Anonymity on the Internet

- Often considered bad
  - Only criminals need to hide

- Identification not possible in all cases
  - Real-world identification is usually too easy to subvert

- Even attribution may be faulty
  - E.g., malware on your system – part of botnet launching a DDoS attack

Some services retain information about you

- Accounts, configuration settings
- Cloud storage
  - Files, email, photos, blogs, web sites
  - Encryption so the server has no access not always possible
- Interests, browsing history, messages
  - Important for data mining & targeted advertising
  - E.g., Facebook, Google

Cookies on the web

- Local name=value data stored at the browser & sent to a server
  - Avoids having to log in to a service repeatedly
  - Keeps track of session, shopping cart, preferences
- Associated with the site (same-origin policy)
  - Facebook cookies don't get sent to google ... and vice versa
- Tracking cookies (third-party cookies)
  - Websites can embed resources from another site (e.g., bugme.com)
    - Via an ad in an iFrame or a 1x1 pixel image
  - bugme.com's cookies will be sent to bugme.com
  - HTTP message contains a Referer header, which identifies the encompassing page
  - Lots of different sites may use bugme.com's services
    - Bugme.com can now build a list of which sites the visitor has visited
- Most browsers have policies to block third-party cookies

Private Browsing

- Browsers offer a "private" browsing modes
  - Apple Private Browsing, Mozilla Private Browsing,
    - Google Chrome Incognito Mode, Microsoft InPrivate browsing
- What do these modes 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
  - Web servers get your IP address
  - Proxies know what you did ... so do firewalls & routers
  - Your ISP knows who you are and where you went
  - DNS servers know what addresses you're looking up
- Some store and use this data

Answer: not really

Encrypted sessions?

Great ... eavesdroppers can't see the plaintext
But they can see where it's coming from and where it's going

Anonymous communication: Goal

Communicate while preserving privacy

Why?
- Avoid consequences (social, political, legal)
  - E.g., political dissidents, whistleblowers, crime reporting
- 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 & Anonymous Connectivity

Tor & The Tor Browser

- Tor = The Onion Router
- Tor Browser = preconfigured web browser that uses Tor
  - Provides anonymous browsing
- Hosted on a collection of relays around the world
  - Run by non-profits, universities, individuals
  - Currently over 6,000
- 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 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 → Relay → store.com

Encrypt traffic between Alice & relay

---

**Relay with multiple parties**

Relay

A1 → store-1.com
A2 → store-2.com
A3 → store-3.com
A4 → store-4.com
A5 → store-5.com

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

What if someone eavesdrops on the relay?

---

**Multiple relays**

A1 → Relay1 → store-1.com
A2 → Relay2 → store-2.com
A3 → Relay3 → store-3.com
A4 → Relay4 → store-4.com
A5 → Relay5 → store-5.com

Alice is doing something

You cannot see all activity inside 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
  - *E.g.*, Alice sends a 2 KB request to Relay, at 19:12:15
  - Relay sends a 2 KB request to store-3.com at 19:12:16
  - store-3.com sends a 150 KB response to Relay, at 19:12:17
  - Alice receives a 150 KB response at 19:12:18
    - *we're pretty sure* Alice is talking to store-3.com
Setting up a circuit (1)

- Alice connects to Relay1
  - Sets up a TLS link to Relay1
  - Does a one-way authenticated key exchange with Relay1, agrees 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)

- Alice extends the relay to Relay2
  - Alice sends a message to Relay1
    - First part: on circuit 123, send Relay Extend to Relay1, the message is encrypted with S1
  - Relay2 establishes a TLS link to Relay3 (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 Relay3, e.g., 456
  - Circuit 123 on Relay2 connects to Circuit 456 on Relay3
  - Does a one-way authenticated key exchange with Relay3, agrees on a symmetric key, S2
  - All traffic flows through Relay3, and is encrypted with S2

Setting up a circuit (3)

- Alice extends the relay to Relay3
  - Same process – Alice sends a Relay Extend message to Relay2
  - Messages to Relay3 are encrypted with S2 and then with S3, E_S2( E_S3(M) )
  - Relay3 decrypts the message to identify its circuit (123)
  - Routes message to Relay4 on circuit 456
  - Circuit 123 is connected to circuit 456

Sending a message (5)

- 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
    - Sends messages
    - Gets responses

Correlation Attack

- You can make a correlation attack 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
Not a VPN
• Neither IP nor TCP packets are transmitted in the message
  – Just data streams
  – It would be 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 logs 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
  – iBit (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 (currently) → 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 endpoint services
The end