Bitcoin over Tor isn't a good idea

Alex Biryukov and Ivan Pustogarov
Université du Luxembourg
Ivan.pustogarov@uni.lu

May 18, 2015
San Jose
Bitcoin anonymity

- They can link your transactions with your IP address:
  - NSA, ISP
  - Malicious peers
  - Remote low-resource attacker

[CCS 2014]
Agenda

- Background/Motivation
- Isolating Bitcoin clients
- User fingerprinting
- Low-resource Sybil attacks
- Bitcoin is a decentralized payments system
- Bitcoin address = Hash(Pub_key)
Bitcoin over Tor

- The recommended (by Bitcoin developers) way to avoid IP leakage is to **use Tor**
- Generate **many** Bitcoin pseudonyms
Part 1: Isolation
Idea

• We exploit Bitcoin built-in Anti-DoS protection
  – A Bitcoin peer keeps track of each IP that sent him a message
  – If the message is “bad”, the peer increases the penalty score. Score 100 = Ban

• Let's see what happens when Tor is used to connect to the Bitcoin network
Eve sends a bad message

Eve sends a malformed transaction (TX)

R1 → R2 → R3 → R4 → R5 → Malformed TX

Eve

Malformed TX

R1

R2

R3

R4

R5

Malformed TX

Bitcoin nodes
Eve sends a bad message
Now bob cannot connect
Strategy

- We **ban** all "good" Tor exit relays and all "good" Bitcoin peers
- We **do not ban** our "bad" exits and peers
- Bob will try different Tor Exits and different Bitcoin peers until she chooses the attacker's ones.
Ban result

Bad relay/peer

Good relay/peer

Client 1

Client 2
Attack vectors

- Delaying/dropping blocks and transactions
  - Increases chances for double-spending
- Deanonymization using traffic confirmation
- Revealing client's Guard node
- Linking different pseudonyms
Part 2: Fingerprinting
Idea

- Bitcoin uses the following peer discovery mechanism:
  - Clients send GETADDR messages to known peers to get a list of IP addresses of other peers.
  - Clients receive back ADDR messages with lists of IP addresses.

- Rule 1 (simplified): When I receive ADDR message with IP's, I try to store all of them.

- Rule 2 (simplified): When I get GETADDR message, I am ready to tell all IP's I know.
Idea

- Bitcoin uses the following peer discovery mechanism:
  - Client sends a GETADDR message to known peers to get a list of IP addresses of other peers.
  - Client receives back ADDR messages with a lists of ~2500 IP address
- Rule 1 (simplified): *When I receive ADDR message with IP's, I try to store all of them*
- Rule 2 (simplified): *When I get GETADDR message, I am ready to tell all IP's I know*
Fingerprinting

- We can put any IP addresses combination at any client connected to us (Tor, vpn, etc.)
- We can later request this combination
Set “address” fingerprint
Request fingerprint
(Apparent) Problem

- Client's address database can store 20,480 address
- When a client switches on, it connects to 8 peers and typically requests 2,500 new IP address = 20,000 non-unique addresses
- Looks like our fingerprint will be overwritten within several new sessions
Fingerprint is stable

- When clients receive an ADDR message (simplified):
  - Discard already known addresses
  - **Note**: Address has a timestamp, the older it is the “worse” it is.
  - If IP is not in the database, take 4 random addresses from the database and overwrite the oldest one (addresses have timestamps).

- We queried Bitcoin peers and got 4,941,815 IP's
  - Only 6% were unique
  - For 45% of addresses the timestamp was older than 10 hours
Fingerprint is stable

<table>
<thead>
<tr>
<th>Session number</th>
<th>Time since start, hours</th>
<th>Remaining addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>1.5</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td>2.5</td>
<td>100</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>98</td>
</tr>
<tr>
<td>8</td>
<td>3.5</td>
<td>92</td>
</tr>
<tr>
<td>9</td>
<td>5.5</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>36</td>
</tr>
</tbody>
</table>

**TABLE II. ADDRESS COOKIE DECAY RATE (EXAMPLE)**
Fingerprint is stable

- Survives restarts
- Is preserved for many hours (two sessions):
  - 76% after 10 hours
  - 55% after 24 hours
- Going to be fixed.
Part 2: Updating fingerprints (sibyl attacks)
Motivation

- Clients should periodically connect to our peers so that we update their fingerprints
- We want to be the first to receive their transactions when Tor is not used (to deanonymize them)
- Problem: one peer = 1 IP address
Idea 1 (old)

(known to the Bitcoin community for quite some time):

• A Bitcoin peer by default accepts only 117 connections
• A Bitcoin peer does not check if they come from the same IP address
• Strategy: fill all connection slots of “good” peers
Idea 2 (new)

- Peer addresses are of the following form (IP,PORT)
- If client receives (IP₀, PORT₁) and there is already an entry in the client’s database (IP₀, PORT₀), (IP₀, PORT₁) will be discarded
- Strategy: flood with clients with addresses of legitimate Bitcoin servers but wrong port numbers.
Conclusion

- We force all Bitcoin-over-Tor clients to connect through our “bad” peers/relays
- We set a fingerprint when see a suspicious transaction (or check if it was already set)
- When a client connects to our peer without Tor we check the fingerprint and deanonymize his Tor transactions
- We also can increase our chances to be chosen by client without Tor
Thank you.