP+1)</td>
</tr>
<tr>
<td>65312</td>
<td>993</td>
<td>89.8</td>
<td>No</td>
<td>IMAP SSL</td>
</tr>
<tr>
<td>65313</td>
<td>80</td>
<td>90.1</td>
<td>No</td>
<td>HTTP</td>
</tr>
<tr>
<td>65314</td>
<td>443</td>
<td>93.0</td>
<td>No</td>
<td>HTTPS</td>
</tr>
</tbody>
</table>

Default-accept ports by exit capacity.
### TorPS: User Model

<table>
<thead>
<tr>
<th>Model</th>
<th>Streams/week</th>
<th>IPs</th>
<th>Ports (#s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical</td>
<td>2632</td>
<td>205</td>
<td>2 (80, 443)</td>
</tr>
<tr>
<td>IRC</td>
<td>135</td>
<td>1</td>
<td>1 (6697)</td>
</tr>
<tr>
<td>BitTorrent</td>
<td>6768</td>
<td>171</td>
<td>118</td>
</tr>
<tr>
<td>WorstPort</td>
<td>2632</td>
<td>205</td>
<td>1 (6523)</td>
</tr>
<tr>
<td>BestPort</td>
<td>2632</td>
<td>205</td>
<td>1 (443)</td>
</tr>
</tbody>
</table>

User model stream activity
TorPS: The Tor Path Simulator

User Model

Network Model

Streams

Client Software Model

Relay statuses

Stream ➔ Circuit mappings
TorPS: The Tor Path Simulator

Network Model

metrics.torproject.org

Hourly consensuses

Monthly server descriptors archive
TorPS: The Tor Path Simulator

Network Model

User Model → Streams → Client Software Model → Stream ↔ Circuit mappings

Relay statuses
TorPS: The Tor Path Simulator

Client Software Model

• Reimplemented path selection in Python
• Based on current Tor stable version (0.2.3.25)
• Major path selection features include
  – Bandwidth weighting
  – Exit policies
  – Guards and guard rotation
  – Hibernation
  – /16 and family conflicts
• Omits effects of network performance
Overview

• Background

• Onion Routing Security Analysis
  o Problem: Traffic correlation
  o Adversary Model
  o Security Metrics
  o Evaluation Methodology
  o Node Adversary Analysis
  o Link Adversary Analysis

• Future Work
### Node Adversary

**100 MiB/s total bandwidth**

<table>
<thead>
<tr>
<th>Relay Type</th>
<th>Number</th>
<th>Bandwidth (GiB/s)</th>
<th>Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>2646</td>
<td>3.10</td>
<td>torservers.net</td>
</tr>
<tr>
<td>Guard only</td>
<td>670</td>
<td>1.25</td>
<td>Chaos Computer Club</td>
</tr>
<tr>
<td>Exit only</td>
<td>403</td>
<td>0.30</td>
<td>DFRI</td>
</tr>
<tr>
<td>Guard &amp; Exit</td>
<td>272</td>
<td>0.98</td>
<td>Paint</td>
</tr>
</tbody>
</table>

**Tor relay capacity, 3/31/13**

**Top Tor families, 3/31/13**
Node Adversary

100 MiB/s total bandwidth

Probability to compromise at least one stream and rate of compromise, 10/12 – 3/13.
Node Adversary

100 MiB/s total bandwidth
83.3 MiB/s guard, 16.7 MiB/s exit
Node Adversary Results

Time to first compromised stream, 10/12 – 3/13

Fraction compromised streams, 10/12 – 3/13
Node Adversary Results

Time to first compromised guard, 10/12 – 3/13

Fraction streams with compromised guard, 10/12 – 3/13
Node Adversary Results

Time to first compromised exit, 10/12 – 3/13

Fraction compromised exits, 10/12 – 3/13
Node Adversary Results

Time to first compromised circuit, 10/12-3/13
Overview

• Background

• Onion Routing Security Analysis
  o Problem: Traffic correlation
  o Adversary Model
  o Security Metrics
  o Evaluation Methodology
  o Node Adversary Analysis
  o Link Adversary Analysis

• Future Work
Link Adversary
1. Autonomous Systems (ASes)
1. Autonomous Systems (ASes)
2. Internet Exchange Points (IXPs)
1. Autonomous Systems (ASes)
2. Internet Exchange Points (IXPs)
3. Adversary has fixed location
1. Autonomous Systems (ASes)
2. Internet Exchange Points (IXPs)
3. Adversary has fixed location
1. Autonomous Systems (ASes)
2. Internet Exchange Points (IXPs)
3. Adversary has fixed location
4. Adversary may control multiple entities
   a. “Top” ASes
   b. IXP organizations
Link Adversary

Client locations
- Top 5 non-Chinese source ASes in Tor (Edman&Syverson 09)

AS/IXP Locations
- Ranked for client location by frequency on entry or exit paths
- Exclude src/dst ASes
- Top k ASes /top IXP organization

<table>
<thead>
<tr>
<th>AS#</th>
<th>Description</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>3320</td>
<td>Deutsche Telekom AG</td>
<td>Germany</td>
</tr>
<tr>
<td>3209</td>
<td>Arcor</td>
<td>Germany</td>
</tr>
<tr>
<td>3269</td>
<td>Telecom Italia</td>
<td>Italy</td>
</tr>
<tr>
<td>13184</td>
<td>HanseNet Telekommunikation</td>
<td>Germany</td>
</tr>
<tr>
<td>6805</td>
<td>Telefonica Deutschland</td>
<td>Germany</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS</td>
<td>3356</td>
<td>Level 3 Communications</td>
</tr>
<tr>
<td>AS</td>
<td>1299</td>
<td>TeliaNet Global</td>
</tr>
<tr>
<td>AS</td>
<td>6939</td>
<td>Hurricane Electric</td>
</tr>
<tr>
<td>IXP</td>
<td>286</td>
<td>DE-CIX Frankfurt</td>
</tr>
<tr>
<td>IXP Org.</td>
<td>DE-CIX</td>
<td>DE-CIX</td>
</tr>
</tbody>
</table>

Example: Adversary locations for BitTorrent client in AS 3320
### IXP organizations ranked by size

<table>
<thead>
<tr>
<th>#</th>
<th>IXP Organization</th>
<th>Size</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Equinix</td>
<td>26</td>
<td>global</td>
</tr>
<tr>
<td>2</td>
<td>PTTMetro</td>
<td>8</td>
<td>Brazil</td>
</tr>
<tr>
<td>3</td>
<td>PIPE</td>
<td>6</td>
<td>Australia</td>
</tr>
<tr>
<td>4</td>
<td>NIXI</td>
<td>6</td>
<td>India</td>
</tr>
<tr>
<td>5</td>
<td>XChangePoint</td>
<td>5</td>
<td>global</td>
</tr>
<tr>
<td>6</td>
<td>MAE/VERIZON</td>
<td>5</td>
<td>global</td>
</tr>
<tr>
<td>7</td>
<td>Netnod</td>
<td>5</td>
<td>Sweden</td>
</tr>
<tr>
<td>8</td>
<td>Any2</td>
<td>4</td>
<td>US</td>
</tr>
<tr>
<td>9</td>
<td>PIX</td>
<td>4</td>
<td>Canada</td>
</tr>
<tr>
<td>10</td>
<td>JPNAP</td>
<td>3</td>
<td>Japan</td>
</tr>
<tr>
<td>11</td>
<td>DE-CIX</td>
<td>2</td>
<td>Germany</td>
</tr>
<tr>
<td>12</td>
<td>AEPROVI</td>
<td>2</td>
<td>Equador</td>
</tr>
<tr>
<td>13</td>
<td>Vietnam</td>
<td>2</td>
<td>Vietnam</td>
</tr>
<tr>
<td>14</td>
<td>NorthWestIX</td>
<td>2</td>
<td>Montana, US</td>
</tr>
<tr>
<td>15</td>
<td>Terremark</td>
<td>2</td>
<td>global</td>
</tr>
<tr>
<td>16</td>
<td>Telx</td>
<td>2</td>
<td>US</td>
</tr>
<tr>
<td>17</td>
<td>NorrNod</td>
<td>2</td>
<td>Sweden</td>
</tr>
<tr>
<td>18</td>
<td>ECIX</td>
<td>2</td>
<td>Germany</td>
</tr>
<tr>
<td>19</td>
<td>JPIX</td>
<td>2</td>
<td>Japan</td>
</tr>
</tbody>
</table>

IXP organizations obtained by manual clustering based on PeerDB and PCH.
Link Adversary

Adversary controls one AS,
Time to first compromised stream, 1/13 – 3/13
“Best”: most secure client AS
“Worst”: least secure client AS

Adversary controls one AS,
Fraction compromised streams, 1/13 – 3/13
“Best”: most secure client AS
“Worst”: least secure client AS
Link Adversary

Adversary controls IXP organization,
Time to first compromised stream,
1/13 – 3/13,
“Best”: most secure client AS
“Worst”: least secure client AS

Adversary controls top ASes,
Time to first compromised stream,
1/13 – 3/13,
Only “best” client AS
Overview

• Background

• Onion Routing Security Analysis
  o Problem: Traffic correlation
  o Adversary Model
  o Security Metrics
  o Evaluation Methodology
  o Node Adversary Analysis
  o Link Adversary Analysis

• Future Work
Future Work

1. Extending analysis
2. Improving guard selection
3. Using trust-based path selection to protect against traffic correlation
4. Dealing with incomplete and inaccurate AS and IXP maps
5. Include Tor’s performance-based path-selection features in TorPS