

## Breaking TOR Anonymity

The TOR network provides anonymity, has wide support and enjoys great popularity. TOR is often used for malicious activities such as network attacks or SPAM and therefore we had a look into how to break the anonymity.

A technical analysis of the onion router (also known as TOR), the relayed traffic and possibilities to reveal user identities

### Introduction

The goal of TOR is to provide anonymity but also security in the Internet. The first ideas for the TOR project go back to the year 2000. Matej Pfajfar started to work on it in 2002 at Cambridge University. Until 2004, the project was supported by the Office of Naval Research and Defense Advanced Research Projects Agency (DARPA).

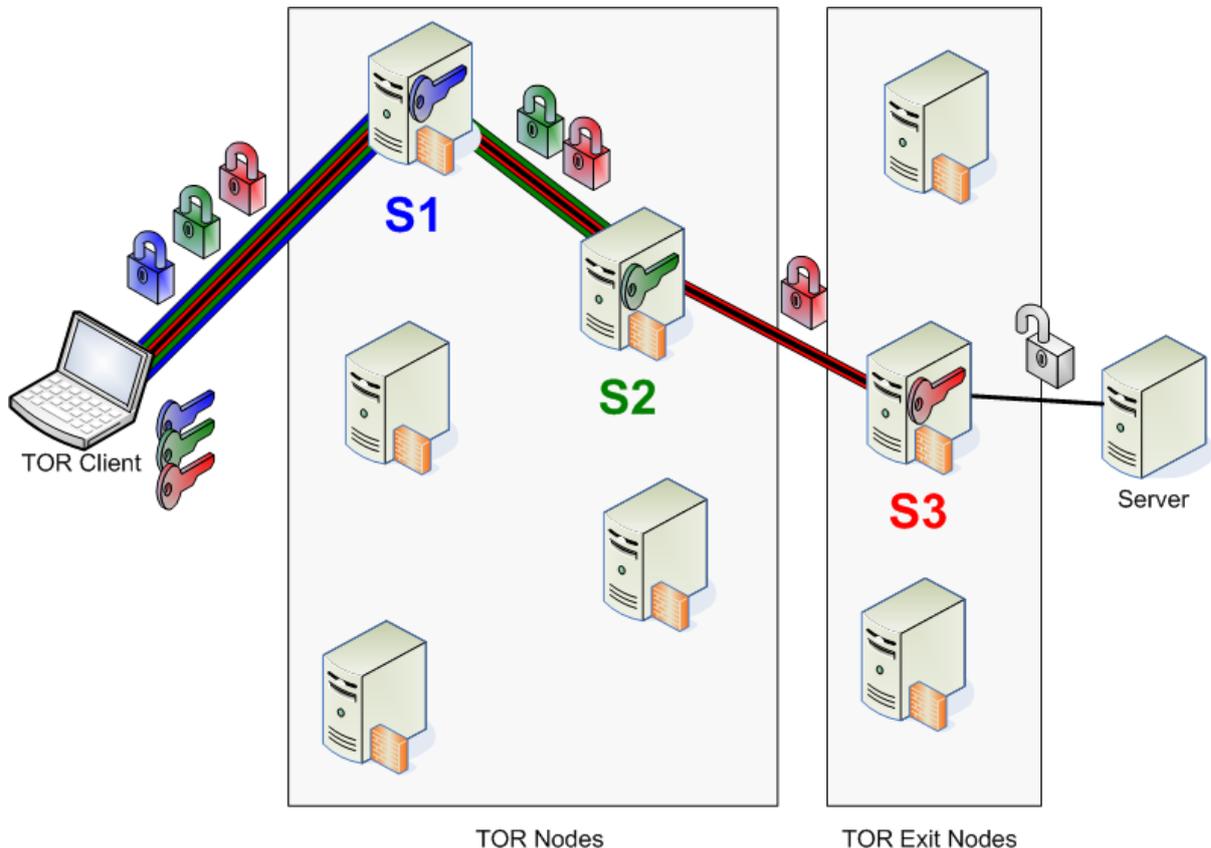
TOR is used by governments, businesses as well as private citizens who want to stay anonymous for various reasons.

According to Roger Dingledine [2] at 24C3 in

December 2007, there are currently more than 200'000 TOR users and the TOR network pushes over 1Gbit/s over roughly 2500 TOR nodes. Only about one third of the nodes are exit nodes, which are the systems that finally connect to the server the TOR user requested.

### Onion Routing

So how does it work? From a user's point of view, the TOR software that is installed on the client is just a proxy. When the software is started, the program fetches a list of all TOR servers from the Internet. The list contains the capabilities, IP address and more about each server. Based on this list, the TOR software generates a path how the next connection should be routed through the TOR network. The data packets from the client are then encrypted with the key of the last node (S3), which is also called exit node, afterwards encrypted with the key of the middle node (S2) and finally encrypted with the key of the first node (S1). When the data packet is sent, it is decrypted once on each TOR server and forwarded to the next hop until it reaches the exit node (S3) which sends the decrypted packet to the destination server. Packets in the other



direction are encrypted and decrypted the opposite way. TOR does not provide end-to-end encryption. Traffic from the exit node to the destination server is not encrypted by TOR. It "only" provides anonymity, nothing else. The exit node is able to view all the original traffic bits and bytes.

Usually TOR is used together with Vidalia [4] and Privoxy [5].

## TOR Exit Nodes & Statistics

To gain knowledge about the relayed traffic and the servers talked to, we set up exit nodes running for a while and dumping all the traffic.

As a TOR node, it is optional to act as an exit node. The exit node is the last hop in each TOR connection. On these special nodes, each packet from the TOR client is decrypted and sent to the destination server. To the destination server and other parties (who may eavesdrop on the network) it looks like the exit node were the source of the connection.

Suddenly, it looks as if a provider of a TOR exit node were downloading child porn, sending junk mails, uploading illegal archives containing music or movies or attacking various services.

Our traffic analysis shows that a big part of the sniffed data contains illegal or unethical activities including SPAM, brute force attacks on logins, porn and more.

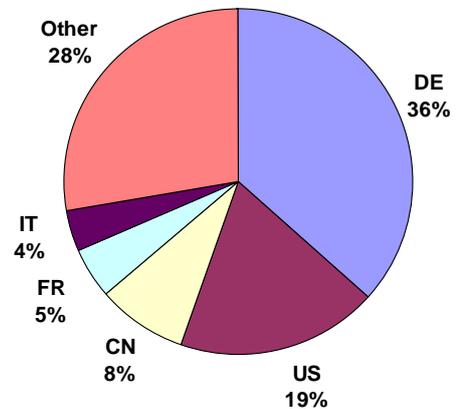
Since TOR does not provide end-to-end encryption, it is possible to sniff unencrypted protocols such as HTTP, POP or SMTP on TOR exit nodes. Obviously many users are still not aware of the fact that TOR does not make a connection more secure, but "only" provides a certain level of anonymity. We were also able to find a few passwords but too few to real logins to complain about that. Most login attempts look like someone is bruteforcing accounts.

As an exit node it is also possible to inject arbitrary content. The client is not able to check the integrity of protocols like HTTP and therefore cannot detect an injected part of a website.

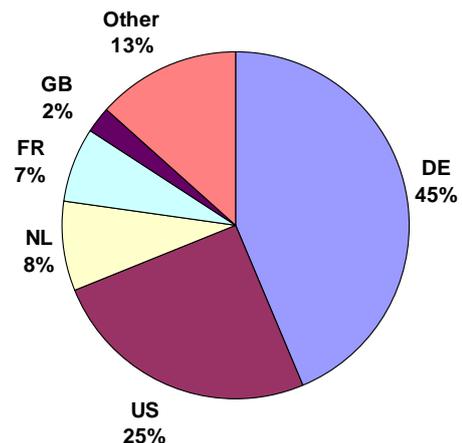
From the traffic gathered, we are now able to roughly tell what protocols are relayed, for what purpose it is used for and where the traffic goes to.

The following charts are excerpts from what we have gained.

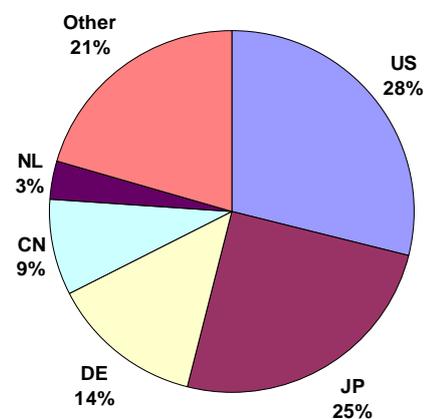
TOR nodes:



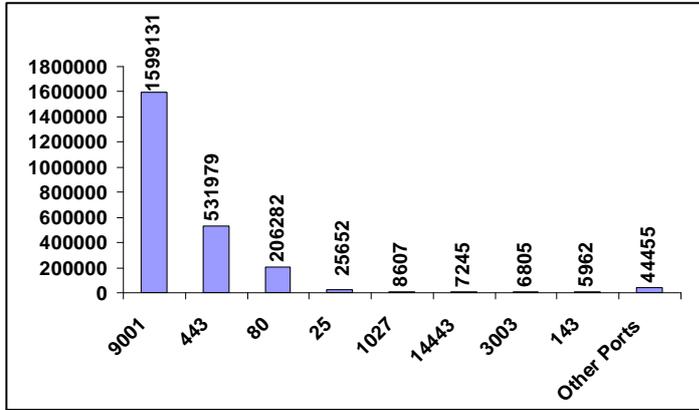
Targeted servers (by volume):



Targeted servers (unique IPs):



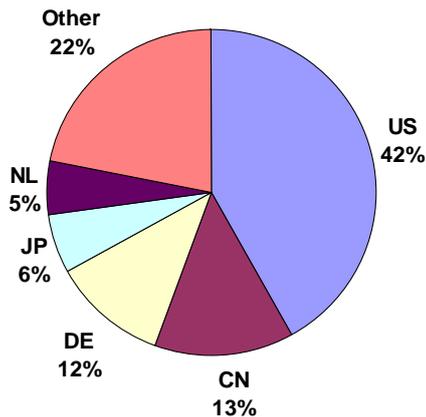
Destination ports:  
(filtered connections made from port 9001)



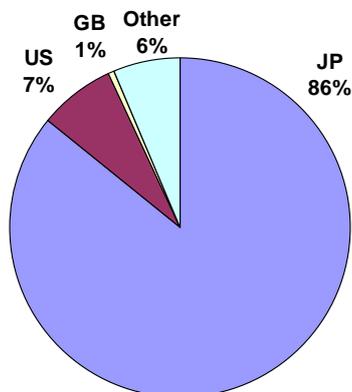
approximately 50 hours, we gathered 22 GB of data (average speed 128kB/s).

A quick look at the User-Agent tag in HTTP traffic (destination port 80) revealed, that the variety amongst the operating systems and browsers is quite big. We could see everything from Windows (vast majority is XP but also some Vista and even NT 4) to Linux (Ubuntu and SUSE) and Mac (PPC and Intel). The big majority of browsers that showed up were Firefox (mainly 2.0.0.x and even quite a few old ones). Some requests were made with Internet Explorer between version 5 and 7.

Targeted web servers (port 80):



Targeted mail servers (port 25):



After booting the TOR server, it took roughly an hour, before the server was heavily used. The available bandwidth in upload direction (100kB/s) was used completely. Within

## Breaking TOR Anonymity

As stated before, the number one goal of TOR is to provide anonymity. Are there any ways to get the real client IP address anyway?

By monitoring big amounts of the traffic of TOR, namely on the first and the last node, and by using statistical analysis, it is possible to determine the source of a connection. This scenario is realistic for big ISPs or Internet backbones.

However, we focused on another aspect. Is it possible to determine the identity of a user on a web server or on an exit node (since the biggest amount of traffic was unencrypted HTTP)?

The answer is clearly yes, it is easily possible to discover the real identity of a TOR client. The results, however, heavily depend on the client configuration. All tests were conducted on these two systems:

Client Setup:

- Windows XP Pro SP2
  - Firefox 2.0.0.11
  - Internet Explorer 7
  - JRE 1.6 (default settings)
- Mac OS X 10.5.1
  - Firefox 2.0.0.11
  - Safari 3.0.4
  - JRE 1.5 (default settings)
  - Flip4Mac WMV

Additionally it was made sure that Quicktime, RealPlayer and Shockwave are installed. TOR was installed together with Vidalia and Privoxy

(on Windows only), leaving the default settings on both machines.

We conducted various tests with Java Applets, Windows Media, JavaScript, Shockwave, Real Media and Quicktime Media on Windows XP and Mac OS X 10.5.1 with two browsers each.

The idea behind this kind of "attack" is to find commonly installed browser plugins which are not configured properly in order to reveal the client IP, either through HTTP or even with raw sockets (TCP or UDP).

## Conclusion

From our analysis we can say that all tested plugins can be used to disclose the real IP address of the user. However, all plugins also offer to set a proxy to be used as well. The only technology we could not get to work was JavaScript.

### Java

It was possible to break TOR's anonymity by using RAW TCP and UDP sockets. This approach only worked on OS X though. The JRE default policy on Windows XP did not allow to use sockets.

### Shockwave

As one of the most often installed plugins in browsers, Shockwave has a huge functionality including opening sockets. The following code was used to successfully connect back to the server, bypassing the TOR proxy (for all browsers tested):

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application
xmlns:mx=http://www.adobe.com/2006/
mx:xml xmlns="" layout="absolute"
minWidth="20" minHeight="20"
creationComplete="startService()"
pageTitle="flexToRAP">
<mx:Script><![CDATA[
import flash.net.XMLSocket;
private var socket:XMLSocket;
private function
startService():void {
socket = new XMLSocket();
socket.connect("server", 2000); }
]]> </mx:Script></mx:Application>
```

### Windows Media

Embedding a Windows Media file on a website activates the Windows Media plugin (Flip4Mac on OS X). The following code was used (testmedia.wmv):

```
<ASX version="3.0"><entry>
<ref HREF="mms://server/test.wmv"/>
</entry></ASX>
```

The plugin tried to access the resource through the proxy and since this failed (resource was not available), the plugin tried to access it without proxy and thus disclosed the real IP for all four browsers tested.

### Real Media

The Real Media plugin behaved similarly to the Windows Media plugin and disclosed the identity of all tested browsers. The following URL was put in a real media file - realvideo.rm:

```
rtsp://server:554/realvideo.rm?cloa
kport=8080,554
```

### Quicktime

Last but not least we also tested the behavior of the Quicktime plugin. By embedding a MOV object in the website, Safari on OS X and Internet Explorer on Windows disclosed the client identity.

## Revealing More Client Identity

There are other methods to disclose client identities we did not experiment with. We just want to give some ideas here.

### DNS Leak [7]

As explained above, TOR can be used as a SOCKS proxy and can be used for any TCP connection. There are differences in how requests are sent to the SOCKS proxy for different SOCKS versions. With SOCKS 4 and in practice mostly also with SOCKS 5 requests, the target host is given with its IP address and not by its FQDN. This implies that the browser makes his own DNS requests to resolve the FQDN the user entered to an IP address which can then be sent to the SOCKS proxy (TOR). It should easily be possible to correlate the DNS requests with the HTTP requests made on a website. As an exit node something like the following could be injected for every HTTP response passing the exit node:

```
<img src= "http://server/img.jpg">
```

However, this has not been tested by us. This approach will not work when using SOCKS 4 (which uses hostnames) and might not be working when using SOCKS 5.

### *Mail Traps*

What we tested was directed against web traffic. What about emails? As an entry node (if unencrypted) or mail server we are able to inject new parts of mails if we want. We could plant an image tag in an existing real mail or add an additional mail to the inbox.

However, this was not tested and depends on the mail client (proxy settings, viewing method, text vs. html mails, etc.). So this needs more investigation.

### *Cookies*

A rather obvious way to reveal an identity is to set a persistent cookie on the client browser and thus being able to track the user over various requests and sessions, no matter if he is using TOR or not. However, if the client uses the Privoxy tool, it is unlikely that these cookies are still returned to the server.

The TOR project website [1, 7] clearly states that there are ways of revealing the real identity of a client and they give support to configure a client to be less likely to disclose his identity. How many users really follow those ideas would be interesting but tricky to figure out.

## Getting Rid of TOR Connections

Given that you do not want to have TOR clients accessing your infrastructure, the easiest way to block TOR clients is to block TOR exit nodes. The following script gets a list of TOR nodes (not only exit nodes though) from the server and saves the IP addresses in a list "torips.txt".

```
#!/bin/bash
wget -q
http://tor.noreply.org/tor/status/all
grep -E "^r" all | awk '{print $7}'
| sort | uniq > torips.txt
rm -f all
```

There are services providing up-to-date blacklists in "mod\_rewrite"-style to block TOR

clients from accessing your Apache web server [3]. This approach has a small percentage of false-positives, because all clients coming from an IP address with a TOR node behind are rated as TOR clients even if they are not using TOR. The amount can be limited by only blocking TOR exit nodes (stated in the server list).

A few steps that can be taken to prevent clients from using TOR in your network (one or more approaches may be used):

1. Firewall does not allow incoming connections.
2. Firewall does not allow direct outgoing connections (through proxy only).
3. Block downloads of the TOR server list URLs (on proxy).
4. Block connections going to IPs on the TOR server list (certainly has other implications).
5. Block connections with ports that are listed on the TOR server list (usually 9001 and 9030).
6. Monitor TOR server lists to check if one of your public IPs is on the list.

## Finally...

TOR has a big community and is widely used. The provided anonymity heavily depends on the system configuration and the installed plugins on the client computer. However, we consider the chance to reveal the client identity for HTTP traffic as quite real at least if more than just one approach is used.

On the infrastructure side, requests coming from the TOR network can be blocked automatically by using the provided (periodically updated) IP address list.

## About the Author



Martin Suess completed his studies for BS in Computer Sciences in December 2004 with his diploma thesis in the fields of ZigBee networks in Singapore. During his studies he concentrated on IT security, as well as on network and Internet technologies. After his study, he worked on Bluetooth- and embedded-projects at the University for Applied Sciences in Rapperswil. At the same time he coached student projects as well as tutorials in "Algorithms and Datastructures in Java". He joined Compass Security AG as a full time security analyst in January 2006.

martin.suess@csnc.ch  
<http://www.csnc.ch/>

## About Compass Security

The Job of a security specialist is like searching in the fog. The more opaque the environment, the harder it is to find traces and establish methods and tactics. A good compass can help to determine the direction and choose a path that will securely lead to the destination.

Compass Security Network Computing AG is an incorporated company based in Rapperswil (Lake of Zurich) Switzerland that specializes in security assessments and forensic investigations. We carry out penetration tests and security reviews for our clients, enabling them to assess the security of their IT systems against hacking attacks, as well as advising on suitable measures to improve their defenses.

Compass Security has considerable experience in national and international projects. Close collaboration with the technical universities of Lucerne and Rapperswil enable Compass to carry out applied research so that our security specialists are always up-to-date.

## Credits

Thanks to Axel for his support with some nifty shell scripting and reviewing!

## References

- [1] The TOR Project  
<http://www.torproject.org/>
- [2] TOR Presentation at 24C3 2007  
<http://events.ccc.de/congress/2007/Fahrplan/events/2325.en.html>
- [3] TOR Blacklist  
[http://proxy.org/tor\\_blacklist.txt](http://proxy.org/tor_blacklist.txt)  
<http://torstatus.blutmagie.de/>
- [4] Privoxy – Privacy Proxy  
<http://www.privoxy.org/>
- [5] Vidalia – Controller GUI for TOR  
<http://vidalia-project.net/>
- [6] Bruce Schneier on TOR  
[http://www.wired.com/politics/security/commentary/securitymatters/2007/09/security\\_matters\\_0920](http://www.wired.com/politics/security/commentary/securitymatters/2007/09/security_matters_0920)
- [7] TOR and DNS Leak with SOCKS  
<http://wiki.noreply.org/noreply/TheOnionRouter/TorFAQ>