Introduction

- Overview of Tor
  - What is Tor? Why use Tor?
- How Tor works
  - Encryption, Circuit Building, Directory Server
- Drawback of Tor’s directory server
- Potential solution
  - Using DNS Security Extension
What is Tor

- A distributed overlay network based on voluntarily run relays around the world
- Provides low latency anonymity to TCP-based applications
- Protects users from being identified online
  - Journalists, activists, business people
- Circumvents Censorship
Tor Network: the Basic

- **Directory nodes**
  - Servers set up by Tor project
  - List all the nodes available in Tor network

- **Relay nodes**
  - Servers run by volunteers around the world

- **Onion proxy**
  - Proxy running on client computer

- **Circuit**
  - An encrypted virtual tunnel
  - Made of a chain of Tor relay nodes
  - Traffic routed through multiple relays from the user to the final destination
Tor begins building circuits as soon as it has enough information to do so.

Client (onion proxy) downloads the info about all the ORs from directory server. Includes keys, ips, ports, bandwidth, etc.

OP opens a new connection, attaches to a circuit
Diff-Hellman Key Exchange

Both agree on a prime number $p=23$ and base $g=5$.

Alice chooses a secret integer $x=6$.

Bob chooses a secret integer $y=15$.

$g^x \mod p = 5^6 \mod 23 = 8$

$g^y \mod p = 5^{15} \mod 23 = 19$

$\text{key} = B^x \mod p = 19^6 \mod 23 = 2$

$\text{key} = A^y \mod p = 8^{15} \mod 23 = 2$

“Two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel”

Example from wikipedia
How a Tor Circuit is built

1. **OP** sends a create cell to the 1st node in the path it chooses, containing the first half of the Diffie-Hellman handshake ($g^x_1$) encrypted with the onion key of the OR1.

2. OR1 responds with a created cell containing $g^y_1$, H(K1).

3. OR1 copies $E(g^x_1)$ into a create cell and sends it to OR2.

4. OR2 returns the create cell to OR1.

5. OR1 wraps $g^y_2$ and $K_2$ and sends them back to OP.

6. The circuit between OP and OR1 is established.

- OR1 includes $E(g^x_2)$ in the created cell sent to OR2.
- OR2 copies $E(g^x_2)$ and sends it to OR1.
- OR1 decrypts $E(g^x_2)$ and sends it to OP.
- OR1 decrypts $g^y_2$ and $K_2$ and sends them back to OP.
Tor’s Message

How Tor Fetches a Website

Know originator
Does not know the destination

OP iteratively encrypts the cell with the symmetric key of each hop up to that OR

TCP Handshake

Knows originator
Does not know destination

OP
Each onion router periodically signs and sends its keys, bandwidth, port, etc., to the Tor directory servers.

Clients (onion proxies) download the consensus file from a directory server.

Each directory server periodically signs and sends its individual view of the Tor network to other directory servers.

DS: Directory Server
OR: Onion Retours
OP: Onion Proxy
A Problem with Tor Directory Servers

- Tor requires each directory server and client user (onion proxy) to know all of the relay nodes in the Tor network.

- Clients periodically ask directory servers:
  - Who is there in the Tor network?
  - What is their status and info?
    - Is a relay node active? Public key, port, IP, etc.

- What if the directory server is inaccessible
  - E.g., Blocked by ISP?

- Potential solution: Ask DNS for directory server information
NO authentication and integrity
DNS Security Extension Resolution

Where is www.example.com

Root DNS server
- IP of .com DNS
- Pub Key of .com DNS
- All signed by root key

.example.com DNS server
- IP of example DNS
- Pub Key of example DNS
- All signed by .com key

Example.com DNS server

Signed(example.com, key_example.com)

Signed(example.com)

Signed(.com, key_com)

Signed

xxx.xxx.xxx.xxx

DNS resolver

Where is www.example.com

xxx.xxx.xxx.xxx
network-status-version 3
vote-status consensus
consensus-method 13
valid-after 2013-04-25 19:35:00
fresh-until 2013-04-25 19:40:00
valid-until 2013-04-25 19:50:00
voting-delay 20 20
client-versions
server-versions

A document format version.
Vote status
Consensus methods that are using
Start time of the consensus
Time to produce next consensus
The time this consensus expires
Consensus File (Partial)

- @downloaded-at 2013-04-28 06:52:04
- router relay8 128.220.221.150 9000 0 9500
- onion-key
- signing-key
- router-signature
- Signature from directory servers
Replacing Directory Server with DNSSEC

Still not fully resilient
Adversary can simply block tordir.org

Where is torDir.org

Root DNS server

DNS resolver

Signed(tordir.org, key_example.com)

Signed Info about tor relay nodes

Signed(tordir.org, key_example.com)

.orgDNS server

Info about tor relay nodes

Signed Info about tor relay nodes

TorDir.org

DNS server
Replacing Directory Server with DNSSEC

- Many random domain names
  - Change regularly
  - Generate by hash function
- Each domain name is only responsible for a subset of all available Tor relay nodes.
  - When querying one domain, a client is only provided with a subset of relay nodes
- Info about relays is encrypted using domain name’s keys
  - Domain name key changes regularly
Conclusion

- Difficult to block all domain name
  - Thousands of domain name
  - Each responsible for subset of relays
  - As long as one domain name is not blocked
- Difficult to block all IP address of relay nodes
  - Directory info is encrypted
  - Encrypted key regular change
References

- https://www.torproject.org/
- Main Tor Specification, https://gitweb.torproject.org/torspec.git?a=blob_plain;hb=HEAD;f=tor-spec.txt