Computer Security
16. Tor & Anonymous Connectivity

Paul Krzyzanowski
Rutgers University
Spring 2017

Private Browsing

• Browsers offer a "private" browsing modes
  – Apple Private Browsing, Mozilla Private Browsing,
    Google Chrome Incognito Mode, Microsoft InPrivate browsing

• What does it do
  – Do not send stored cookies
  – Do not allow servers to set cookies
  – Do not use or save auto-fill information
  – List of downloaded content
  – At the end of a session
    • Discard cached pages
    • Discard browsing & search history

• Does not protect the user from viruses, phishing, or security attacks

Is private browsing private?

• It doesn't leave too many breadcrumbs on your device
• It limits the ability of an attacker to use cookies
• But
  – Your system may be logging outbound IP addresses
  – Proxies know what you did … so do firewalls & routers
  – Your ISP knows who you are and where you went
  – Web servers get your IP address

Answer: not really

Goal

Communicate while preserving privacy

Why?
  – Avoid consequences (social, political, legal)
  • E.g., political dissidents
  – Avoid geolocation-based services
  – Hide corporate activity (who's talking to whom)
  – Perform private investigations
  – Hide personal info, like searching about diseases you have
  – …

Tor & The Tor Browser

• Tor = The Onion Router
• Tor Browser = preconfigured web browser that uses Tor
  – Provide anonymous browsing
• Hosted on a collection of relays around the world
  – Run by non-profits, universities, individuals
• 100K to millions of users
  – Exact data unknown – it's anonymous
  – Terabytes of data routed each second

History

• Onion routing developed in the mid 1990s at the U.S. Naval Research Laboratory to protect U.S. intelligence communications
• Additional work by the Defense Advanced Research Projects Agency (DARPA)
• Patented by the U.S. Navy in 1998
  – Naval Research Laboratory released to code for Tor under a free license
• The Tor Project
  – Founded in 2006 as a non-profit organization with support of the EFF
What is anonymity?

- **Unobservability**
  - Inability of an observer to leak participants to actions
- **Unlinkability**
  - Inability to associate an observer with a profile of actions
  - *E.g.*, Alice posts a blog under an assumed name
  - Unlinkability = inability to link Alice to a specific profile

Relay

Alice \(\rightarrow\) store.com

Encrypt traffic between Alice & relay

We can use encrypted connections (TLS) to hide network traffic

What if someone eavesdrops on the relay?

Multiple relays

You cannot see all activity at one relay

Correlation Attack

- If an eavesdropper watches entry & exit of data
  - She can correlate timing & size of data at the 1st relay with outputs of the last relays
  - If Alice sends a 2 KB request to Relay 1 at 19:12:15 and Relay 2 sends a 2 KB request to store-3.com at 19:12:16 and store-3.com sends a 150 KB response to Relay 3 at 19:12:17 and Alice receives a 150 KB response at 19:12:18
  - ... we're pretty sure Alice is talking to store-3.com
**Correlation Attack**

- You can make a correlation attack difficult
  - Pad or fragment messages to be the same size
  - Queue up multiple messages, shuffle them, and transmit them at once
- This works in theory but is a pain in practice
  - Extra latency, traffic
  - You still need a LOT of users to ensure anonymity
- Relays should be hosted by third parties to get many different groups as input
  - E.g., a relay within fbi.gov tells you all input comes from fbi.gov

**Circuits**

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted</td>
<td>Relay1</td>
<td>Relay2</td>
<td>Relay3</td>
<td>store-1.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Relay1</td>
<td>Relay2</td>
<td>Relay3</td>
<td>store-2.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Relay1</td>
<td>Relay2</td>
<td>Relay3</td>
<td>store-3.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Relay1</td>
<td>Relay2</td>
<td>Relay3</td>
<td>store-4.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Relay1</td>
<td>Relay2</td>
<td>Relay3</td>
<td>store-5.com</td>
</tr>
</tbody>
</table>

- Alice extends the relay to Relay2
  - Alice sends a message to Relay2
    - First part: "On circuit 123, send Relay Extend to Relay2, the message is encrypted with S2, Relay2 establishes a TLS link to Relay2, if it didn't have one"
    - Second part of the message from Alice: initial handshake with Relay2, encrypted with Relay2's public key
      - Relay2 picks a random circuit for identifying this data stream to Relay2, e.g., 456
      - Circuit 123 on Relay2 connects to Circuit 456 on Relay2
  - Alice picks a circuit ID (e.g., 123) and asks Relay1 to create the circuit

**Setting up a circuit (1)**

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted</td>
<td>Relay1</td>
<td>Relay1</td>
<td>Relay2</td>
<td>store-1.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Relay1</td>
<td>Relay1</td>
<td>Relay2</td>
<td>store-2.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Relay1</td>
<td>Relay1</td>
<td>Relay2</td>
<td>store-3.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Relay1</td>
<td>Relay1</td>
<td>Relay2</td>
<td>store-4.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Relay1</td>
<td>Relay1</td>
<td>Relay2</td>
<td>store-5.com</td>
</tr>
</tbody>
</table>

- Alice connects to Relay1
  - Sets up a TLS link to Relay1
  - Does a one-way authenticated key exchange with Relay1, agree on a symmetric key, S1
  - Alice picks a circuit ID (e.g., 123) and asks Relay1 to create the circuit

**Setting up a circuit (2)**

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>Relay1</td>
<td>Relay1</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>Relay1</td>
<td>Relay1</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>Relay1</td>
<td>Relay1</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>Relay1</td>
<td>Relay1</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>Relay1</td>
<td>Relay1</td>
</tr>
</tbody>
</table>

- Alice extends the relay to Relay2
  - Alice sends a message to Relay2
    - First part: "On circuit 123, send Relay Extend to Relay2, the message is encrypted with S2
      - Relay2 establishes a TLS link to Relay2, if it didn't have one"
    - Second part of the message from Alice: initial handshake with Relay2, encrypted with Relay2's public key
      - Relay2 picks a random circuit for identifying this data stream to Relay2, e.g., 456
      - Circuit 123 on Relay2 connects to Circuit 456 on Relay2
  - Alice picks a circuit ID (e.g., 123) and asks Relay1 to create the circuit

**Setting up a circuit (3)**

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>store-1.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>store-2.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>store-3.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>store-4.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>store-5.com</td>
</tr>
</tbody>
</table>

- Alice extends the relay to Relay3
  - Same process - Alice sends a Relay Extend message to Relay3
  - Messages to Relay3 are encrypted with S2, then with S3
  - Relay3 decrypts the message to identify its circuit (123)
  - Routes message to Relay3, on circuit 456
    - Circuit 123 is connected to circuit 456

**Sending a message (5)**

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>store-1.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>store-2.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>store-3.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>store-4.com</td>
</tr>
<tr>
<td>Encrypted</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>store-5.com</td>
</tr>
</tbody>
</table>

- Alice sends a message to store-3.com
  - Each router strips off a layer of encryption
  - At the end:
    - Directive to S3 to open a TCP connection to store-3.com
    - Send messages
    - Get responses
Not a VPN

- Neither IP nor TCP packets are transmitted – just data streams
  - Too easy to identify the type of system by looking at TCP formats and responses
- Just take contents of TCP streams and relay the data
- End-to-end TLS works fine
  - TLS sits on top of TCP ... it's just data going back and forth

Finding nodes

- Ideally, everyone would use some of the same nodes
  - Otherwise traffic would be distinguishable
- Multiple trusted parties supply node lists
  - Merge lists together
    - Union: if popularity-based, danger of someone flooding a list of nodes to capture traffic
    - Intersection: someone can block out nodes
  - Multiple parties vote on which nodes are running and behaving well
    - Distributed consensus
- Clients get
  - List of nodes and their public keys

Is it anonymous?

- Not really
- You can do a correlation attack
  - ISPs know who's talking to whom
  - May need to access traces from multiple ISPs
  - Can be really difficult if nodes have a lot of traffic (and it's similarly dense)
- Compromised exit node
  - Exit node decrypts the final layer and contacts the service

I2P and Garlic Routing

I2P = Invisible Internet Project

- Tor uses "onion routing"
  - Each message from the source is encrypted with one layer for each relay
- Garlic routing
  - Combines multiple messages at a relay
  - All messages, each with its own delivery instructions going to one relay are bundled together
  - Makes traffic analysis more difficult
- Tor circuits are bidirectional: responses take the same path
- I2P "tunnels" are unidirectional
  - One for outbound and one for inbound: the client builds both
  - Sender gets acknowledgement of successful message delivery

Services on top of I2P

- I2PTunnel: TCP connectivity
- Chat via IRC (Internet Relay Chat)
- File sharing
  - BitTorrent
  - iMule (anonymous file sharing)
  - I2PHex: Gnutella over I2P
- I2P-Bote: decentralized, anonymized email
  - Messages signed by the sender's private key
  - Anonymity via I2P and variable-rate delays
  - Destinations are I2P-Bote addresses
- I2P-Messenger, I2P-Talk
- Syndie: Content publishing (blogs, forums)

- Tor: far more users → more anonymity
  - Focused on anonymous access to services
- I2P: focuses on anonymous hosting of services
  - Uses a distributed hash table (DHT) for locating information on servers and routing
  - Server addressing
    - Uses cryptographic ID to identify routers and end services
The end