Computer Security

17. Anonymous Connectivity & Tor

Paul Krzyzanowski
Rutgers University
Spring 2019

Anonymous Connectivity

Anonymous communication

Communicate while preserving privacy

Often considered bad: "only criminals need to hide"

- Drugs
- Hit men
- Stolen identities
- Counterfeit $
- Stolen credit cards
- Guns, hacking
- Bitcoin laundering
- Fraud
- Porn

Anonymous communication

Communicate while preserving privacy

But there are legitimate uses

- Avoid consequences (social, political, legal)
  - Accessing content in oppressive governments
  - 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
  - searching for information about diseases you have, loans, credit problems

Anonymity on the Internet

Even without anonymity:

- 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
- Your 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
    - They can also correlate with past traffic
  - 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*

Improvements to Chrome's Incognito Mode

Detecting Incognito mode allows websites to block users if they cannot be tracked

- Services had a simple trick to determine whether a user is using Incognito Mode
  - Use FileSystem API – Chrome-specific method that gives a website a sandboxed file system for its own use
  - API is completely disabled in Incognito mode
- Near-term plan (early 2019)
  - Google will create a virtual file system in RAM
  - Will be erased when the user leaves Incognito Mode

Other browsers have similar problems

- Firefox, IE/Edge
  - `IndexedDB` is not available
  - Attempts to access it causes it to throw an `InvalidStateError`
- Safari
  - Disables its `localStorage` API in Private Browsing
  - An attempt to call the `setItem` method throws an exception
- Older versions of IE10/Edge
  - `IndexedDB` doesn't even exist in privacy mode
- Other techniques exist too
  - Services can send code to check for private browsing modes and block users if they cannot be tracked

Encrypted sessions?

Great … eavesdroppers can't see the plaintext

But they can see where it's coming from and where it's going

The service knows your IP address & can track you

Surface Web
Deep Web
Dark Web

The different types of web

- Surface Web
  - **Web content that can be indexed by mainstream search engines**
  - Search engines use web crawlers
    - Go through a list of addresses from past crawls
    - Access pages provided as sitemaps by website owners
    - Traverse links on pages being crawled to find new content
- Deep Web
  - **Web content that a search engine cannot find**
  - Unindexed content, often from dynamically-generated pages
  - E.g., query results from libraries, govt and corporate databases

Dark Web

Part of the Deep Web that has been intentionally hidden

- Not accessible through standard browsers
  - Need special software, such as a Tor browser
- Servers do not register names with DNS
  - Sometimes use a .onion pseudo-top-level domain
- Still uses
  - HTML web pages
  - HTTP & FTP for moving content
- Illicit & legitimate services
  - Drugs, stolen identities, counterfeit $, etc.
  - Blackbook (similar to Facebook), recipes, books
  - Anonymous news access: https://www.nytimes3xbfgragh.onion/

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

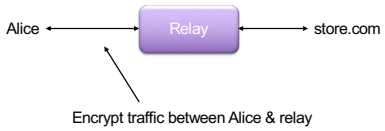
- **Onion routing** developed in the mid 1990s at the U.S. Naval Research Laboratory to protect U.S. intelligence communications
- Additional work by the Defense Advanced Research Projects Agency (DARPA)
- Patented by the U.S. Navy in 1998
  - Naval Research Laboratory released to code for Tor under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

What is anonymity?

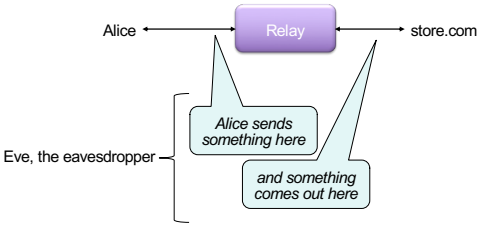
- **Unobservability**
  - Inability of an observer to leak participants to actions
- **Unlinkability**
  - Inability to associate an observer with a profile of actions
  - *E.g., Alice posts a blog under an assumed name*

  Unlinkability *= inability to link Alice to a specific profile*

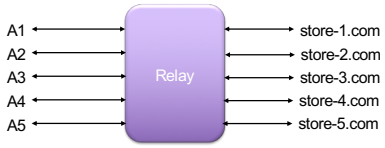
Relay



Relay



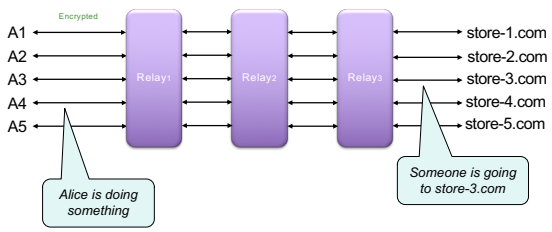
Relay with multiple parties



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

What if someone eavesdrops on the relay?

Multiple relays



Correlation Attack

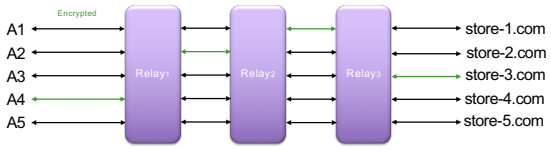
If an eavesdropper watches entry & exit of data

- 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 $Relay_3$ 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

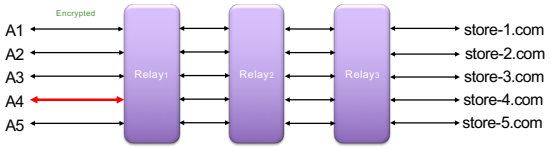
- You can make a **correlation attack** attack difficult
  - Pad or fragment messages to be the same size
  - Queue up multiple messages, shuffle them, and transmit them at once
- This works in theory but is a pain in practice
  - Extra latency, traffic
  - You still need *A LOT* of users to ensure anonymity
- 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



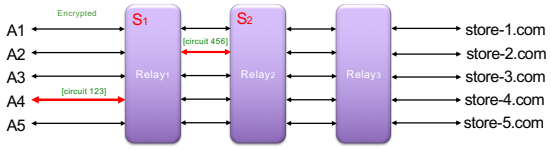
- Alice selects a list of relays through which her message will flow
- This path is called a circuit
- No node knows if the previous node is the originator or relay
  - Only the final node (**exit node**) knows it is the last node

Setting up a circuit (1)



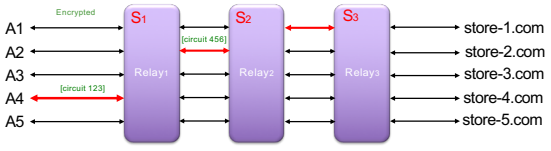
- Alice connects to Relay1
  - Sets up a TLS link to Relay1
  - Does a one-way authenticated key exchange with **Relay1** – agree on a symmetric key, **S1**
  - Alice picks a circuit ID (e.g., 123) and asks Relay1 to create the circuit

Setting up a circuit (2)



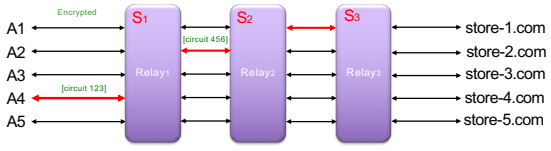
- Alice extends the relay to $Relay_2$
  - Alice sends a message to $Relay_1$:
    First part = "on circuit 123, send ***Relay Extend*** to $Relay_2$ – the message is encrypted with S_1
  - $Relay_1$ establishes a TLS link to $Relay_2$ (if it didn't have one)
  - **Second part** of the message from Alice: **initial handshake with $Relay_2$, encrypted with $Relay_2$'s public key**
  - $Relay_2$ picks a random circuit for identifying this data stream to $Relay_2$, e.g., 456
    - Circuit 123 on $Relay_1$ connects to Circuit 456 on $Relay_2$
  - Does a one-way authenticated key exchange with **$Relay_2$** – agree on a symmetric key, **S_2**
    - All traffic flows through **$Relay_1$** and is encrypted with **S_1**

Setting up a circuit (3)

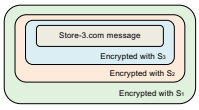


- Alice extends the relay to $Relay_3$
  - Same process – Alice sends a ***Relay Extend*** message to $Relay_2$
  - Messages to $Relay_2$ are encrypted with S_2 and then with S_1
    $E_{S1}(E_{S2}(M))$
  - $Relay_1$ decrypts the message to identify its circuit (123)
  - Routes message to $Relay_2$ on 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 S_3 to open a TCP connection to store-3.com
  - Send messages
  - Get responses



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

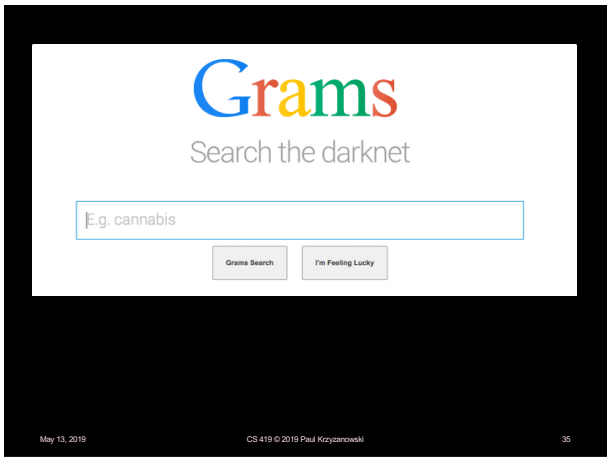
- Ideally, everyone would use some of the same nodes
  - Otherwise traffic would be distinguishable
- Multiple trusted parties supply node lists
  - Merge lists together
    - Union: if popularity-based, danger of someone flooding a list of nodes to capture traffic
    - Intersection: someone can block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

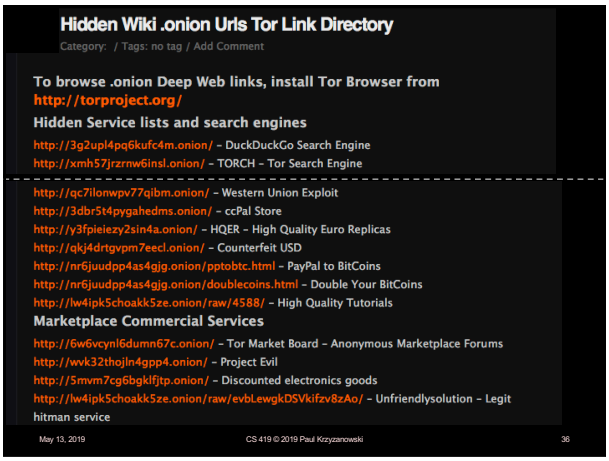
Is it anonymous?

- Not really
- You may be able to 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

Some problems

- Searching is difficult
  - Search engines, such as **Grams**, often give bad results
  - **Hidden Wiki** (http://thehiddenwiki.org) – Directory of Tor .onion sites
    - Often full of bad links
- Users are the weakest link
  - Sites constantly changing addresses to avoid DDoS attacks
  - Lots of scammers
  - Honeypots set up by law enforcement
  - Many ISPs block access to Tor
- Sites can get found & shut down
  - Silk Road 2.0: shut down by the FBI & Europol on Nov 6 2014
  - Silk Road 3.0: went offline due to loss of funds in 2017
  - AlphaBay (largest source of contraband): shut down in July 2017
  - Hansa Market (competitor to AlphaBay): also shut down in 2017 by Dutch police





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)

Status

- **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