Tor — Der Zwiebel Router: Ein TCP anonymisierendes Netzwerk

Matthias Bauer
bauerm@weggla.franken.de

20. November 2005

Übersicht

Anonymität?

Anonymität
Gibts im Netz nicht so recht

- IP Adressen
- TelekommunikationsÜberwachungsVerordnung, CyberCrimeConvention
- Logs aller Arten
- Verschlüsseln (PGP, STARTTLS, ssh, https) versteckt nur den Inhalt

Anonymität: Wer brauchts?

Wer braucht Anonymität?

- Aktivisten: Indymedia, Regimekritiker, Dissidenten
- Journalisten (Cicero . . .)
- Beratungsstellen
- Strafverfolger: Insidertips
- Unternehmen: Anfragen bei der Konkurrenz
- Unternehmen: Haftung für Handlungen von Mitarbeitern

Anonymität: Wer brauchts?

Wer brauchts?
Eigentlich jeder.
Will man wirklich völlig Unbekannten mitteilen:

- Wem man E-mails schreibt?
- Welche Web-Pages man besucht?
- Was man übers Netz ein/verkauft?
- ...

Kontrollverlust über persönliche Daten ermöglicht "Identitätsdiebstahl"

Anonymität: Nebenwirkungen?

Mögliche Nebenwirkungen

- Schützt das nicht auch [*Lieblingsfeindbild hier einsetzen*]?
- Die brauchen das nicht!
  - Verbrecher kaufen falsche Identitäten
  - Cracker übernehmen fremde Rechner
  - Ehrliche Bürger nicht!
- Besser den vielen Ehrlichen einfacher machen

Anonymität braucht Gesellschaft

Anonymität gibts nur in Gesellschaft

- Anonymität = Ununterscheidbarkeit in einer Gruppe
- → Gruppe bilden
- Daten für andere transportieren, also kanns jeder gewesen sein

Zum Beispiel E–Mail:

- "Umschlag im Umschlag . . ."
- Seit den frühen 90ern: Remailer (`mixmaster`, `mixminion`)
- Nicht für zeitnahe Kommunikation
- Nur in eine Richtung

Onion Routing: Prinzip

Zeitnahe, anonymisierte zwei-Wege Kommunikation

- "Strohhalm in Strohhalm . . ."
- Von Paul Syverson vom Naval Research Laboratory
- Init: Vorhandene Knoten und deren Pubkeys nachschaun
- Init: Verbindung (Circuit) durch mehrere Knoten aufbauen (PubKey Crypto, langsam)
- Betrieb: Knoten reichen Daten weiter (Sym Crypto, schnell)
- Der End–Knoten redet mit Maschinen im Netz
- Daten werden gleichen Pfad zurück durchgereicht
- Auf dem Weg zum Ziel wird eine Schicht Verschlüsselung entfernt
- Auf dem Weg zurück eine Schicht hinzugefügt
- $\longrightarrow$ Zwiebel Routing

Onion Routing: Implementation und Infrastruktur

`tor` — Die Implementation

- Seit 2001: "The Onion Router" = TOR
- von Matej Pfajfar, Roger Dingledine und Nick Mathewson
- Gefördert von der Electronic Frontier Foundation und der US Marine
- Läuft auf praktisch allem (einschließlich Windows)

`tor` — Das Netz

- Im Daemon mode vermittelt `tor` für andere TCP/SOCKS
- Diese Knoten bilden das Tor Netz
- Momentan $>$ 300 Knoten
- Mindestens einer im KNF, mindestens sieben in Franken

Onion Routing für Anwender

Auch nutzbar, wenn man keinen Knoten betreibt

- Client: Nutzt das Tor Netz
- Bieten SOCKS4A und SOCKS5 Service für Socks-fähige Programme
- SOCKS ist ein schon lang genutztes Protokoll für gezieltes Passieren von Filtern (gabs schon vor NAT)
  - In vielen populären Web Browsern
  - Mit kleineren Verrenkungen für meisten TCP-basierten Protokolle adaptierbar
- "Socksifizierer" wie NEC Socksify, Dante, `tsocks`, `dsocks` überladen die Syscalls (`bind`, `connect`, `...`), die typischerweise für TCP Verbindungen genutzt werden

Onion Routing: Clients, DNS Tricksereien

Problem mit den meisten SOCKS Implementationen: DNS

- DNS Anfragen laufen oft am Socksifizierer vorbei und die verraten, mit wem mal als nächstes reden will
- SOCKS4A/5 tunnelt die Anfragen, aber viele Programme nutzen das nicht
- Also auch die Name-Lookup Funktionen umbiegen
- Einfach bei HTTP: Proxy macht Lookup z.B. `privoxy`
- Für BSDs: Dug Songs `dsocks` hat ein `dsocks-torify`, macht alles automatisch, überlädt auch `gethostbyname`
- Geht immer: einen TCP-nach-SOCKS4A Konverter starten

```
socat TCP4-LISTEN:1234 \
SOCKS4A:localhost:weggla.franken.de:22,socksport=9050&
ssh -v -p 1234 localhost
```

Onion Routing: Server

Knoten sind einfach zu betreiben

- Name ausdenken, Key erzeugen und an die Autoren schicken
- Braucht keine `root` Privilegien
- Gut geschriebener Code
- Betreiber bestimmt, welche Dienste aussen genutzt werden können
- Kapazität kann per Konfig beschränkt werden
- Tor kontrolliert Kapazität selber, kein Getrickse mit Firewall Regeln nötig
- Kann auch Maxima pro Zeitraum beschränken, z.B. nur 1GB pro Woche
  Sobald das GB voll ist, legt `tor` sich schlafen bis zur nächsten Woche

Cooles Feature: Hidden Services

- Idee: Wenn `tor` sowieso alle DNS Anfragen auflösen muss, kann das Netz auch `tor`-spezifische Top Level Domains haben
- TLDs: `.onion` und `.exit`
- Mit `.exit` kann ein Client angeben, wer der letzte Knoten auf dem Pfad sein soll. Beispiel
  `www.franken.de.schnarpf2nd.exit`
- In `.onion` ist der Name eines Servers der SHA1 Hash seines Public Keys
- Der Server meldet seinen Namen an und öffnet ein oder mehrere Pfade ins `tor` Netz ("Rendezvous Points")
- Der Client öffnet ebenfalls einen Pfad und bittet den letzten Knoten, ihn mit einem der Rendezvous Points zu verbinden
- Client authentifiziert den Server mit dem Hash des Pubkeys (= Name)

Hidden Services Anwendungen

Ergebnis: IP des Servers ist geheim, IP des Clients ist geheim, trotzdem können sie miteinander reden.

- Anonymes Publizieren
- Es gibt ein hidden Wiki für Tor Doku
- OFTC bietet anonymes IRC
- Als `dyndns` Ersatz
- Erlaubt Redundanz (mehrer Eingänge mit gleichem Namen)
- Um Dienste auf Rechnern anzubieten, die keine (routebare) IP haben
- Config für SSH: `Host hidden_homehost`

```
  HostName 127.0.0.1
  ProxyCommand socat -
SOCKS4A:localhost:vcg6nibhbturwioz.onion:22,socksport=9050
```

Wie Anonym?

Wie anonym ist man mit `tor`?

- Nicht vor allwissendem Beobachter (timing)
- Nicht wenn man persönliche Daten im Klartext rausschickt
- Nicht bei gezielter Beobachtung beider Endpunkten der Kommunikation
- Möglicherweise noch andere Situationen

Holt mit wenig Aufwand viel raus, bessere Anonymität wird deutlich teurer

Zusammenfassung

http://tor.eff.org/

- Tor ermöglicht anonyme TCP Sessions unter vernünftigen Annahmen
- Viele Knoten, mehr wären aber besser
- Serve aufsetzen ist einfach
- "Hidden Services" bieten
  - Versteckte Dienste
  - Adressierung durch Public Keys
  - Tunneling durch Filter
- Bisher beste Infrastruktur für Fast-Echtzeit Anonymität