Anonymous Network Concepts & Implementation

kevinkoo001@gmail.com
Overview

1. Overview & Background

2. Anonymous Network

   tor

   freenet

   Gnunet

   I2P

3. Circumvention Techniques against Censorships

   Obfsproxy

   flashproxy
Overview

- **Attack Trends Summary**
  - Modern attack mostly involves malware, which
    - Attempts to conceal attack itself
    - Makes it hard to trace themselves down from network perspective
    - Makes it difficult to find artifacts by wiping out themselves from system perspective
    - Employs many techniques to be hard for analysis including: Anti-VM, Anti-disassembly, Anti-debugging and cryptography
    - Infects a target but do nothing harm until they achieve their goals
  - Imagine how future malware will evolve, which
    - Employs the combination of existing – even legitimate – tools/techniques in a malicious fashion
    -Emerges new variables targeting cloud computing
    -Focuses highly on target-oriented attack which does not affect others
    -Uses steganography technique in a wild more often
    -Forms private tor network with exploited zombie machines
Overview

- **Malware/Crimeware**

  - Let’s briefly take a look at what to have, how to spread and what to do.

<table>
<thead>
<tr>
<th>Features</th>
<th>Distribution</th>
<th>Misuse</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Key Loggers</td>
<td>✓ Attachment</td>
<td>✓ Information Compromise</td>
</tr>
<tr>
<td>✓ Screenscrapers</td>
<td>✓ Peer-to-Peer Networks</td>
<td>✓ Spam Transmission</td>
</tr>
<tr>
<td>✓ Email, IM Redirectors</td>
<td>✓ Piggybacking</td>
<td>✓ Denial-of-Service, DDoS</td>
</tr>
<tr>
<td>✓ Session Hijackers</td>
<td>✓ Internet Worms, Virus</td>
<td>✓ Click Fraud</td>
</tr>
<tr>
<td>✓ Web Trojans</td>
<td>✓ Web Browser Exploits</td>
<td>✓ Data Ransoming</td>
</tr>
<tr>
<td>✓ Transaction Generators</td>
<td>✓ Server Compromise</td>
<td>✓ Identity Stealing</td>
</tr>
<tr>
<td>✓ Data Theft</td>
<td>✓ Affiliate Marketing</td>
<td>✓ Credit Card Abuse</td>
</tr>
<tr>
<td>✓ Man-in-the-Middle</td>
<td>✓ Phishing</td>
<td>✓ Defamation</td>
</tr>
<tr>
<td>✓ Rootkits</td>
<td>✓ Pharming</td>
<td>✓ Embezzlement</td>
</tr>
</tbody>
</table>

- **Features**
- Distribution
- Misuse
Background

- Necessity / Motivation (1/2)

  “Anonymity serves different interests for different user groups.”
  by Roger Dingledine, the creator of the Tor

![Diagram showing the different interests served by anonymity for different user groups.](attachment:diagram.png)
Background

- Necessity / Motivation (2/2)
  - Regular citizens do not want to be watched and tracked.
  - Businesses need to keep trade secrets.
  - Law enforcement needs anonymity to get the job done.
  - Government need anonymity for their security.
  - Journalists and activists need anonymity for their personal safety.
  - Hard to configure your own network though!!

But

- Compromised botnets
- Stolen mobile phones
- Open wireless nets
- Malware spread (trojans, virus, worms)
- Spamming
- Phishing
Implemented Anonymous Network

(1) tor
(2) freenet
(3) Gnunet
(4) I2P
The Tor (the Onion Routing) at a glance

- When there is an evil user or server, then it could be blocked with ease.
- Tor is designed for hiding where the communication comes from, and going to.
The Tor (the Onion Routing) at a glance

- Tor network consists of many relay and exit nodes.

- The idea comes from **Chaum’s Mix-Net design** at first. 
  Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms 
  (Communications of the ACM February 1981 Volume 24 Number 2)
Implemented Anonymous Network - Tor

- The Tor (the Onion Routing) Concept
  - Open source software
    https://svn.torproject.org/cgi-bin/viewvc.cgi/Tor/
    http://sourceforge.net/projects/advtor/
  - A distributed, anonymous Network
  - A Protocol
  - Tor provides online anonymity
 Implemented Anonymous Network - Tor

- The Tor (the Onion Routing): How it works
  - Alice’s Tor client obtains a list of Tor nodes from a directory server, Dave.
  - Alice’s Tor client picks a random destination server.
  - Alice’s Tor client selects a second random path.

[Diagram showing how Tor works]

https://www.torproject.org/about/overview.html.en
The Tor (the Onion Routing): Diffie-Hellman Key Exchange Algorithm in TLS (1/2)

- DH establishes a shared secret that can be used for secret communications while exchanging data over a public network.
- (Step A) Alice and Bob have common information and secrets which belong to one’s own.
- (Step B) Each creates a value with a secret, and transmit it to the other.
- (Step C) Using a value by the other, each creates common secret.

\[
\begin{align*}
A &= g^a \mod p \\
B &= g^b \mod p \\
K &= B^a \mod p \\
K &= A^b \mod p \\
K &= (g^a \mod p)^b \mod p = (g^b \mod p)^a \mod p = B^a \mod p
\end{align*}
\]
The Tor (the Onion Routing): Diffie-Hellman Key Exchange Algorithm in TLS (2/2)

SSL / TLS (Secure Socket Layer / Transport Layer Security)

The Tor (the Onion Routing): Entire Mechanism (1/3)

- Circuit (Chain) establishment

1. Provide Node List
2. Choose nodes (chain, circuit)

- All communication between nodes over TLSv1
- The originator and directory node actually involve with DH over TLS.
The Tor (the Onion Routing): Entire Mechanism (2/3)

- Sending HTTP data over the Internet anonymously

- C1 = \{RELAY C1: [RELAY (Send HTTP request to Web-Server-IP)]\}
- C2 = \{RELAY C2: ENCRYPTED CONTENT\}
- C3 = \{Send HTTP request to Web-Server-IP\}
- OR1 (Entry Node) knows the origin which the packets come from.
- OR2 (Exit Node) knows the destination which the incoming packets go to.
- If any, all OR nodes between entry node and exit node only know the adjacent nodes.
The Tor (the Onion Routing): Entire Mechanism (3/3)

- Diagram about tor operation in details

https://svn.torproject.org/svn/projects/design-paper/tor-design.html
Implemented Anonymous Network - Tor

- The Tor (the Onion Routing): Statistics as of July, 2013 (1/2)
  - One of the largest deployed network
  - Almost 4,000 relays
  - Almost 2,000 bridges
  - Around 500,000 users per each day

Directly connecting users from all countries

The Tor Project - https://metrics.torproject.org/
http://tigerpa.ws/tor_metrics/
The Tor (the Onion Routing): Statistics as of July, 2013 (2/2)

- Some countries (ISPs) have a censorship to prevent users from getting access to certain sites.
- Bridge Relays (almost 25,000)
  Helps censored users access the Tor network
  Are not listed in the same public directories

Top-10 countries by possible censorship events (BETA):

<table>
<thead>
<tr>
<th>Country</th>
<th>Downturns</th>
<th>Upturns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iran</td>
<td>31</td>
<td>14</td>
</tr>
<tr>
<td>Syrian Arab Republic</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>China</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>United Republic of Tanzania</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>India</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Vietnam</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Republic of Korea</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Gibraltar</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Taiwan</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Dominica</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Implement Anonymous Network - Tor

- The Tor (the Onion Routing): Official Record
  - Tor Relay IP Address in the Past
    - [https://metrics.torproject.org/exonerator.html](https://metrics.torproject.org/exonerator.html)

**Was there a Tor relay running on this IP address?**

<table>
<thead>
<tr>
<th>IP address in question:</th>
<th>(Ex.: 86.59.21.38 or 2001:858:2:2:aabb:0:563b:1526)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date or timestamp, in UTC:</td>
<td>(Ex.: 2010-01-01 or 2010-01-01 12:00)</td>
</tr>
</tbody>
</table>

- Tor Relay IP Search
  - [https://metrics.torproject.org/relay-search.html](https://metrics.torproject.org/relay-search.html)

**Tor Metrics Portal: Relay Search**

Search for a relay in the relay descriptor archive by typing (part of) a nickname, $-prefixed fingerprint, or IP address and optionally a month (yyyymm) or up to three days (yyyymm-dd) in the following search field and clicking Search. The search will stop after 30 hits or, unless you provide a month or a day, after parsing the last 30 days of relay lists.
The Tor (the Onion Routing): Detection Technique Example (1/2)

- Someone should talk to directory server (public) to join the tor network.
- A series of unauthorized certificates in SSL/TLS communication before encryption.
- Other than IP/Port (Layer 3), all TLS traffic are encrypted.
The Tor (the Onion Routing): Detection Technique Example (2/2)

- X.509 Certificate has an issuer/subject field.
- Tor initiates a series of SSL/TLS connections with 3~5 hosts at the same time. → This requires behavior-based analysis if bridges are used for censorship bypass.

http://helpforsure.wordpress.com/tag/x-509-version-3-digital-certificates/
Implemented Anonymous Network - Freenet

- **What is Freenet?**
  - A separate network that runs over the internet
  - Only access Freenet content through Freenet including:
    - Freesites (websites on Freenet),
    - in-Freenet chat forums (FMS, Sone, etc),
    - files shared within Freenet,
    - in-Freenet email
  - Distributed Database
  - The more popular a file or page, the more widely it will be cached, the faster it will download.

https://freenetproject.org/faq.html
Properties

- A large distributed storage device
- When storing a file, you receive a key to retrieve the file.
- With a key, *Freenet* returns the appropriate file.

Data Management

Location to store data: `C:\Users\[UserID]\AppData\Local\Freenet\datastore`

Little or no control over what is stored in your datastore

Kept or deleted depending on how popular they are.

Routing

Initially, each node has no information about the performance of the other nodes. (Random Routing)

More documents → same node; begin to cluster with data items (Cuz the same routing rules are used)

As a result, the network will self-organize into a distributed, clustered structure.
## Implemented Anonymous Network - Freenet

### Properties

- **Keys**
  - Each file that exists on Freenet has a key associated with it.
  - Fproxy → `http://localhost:8888/[Freenet Key]`

- **CHK - Content Hash Keys**
  - The decryption key is stored encrypted within the file.
  - **CHK** @ file hash , decryption key , crypto settings

- **SSK - Signed Subspace Keys**
  - Usually for sites that are going to change over time
  - **SSK** @ public key hash , decryption key , crypto settings / user selected name - version

- **USK - Updateable Subspace Keys**
  - Linking to the latest version of a Signed Subspace Key (SSK) site
  - **USK** @ public key hash , decryption key , crypto settings / user selected name - version

- **KSK - Keyword Signed Keys**
  - Allowing to save named pages in Freenet
  - **KSK** @ myfile.txt
Implementing Anonymous Network - *Gnunet*

- **What is *Gnunet***?

  - Started in late 2001
  - Implemented for secure peer-to-peer networking
  - Improved content encoding: **ECRS, the encoding for censorship resistant sharing**

  - A framework for secure peer-to-peer networking that does not use any centralized
  - Focus on anonymous censorship-resistant file-sharing
  - Provides anonymity by
    - making messages originating from a peer indistinguishable from messages that the peer is routing
    - acting as routers and use link-encrypted connections with stable bandwidth utilization
  - Similar to tor, but limited to share files anonymously, searching, swarming, and caching.

https://gnunet.org/
What is *I2P?* (1/2)

- Began in 2003
- An anonymizing network, a low latency mix network
- Goal: producing a low latency, fully distributed, autonomous, scalable, anonymous, resilient, and secure network
- All data is wrapped with several layers of encryption. (End-to-End)
- The network is both distributed and dynamic, with no trusted parties and no centralized resources.

![Diagram of I2P](http://www.i2p2.de/)
Implemented Anonymous Network – *I2P*

- What is *I2P?* (2/2)
  - Made up of a set of nodes ("routers") with a number of unidirectional inbound and outbound virtual paths ("tunnels")
  - Has its own internal network database (using a modification of the *Kademlia* algorithm) for distributing routing and contact information securely

Garlic Wrapped:
- (Garlic Routing)
  1. Layered Encryption
  2. Bundling multiple messages together
  3. ElGamal/AES Encryption

Garlic Wrapped:
- (Garlic Routing)
  1. Layered Encryption
  2. Bundling multiple messages together
  3. ElGamal/AES Encryption

http://www.i2p2.de/how_intro
Circumvention Techniques against Censorships

(1) DPI (Deep Packet Inspection)

(2) Obfsproxy

(3) Flashproxy
Circumvention Techniques against Censorships

- How to circumvent censorships by DPI (deep packet inspections)
  - Even bridges could be blocked by DPI.
  - New techniques have been introduced to circumvent such censorships.
  - Core technology: **pluggable transport transformation**

  ✓ **Obfsproxy** is a Python framework for implementing new pluggable transports. It uses Twisted for its networking needs, and *pyptlib* for some pluggable transport-related features. It supports the obfs2 and obfs3 pluggable transports. (by George Kadianakis)

  ✓ **Flashproxy** turns ordinary web browsers into bridges using websockets, and has a little python stub to hook Tor clients to the websocket connection. (by David Fifield)

  ✓ **ScrambleSuit** is a pluggable transport that protects against follow-up probing attacks and is also capable of changing its network fingerprint (packet length distribution, inter-arrival times, etc.). It's part of the Obfsproxy framework. (by Philipp Winter)

  ✓ **StegoTorus** is an Obfsproxy fork that extends it to a) split Tor streams across multiple connections to avoid packet size signatures, and b) embed the traffic flows in traces that look like html, javascript, or pdf. (by Zack Weinberg)

  ✓ **SkypeMorph** transforms Tor traffic flows so they look like Skype Video. (by Ian Goldberg)

  ✓ **Dust** aims to provide a packet-based (rather than connection-based) DPI-resistant protocol. (by Brandon Wiley)

  [https://www.torproject.org/docs/pluggable-transports.html.en](https://www.torproject.org/docs/pluggable-transports.html.en)
Circumvention Techniques against Censorships

- How to circumvent censorships by DPI - **Obfsproxy**
  - Transforms the Tor traffic between the client and the bridge.
  - Supports multiple protocols, **pluggable transports**.
  - Get bridges in Bridge DB and then add them to tor

https://www.torproject.org/projects/obfsproxy.html.en
https://bridges.torproject.org/?transport=obfs2
Circumvention Techniques against Censorships

- How to circumvent censorships by DPI - **flashproxy**

  - Began as a project in Stanford's class in spring 2011
  - Works at tor version 0.2.3.2-alpha or later
  - This model have supposed that facilitator outside have been already blocked.
    : Client does not communicate directly to facilitator, designed to be covert and very hard to block.

  1. Makes TLS Connection
  2. Sends encrypted email from anonymous address to a special facilitator addr.
  3. Facilitator responds with address
  4. Flash proxy (web browser)
  5. Proxy connects to relay

https://crypto.stanford.edu/flashproxy/