Tor

194.144 Privacy-Enhancing Technologies

Dr. Martin Schmiedecker

$whoami

Dr. Martin Schmiedecker:

- former SBA Research
- currently working for Bosch Engineering
- certified expert witness for 68.60 and 68.62
- member and co-founder of the Foundation for Applied Privacy

$whoami

Foundation for
Applied Privacy

- association based in Vienna
- hosting multiple Tor relays
- 7% exit traffic, globally
- open DoH/DoT resolvers
- https://applied-privacy.net

Outline

Tor

- What is Tor
- Tor in Numbers
- Onion Services

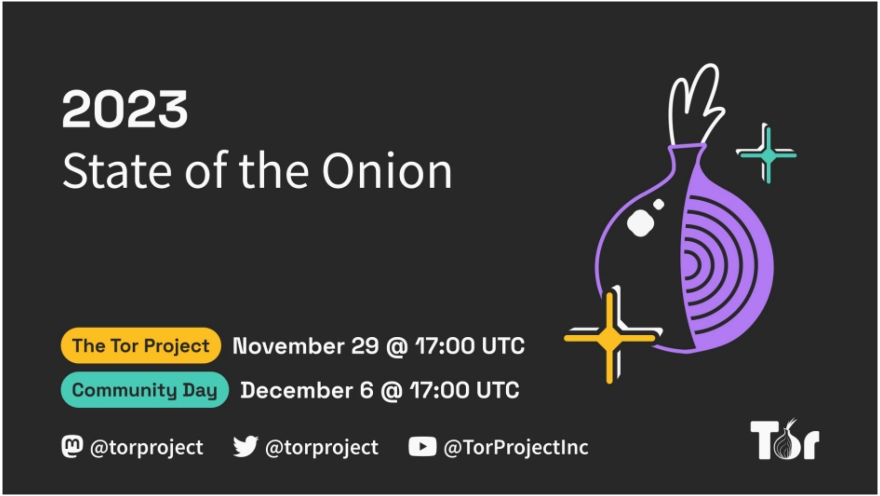
Attacking Tor

Using Tor

Tor

Tor

Self-Study on Tor



2023

State of the Onion

The Tor Project November 29 @ 17:00 UTC

Community Day December 6 @ 17:00 UTC

@torproject @torproject @TorProjectInc

Tor

Tor:

- not TOR, no longer “Tor Onion Router”
- worldwide overlay anonymity network
- hides IP adresses
- approx. 5? million users, every day
- de-facto standard for online anonymity

original paper:

- “Tor: The Second-Generation Onion Router”, Usenix Security 2004

Tor

Target audience:

- By now “everyone who uses the internet”
- … and who doesn’t like to be tracked

Often used by:

- journalists
- law enforcement
- academic research
- IT security
- CAPTCHA enthusiasts :-)

Tor

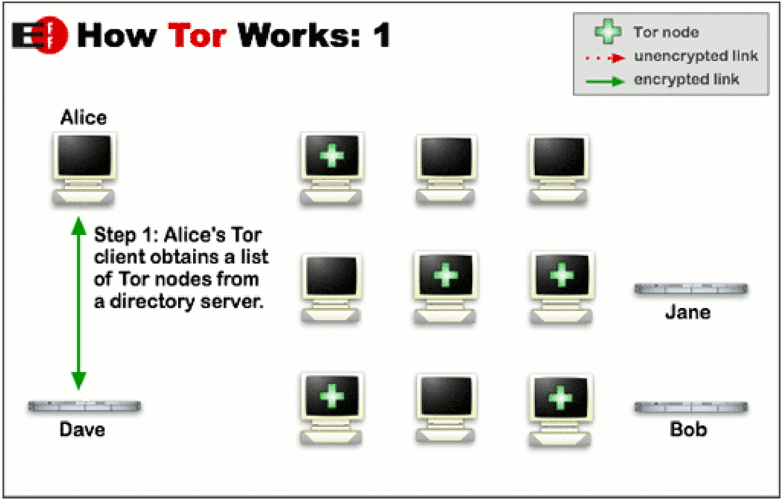
Core components:

- *Tor client*: computer, tablet or smartphone
- *Tor relay*: forwards data
- *Directory Authority*: manages Tor network

Core concept:

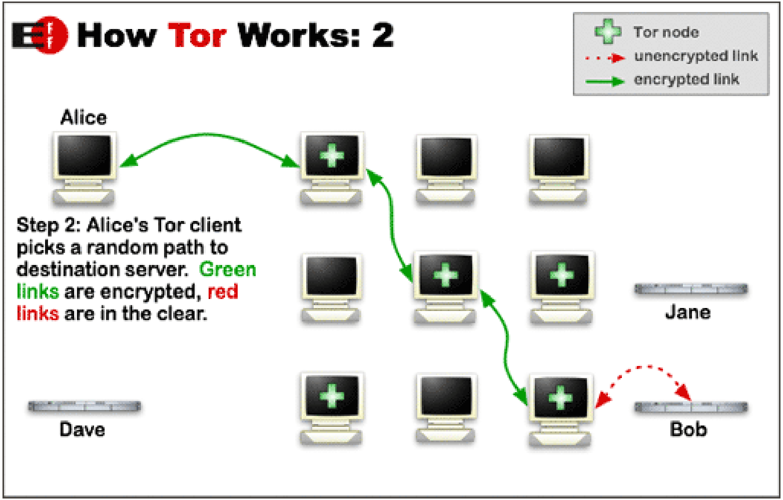
- data is encrypted in multiple layers
- one per relay on the path
- 3 additional hops to target

Tor



Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.

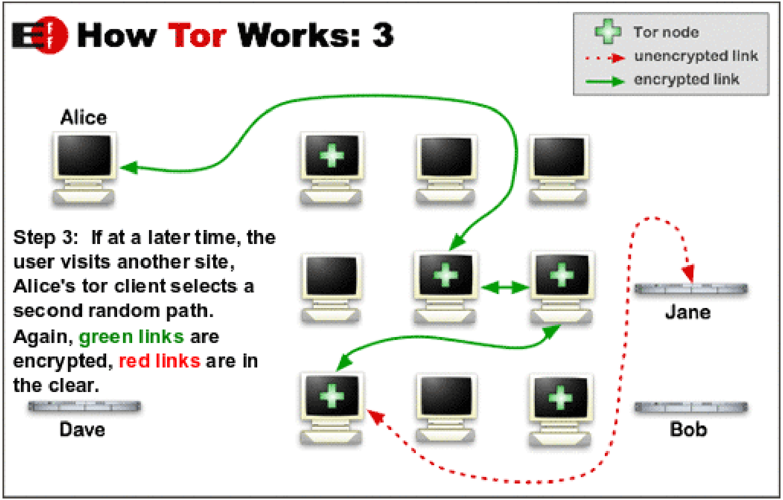
Tor



How Tor Works: 2

Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

Tor



How Tor Works: 3

Step 3: If at a later time, the user visits another site, Alice's tor client selects a second random path. Again, green links are encrypted, red links are in the clear.

Tor

Tor != Onion routing:

- Tor anonymizes TCP connections, everything that speaks SOCKS
- traffic in fixed-size cells, each 512 Byte
- relay bandwidth used for propability for clients to choose specific relay

Tor

Tor circuit[1]

- client chooses Tor path, last hop first
- by default 3 relays used
- each cell encrypted three times = 3 Tor relays = 1 circuit
- multiple TCP connections can share a path
- paths are built on-demand, but also preemptively

[1] term used interchangeably with Tor path

Tor

Threat model:

- first relay: has access to real user IP
- last relay (exit relay): certainly sees target IP
- exit might see communication content, if unencrypted

- "A global passive adversary is the most commonly assumed threat when analyzing theoretical anonymity designs."
- "But like all practical low-latency systems, Tor **does not protect** against such a strong adversary. "

Tor

Attacker might:

- attack actively and passively
- operate (many) relays
- create, modify, drop or delay traffic
- operate fraction of exit relays
- count packets and time between packets

But this assumptions are idealised:

- many documented attacks use exactly these vectors!
- yet still successful

Tor

Tor does not:

- no pure P2P, no UDP
- no steganography
- no protection against complex protocols
- no protection against traffic confirmation attacks
- no cover traffic
- no layer 8 protection

Tor

Cryptography in use:

- public key crypto: each relay has multiple pairs of keys
- elliptic curve on Ed25519, with SHA-256
- Diffie-Hellman key exchange (client-relays)
- TLS between Tor relays, using forward secrecy
- symmetric cryptography with AES in CTR mode

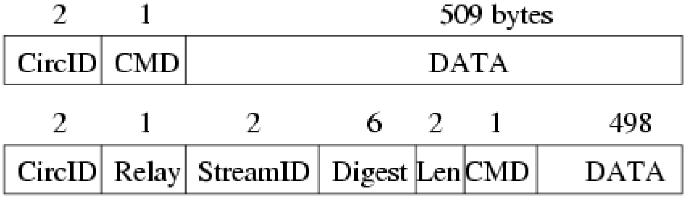
Tor

Cells:

- two major types of cells
- “control cell”: for the specific relay
- “relay cell”: will be forwarded, user data
- other types: link cell, relay early cell, ...
- all 512 byte in size

Tor

Control and relay cells:



2	1	509 bytes
CircID	CMD	DATA

2	1	2	6	2	1	498
CircID	Relay	StreamID	Digest	Len	CMD	DATA

Tor

Examples of control cells:

- “create”: for new tor path
- “padding”: for padding and keep-alive
- “destroy”: destroys path

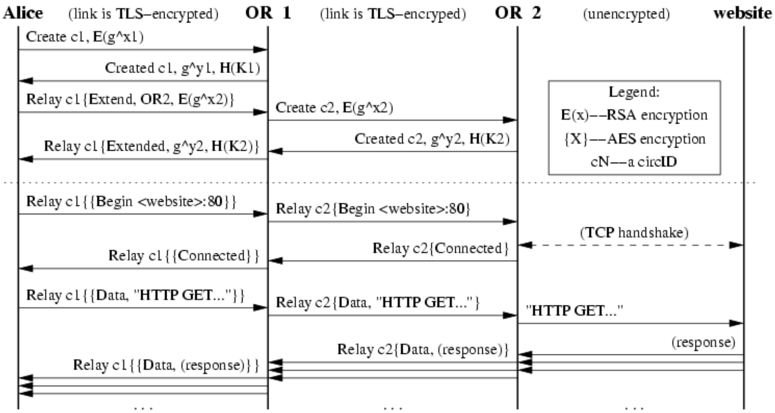
create cell contains first part of DH key exchange, encrypted with pubkey of relay

Tor

Examples of relay cells:

- “relay data”: data flow
- “relay begin”: to open a stream
- “relay extend”: to extend the circuit by a hop
- “relay end”: to close a stream cleanly
- “relay teardown”: to close a broken stream
- many more ...

Tor


Alice
(link is TLS–encrypted)
OR 1
(link is TLS–encryped)
OR 2
(unencrypted)
website
Create c1, E(g^x1)
Created c1, g^y1, H(K1)
Relay c1{Extend, OR2, E(g^x2)}
Create c2, E(g^x2)
Created c2, g^y2, H(K2)
Relay c1{Extended, g^y2, H(K2)}
Legend:
E(x)––RSA encryption
{X}––AES encryption
cN––a circID
Relay c1{{Begin <website>:80}}
Relay c2{Begin <website>:80}
(TCP handshake)
Relay c2{Connected}
Relay c1{{Connected}}
Relay c1{{Data, "HTTP GET..."}}
Relay c2{Data, "HTTP GET..."}
"HTTP GET..."
(response)
Relay c2{Data, (response)}
Relay c1{{Data, (response)}}
. . .
. . .
. . .

Tor

Pros:

- no central point of trust (as e.g. in VPN)
- universal
- robust, well-established

Cons:

- additional delay
- threat model still up-to-date?
- traffic is detectable using deep packet inspection (DPI)

Tor

Software:

- Open Source!
- socks proxy interface
- client needs no administrative privileges
- openly specified & documented

Usage:

- Computer: Tor Browser, Tails VM
- Android: Tor Browser for Android
- iOS: Onion Browser

Tor

Tails - The Amnesic Incognito Live System:

- live Linux (USB/DVD/SD) based on Debian
- all network connections go through Tor
- leaves no traces on local disks
- MAC spoofing
- RAM and VRAM get wiped on shutdown

Tor

Software in Tails:

- Tor Browser, OnionShare
- Thunderbird, Pidgin
- preconfigured for crypto (Electrum, GPG, OTR, LUKS, Truecrypt ...)
- LibreOffice, Gimp, ...

Goals and status:

- aims to be fail-safe
- current version is Tails 5.19.1

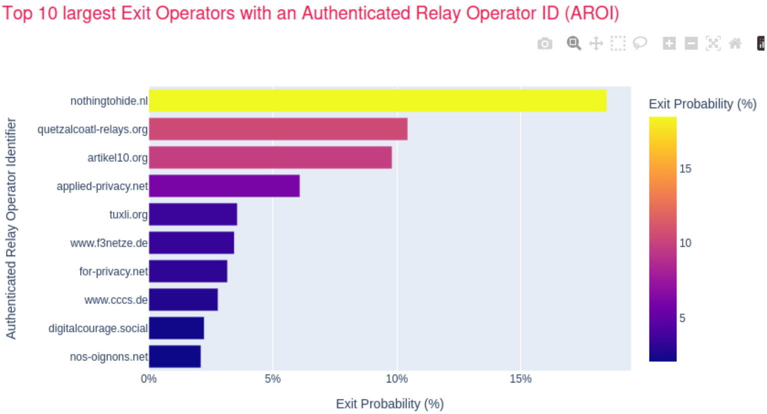
Tor

Tor relays/nodes:

- approx. 7500 24/7 world-wide
- approx. 2500 exit relays
- operated by volunteers
- highly configurable (bandwidth, “Exit Policy”, ...)

Tor

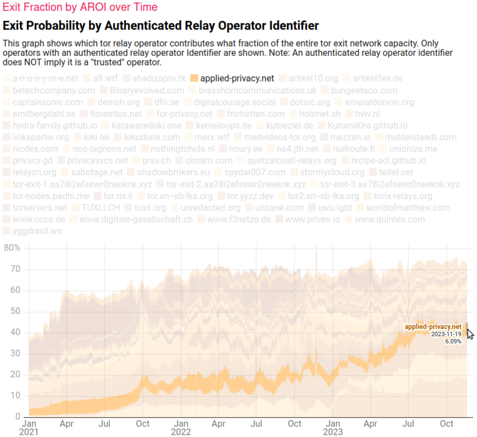
Exit relay families[2]:



[2] Source: OrNetStats

Tor

Exit relay families:



Exit Fraction by AROI over Time

Exit Probability by Authenticated Relay Operator Identifier

This graph shows which tor relay operator contributes what fraction of the entire tor exit network capacity. Only operators with an authenticated relay operator Identifier are shown. Note: An authenticated relay operator identifier does NOT imply it is a "trusted" operator.

a-n-o-n-y-m-e.net a9.wtf ahadizapto.tk applied-privacy.net artikel10.org artikel5ev.de
betechcompany.com Binaryevolved.com brasshorncommunications.uk bungeetaco.com
captainsonic.com demsh.org dfri.se digitalcourage.social dotsrc.org emeraldonion.org
emilbergdahl.se florentius.net for-privacy.net frichetten.com holonet.sh hviv.nl
hydra-family.github.io katawaredoki.one kerneloops.de kubieziel.de KurumiKira.github.io
linkspartei.org loki.tel lokodlare.com marx.wtf medvideos-tor.org mezzan.io middelstaedt.com
nicdex.com nos-oignons.net nothingtohide.nl noury.ee ns4.jth.net nullroute.fi onionize.me
privacy.gd privacysvcs.net prsv.ch qimam.com quetzalcoatl-relays.org recipe-acl.github.io
relayon.org sabotage.net shadowbrokers.eu spydar007.com stormycloud.org teitel.net
tor-exit-1.aa78i2efsewr0neeknk.xyz tor-exit-2.aa78i2efsewr0neeknk.xyz tor-exit-3.aa78i2efsewr0neeknk.xyz
tor-nodes.padhi.me tor.rix.li tor.xn--sb-lka.org tor.yyzz.dev tor2.xn--sb-lka.org torix-relays.org
torservers.net TUXLI.CH tuxli.org unredacted.org unzane.com uwu.lgbt worldofmatthew.com
www.cccs.de www.digitale-gesellschaft.ch www.f3netze.de www.privex.io www.quintex.com
yggdrasil.ws
80%
70
60
50
40
30
20
10
0
Jan 2021
Apr
Jul
Oct
Jan 2022
Apr
Jul
Oct
Jan 2023
Apr
Jul
Oct
applied-privacy.net
2023-11-19
6.09%

Tor

Tor directory authorities:

- 9 world-wide
- somehow semi-trusted
- vote on network consensus every hour (majority voting)

Tor

Exit Policy:

- ORPort, DirPort: open ports, must be reachable
- RelayBandwidthRate: maximum bandwidth
- ExitPolicy: which ports allowed
- e.g. for SSH and HTTPS:
  *ExitPolicy accept *:22, accept *:443, reject *:**

Running an exit relay in .at can cause you trouble!

Tor

Bridges:

- approx. 2000 available
- designed to bypass IP-based blockades
- Bridge at the beginning of Tor path
- should be hard to enumerate
- each client knows a handful
- run your own, privately

Tor

Tor Browser:

- based on Firefox Extended Support Release (ESR)
- many customized Firefox settings
- private mode
- compiled with Tor’s patches
- deterministic builds!!!

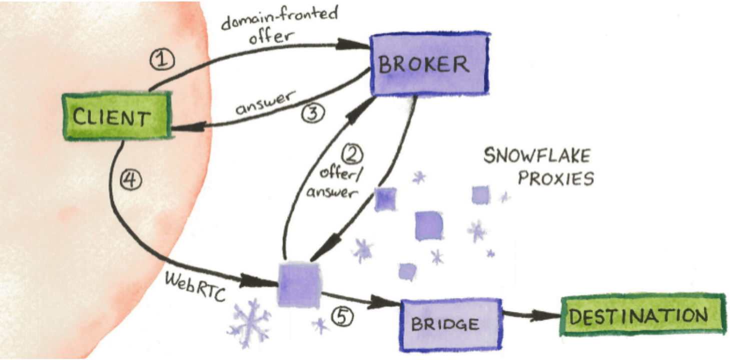
Tor

Pluggable Transports:

- Bridges have been enumerated before
- additional obfuscation on protocol
- currently supported: *obfs4*, *Snowflake (webRTC)*, *meek*, ...
- many more proposed: *TapDance*, *basket2*, ...

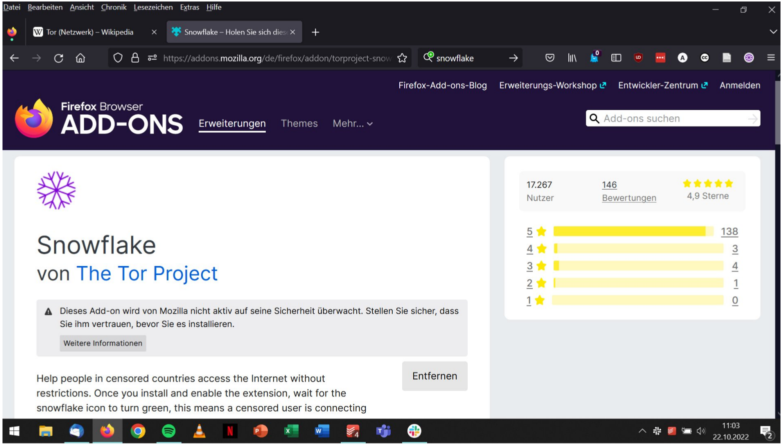
Tor

Snowflake:



Tor

Snowflake:



Snowflake

von The Tor Project

Dieses Add-on wird von Mozilla nicht aktiv auf seine Sicherheit überwacht. Stellen Sie sicher, dass Sie ihm vertrauen, bevor Sie es installieren.

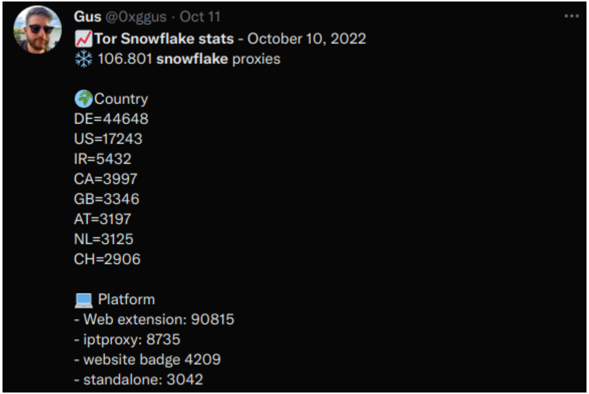
Help people in censored countries access the Internet without restrictions. Once you install and enable the extension, wait for the snowflake icon to turn green, this means a censored user is connecting

17.267	146	★★★★★
Nutzer	Bewertungen	4,9 Sterne

5 ★	138
4 ★	3
3 ★	4
2 ★	1
1 ★	0

Tor

Snowflake:



Gus @0xggus · Oct 11

📈**Tor Snowflake stats** - October 10, 2022
❄️ 106.801 **snowflake** proxies

🌍Country
DE=44648
US=17243
IR=5432
CA=3997
GB=3346
AT=3197
NL=3125
CH=2906

💻 Platform
- Web extension: 90815
- iptproxy: 8735
- website badge 4209
- standalone: 3042

Tor

Domain Fronting:

- is/was a neat trick to obfuscate traffic
- rerouting happened inside the CDN
- *meek* pluggable transport
- worked on large content delivery networks
- Amazon, Google, Azure, Cloudflare

Tor

How did domain fronting work:

- different domains in a single request
- one on DNS layer, e.g. www.legit.com
- one in HTTP header, e.g. www.blocked.com
- see paper Blocking-resistant communication through domain fronting

Tor

STEM library:

- python controller library for Tor
- connects to local Tor control port
- highly configurable
- used in many research projects, e.g. *exitmap*

Tor

Tor simulators, e.g. Shadow:

- network simulator for Tor
- “Tor in a Box”, runs on single PC
- uses Tor source code
- easier for experiments, compared to running private Tor network or Tor on Planetlab
- alternatives: TorPS, ExperimenTor

Tor

TorPS:

- Tor Path Simulator
- works with published network consensus
- Source here: `https://torps.github.io/`

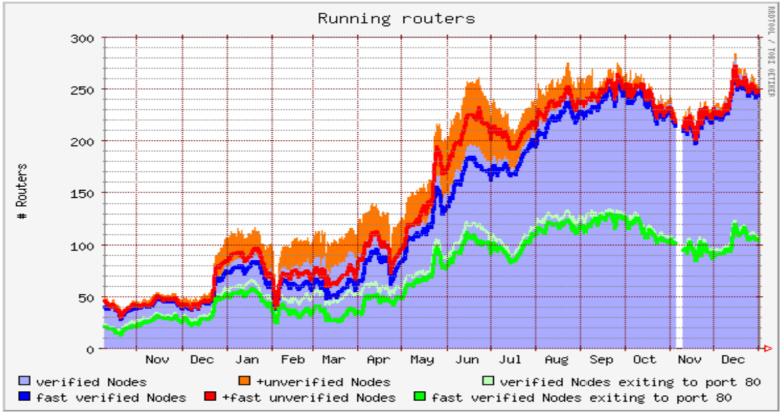
Tor

Web services:

- Tor2Web: access to Tor hidden services without Tor (*.onion.to)
- Tor Metrics: stats from the Tor network
- TorStatus: old page for stats
- Onionoo: web-based protocol for current Tor information

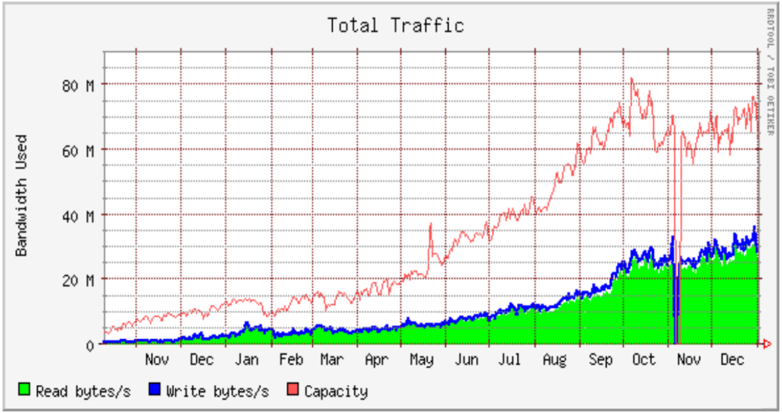
Tor

Tor in numbers, Relays 2004-2005:



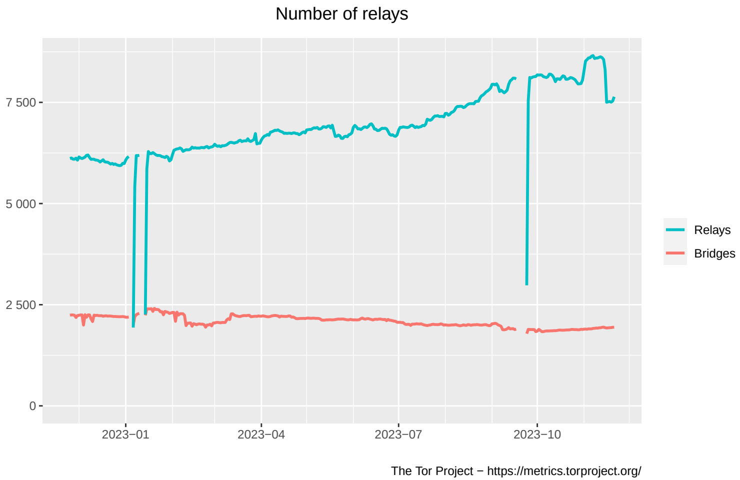
Tor

Tor in numbers, bandwidth 2004-2005:



Tor

Tor relays in the last 12 monhts:



Tor

Tor relays in the last 12 monhts:

Safeguarding the Tor network: our commitment to network health and supporting relay operators

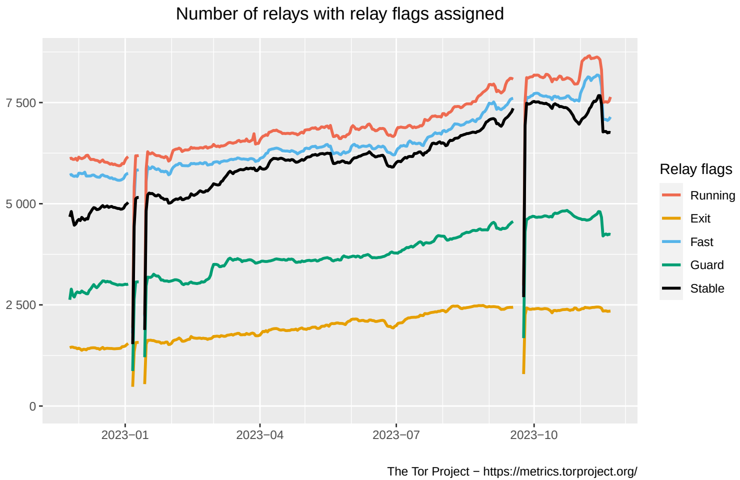
by isabela | November 20, 2023

[3]

[3] See blogpost here

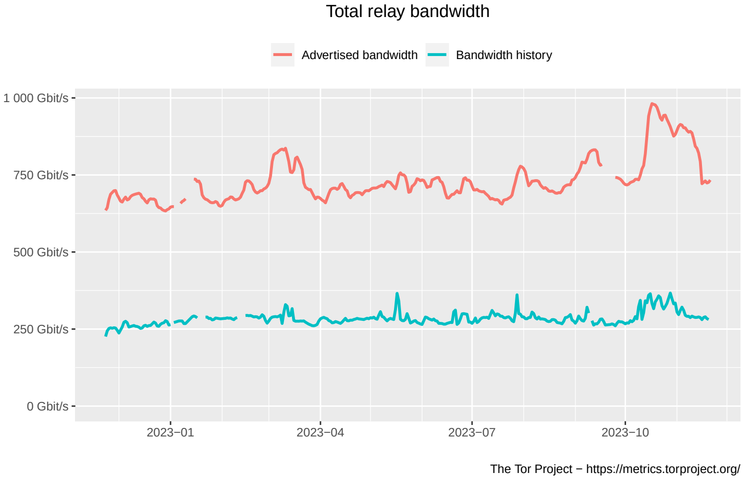
Tor

Relay flags in the last 12 months:



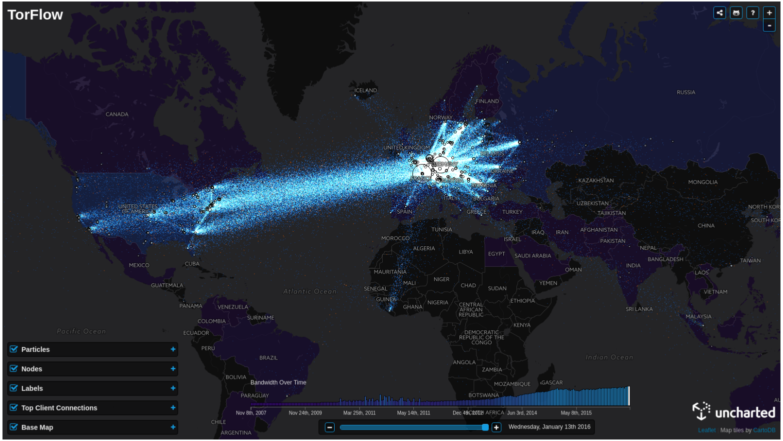
Tor

Bandwidth in the last 12 months:



Tor

TorFlow



☑ Particles

☑ Nodes

☑ Labels

☑ Top Client Connections

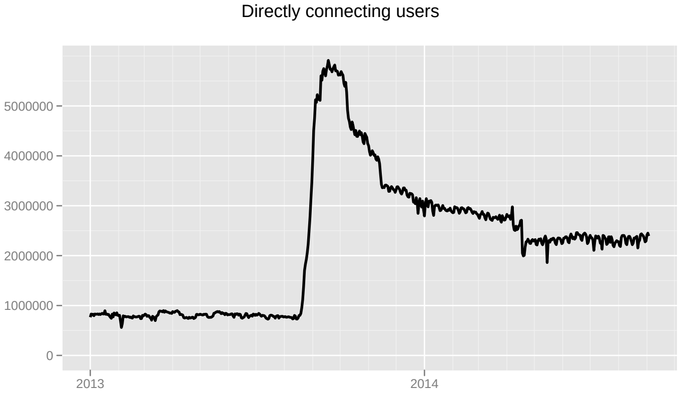
☑ Base Map

Bandwidth Over Time

Wednesday, January 13th 2016

Tor

Botnet (ab-)uses Tor[4]:



The Tor Project – https://metrics.torproject.org/

[4] See paper here

Tor

Tor metrics and more:

- all the data is available: consensus, IPs, ...
- publicly available
- since the beginning of the network, 2004!
- very interesting for science, courts and more

Tor

Tor

Host a Tor relay:

- fix IP and high bandwidth are always needed
- exit relay in .at can cause troubles!
- non-exit relay is without such risks

Host a Tor Bridge:

- not same IP as relay

Run Snowflake in your browser!

~~Hidden~~ Onion Services

Tor Onion services:

- allow anonymous server & services
- no geolocation based on IP possible
- two connections through Tor
- only reachable within the Tor network (or tor2web)
- can operate despite firewalls and NAT

Onion Services

Onion v3:

- switch to Ed25519 & ECC
- 56 characters instead of 16
- e.g. dnlfs2ifuz2s2yf3fc7r–dmsbhm6rw75euj35pac6ap25zgqad.onion
- base32(whole pubkey)

Onion Services

Previously:

- base32(first 80bit of sha1sum(pubkey))
- 16 chars, 2-7 and a-z
- e.g. `http://3g2upl4pq6kufc4m.onion`

.onion TLD:

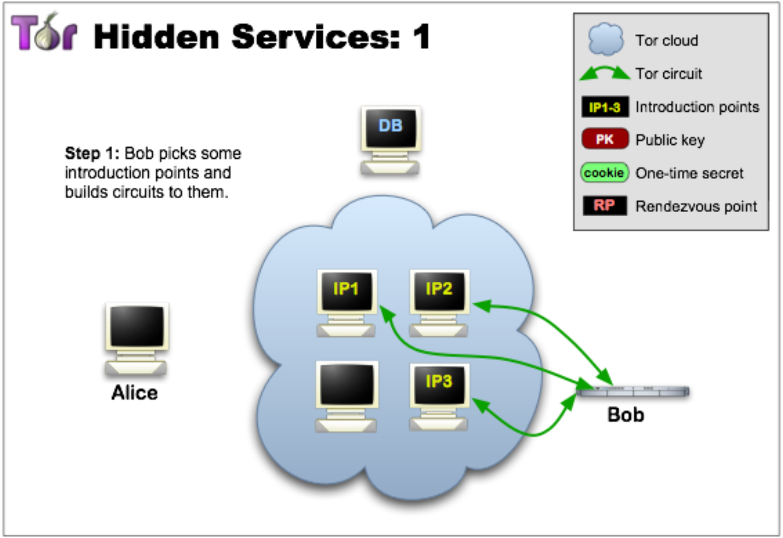
- .onion domain reserved by IETF
- means TLS certificates, Let's encrypt, ...

Onion Services

How-to:

- setup hidden Service locally
- server chooses Tor relays as introduction points
- server sends service descriptor (incl. pubkey) to HSDirectory
- client has to know .onion address
- client chooses rendezvous point
- rendezvous point sends message over introduction point
- connect-back vom Service (DoS Protection)

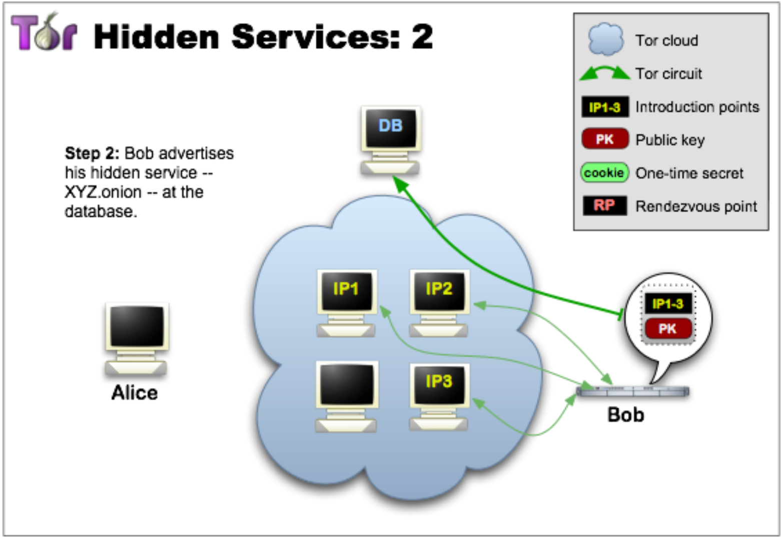
Onion Services



Hidden Services: 1

Step 1: Bob picks some introduction points and builds circuits to them.

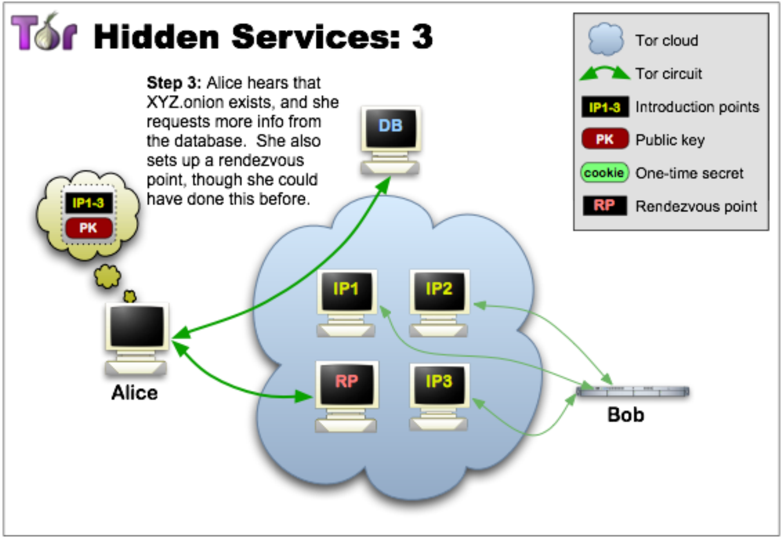
Onion Services



Tor Hidden Services: 2

Step 2: Bob advertises his hidden service -- XYZ.onion -- at the database.

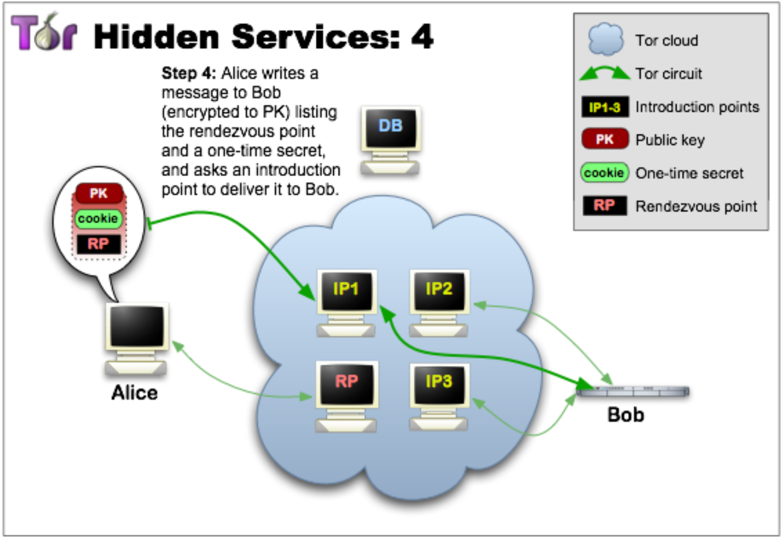
Onion Services



Tor Hidden Services: 3

Step 3: Alice hears that XYZ.onion exists, and she requests more info from the database. She also sets up a rendezvous point, though she could have done this before.

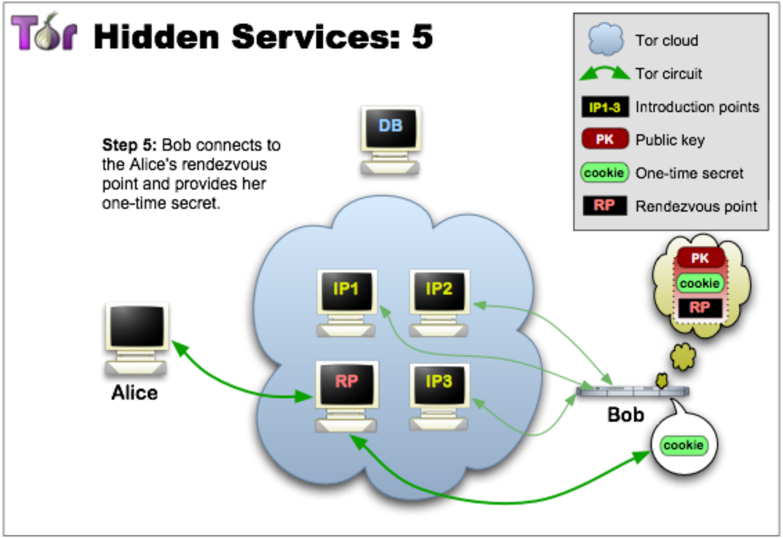
Onion Services



Tor Hidden Services: 4

Step 4: Alice writes a message to Bob (encrypted to PK) listing the rendezvous point and a one-time secret, and asks an introduction point to deliver it to Bob.

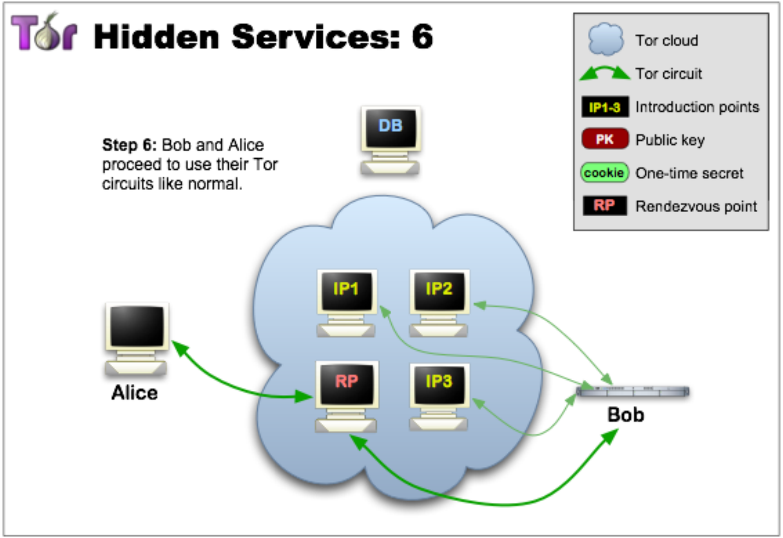
Onion Services



Hidden Services: 5

Step 5: Bob connects to the Alice's rendezvous point and provides her one-time secret.

Onion Services



Tor Hidden Services: 6

Step 6: Bob and Alice proceed to use their Tor circuits like normal.

Onion Services

Facebook:

- Facebook `https://facebookcorewwwi.onion`
- now `https://www.facebookwkhpilnemx\ j7asaniu7vnjjbiltxjqhye3mhbshg7kx5tfyd.onion`
- 1 million users per day (April 2016), that was 0,1%!!!
- they got the first TLS certificate for .onion TLD!
- not per-se anonymity, but bypass DPI and censorship

Other onions with valid TLS certificate:

- Duck Duck Go, SecureDrop by The Intercept, ...

Onion Services

Other services:

- Debian packages
- Wikileaks, GlobaLeaks
- DeadDrop, by Aaron Swartz
- SecureDrop, successor of DeadDrop
- NY Times, ProPublica, ProtonMail, ...
- “Dark Web”

About 3% of Tor traffic (4gbit)

Onion Services

One-click onion services:

- spawns hidden service
- fully automatic
- work out-of-the-box

Examples:

- Ricochet (deprecated!): anonymous chat
- OnionShare: for (large) files

Onion Services

“Infamous” onion services:

- Silk Road (2013)
- Silk Road 2.0 (2015)
- AlphaBay (2017)
- Freedom Hosting (2013)
- Playpen (2015)
- Childs Play (2017)

Onion Services

Silkroad:

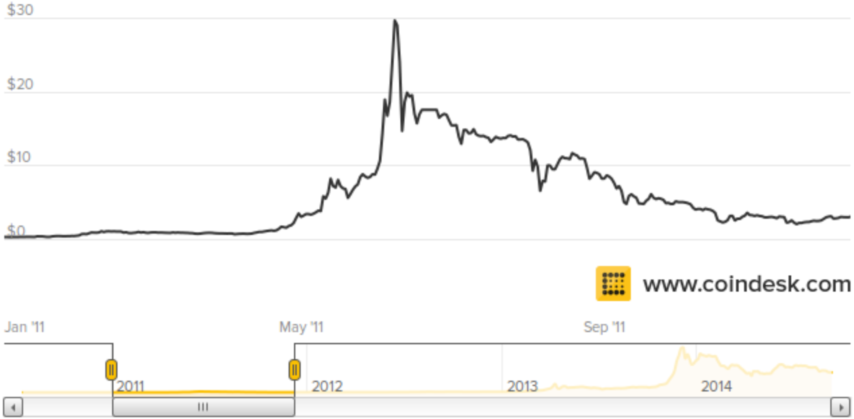
- Ebay/Amazon for drugs and other legal and illegal goods
- Bitcoin for payment
- anything was possible, except things that can harm others
- e.g. no child porn, weapons, counterfeit money or CC-information

Onion Services

Silk Road popularity:

- article in Gawker in June 2011 made it really popular
- U.S. Senator Charles Schumer wants it taken down
- reason for first rush on Bitcoin
- got shutdown in 2013, 20 others spawned

Onion Services



Onion Services

But how popular, really?

- “Traveling the silk road: a measurement analysis of a large anonymous online marketplace”
- Nicolas Christin, CMU
- paper at WWW’13
- 6 months of daily crawls
- more then 1 million $ per month (estimated)
- dataset is available[5]

[5] https://arima.cylab.cmu.edu/sr/

Onion Services

Follow-up paper:

- “Measuring the Longitudinal Evolution of the Online Anonymous Marketplace Ecosystem”[6]
- paper at USENIX Security’15
- 16 marketplaces, 2013-2015
- e.g. Silk Road 2.0, Agora, Evolution, Utopia, Sheep Marketplace, ...
- Blake Benthall arrested in Nov 2014 as Silk Road 2.0 admin

[6] View the presentation here

Onion Services

Silkroad:

- Ross Ulbricht arrested in October 2013
- hidden service website was shut down
- more than 150,000 buyers, 3900 sellers
- FBI seized more than 170k Bitcoins
- sentenced to life in prison without possibility of parole

Onion Services

Interesting for digital forensics:

- Ulbricht was logged in an admin area as he got arrested[7]
- server in Island got imaged, twice!

not really clear how the FBI obtained IP address:

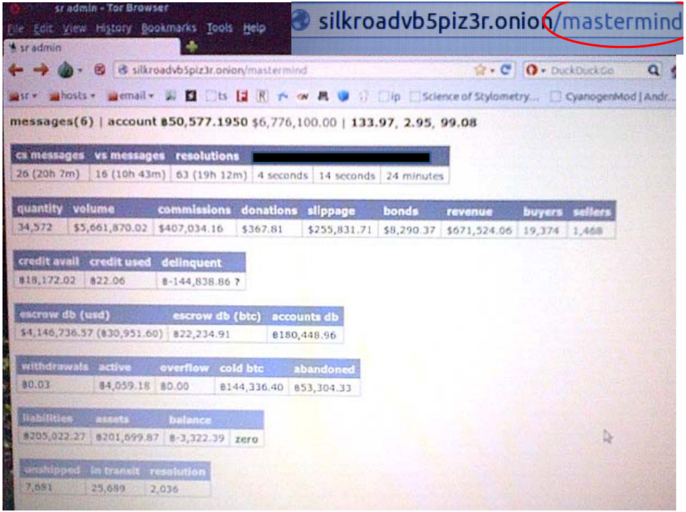
- IP leaked over login field resp. CAPTCHA resp. Header?
- not really clear how it worked[8]
- but numerous times mis-configured[9]

[7] See here

[8] See here

[9] See here

Onion Services


sr admin - Tor Browser
File Edit View History Bookmarks Tools Help
sr admin
silkroadvb5piz3r.onion/mastermind
DuckDuckGo
sr
hosts
email
ts
ip
Science of Stylometry...
CyanogenMod | Andr...
silkroadvb5piz3r.onion/mastermind

messages(6) | account ฿50,577.1950 $6,776,100.00 | 133.97, 2.95, 99.08

cs messages	vs messages	resolutions	[redacted]		
26 (20h 7m)	16 (10h 43m)	63 (19h 12m)	4 seconds	14 seconds	24 minutes

quantity	volume	commissions	donations	slippage	bonds	revenue	buyers	sellers
34,572	$5,661,870.02	$407,034.16	$367.81	$255,831.71	$8,290.37	$671,524.06	19,374	1,468

credit avail	credit used	delinquent
฿18,172.02	฿22.06	฿-144,838.86 ?

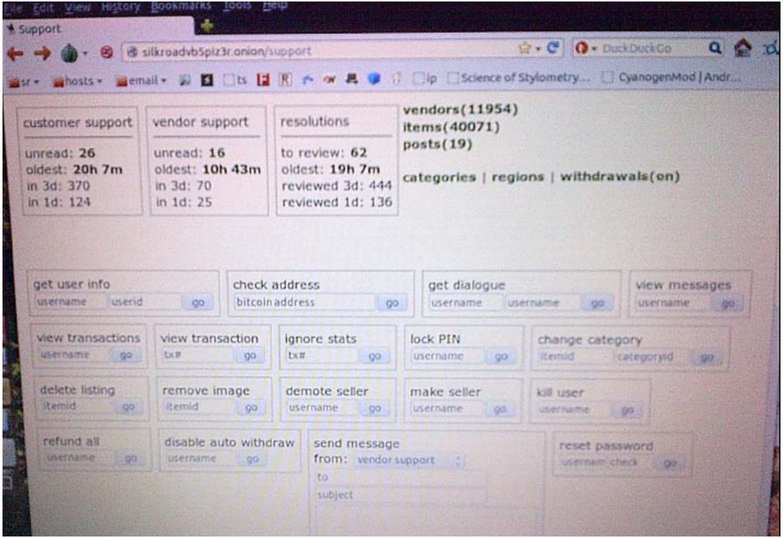
escrow db (usd)	escrow db (btc)	accounts db
$4,146,736.57 (฿30,951.60)	฿22,234.91	฿180,448.96

withdrawals	active	overflow	cold btc	abandoned
฿0.03	฿4,059.18	฿0.00	฿144,336.40	฿53,304.33

liabilities	assets	balance	
฿205,022.27	฿201,699.87	฿-3,322.39	zero

unshipped	in transit	resolution
7,681	25,689	2,036

Onion Services



customer support
unread: 26
oldest: 20h 7m
in 3d: 370
in 1d: 124

vendor support
unread: 16
oldest: 10h 43m
in 3d: 70
in 1d: 25

resolutions
to review: 62
oldest: 19h 7m
reviewed 3d: 444
reviewed 1d: 136

vendors(11954)
items(40071)
posts(19)

categories | regions | withdrawals(on)

Onion Services

Freedom Hosting:

- anonymous webhosting
- webspace without any restrictions
- among others: a lot of child pornography
- Anonymous DDoS “Operation Darknet” in 2011
- shutdown in July 2013

Onion Services

Freedom Hosting Malware:

- "Down for maintenance"
- shipped exploit from FBI (CVE-2013-1690)
- arbitrary code execution
- targeted Firefox 17 ESR on Windows
- issue was fixed for a month in the most recent version of Firefox
- "Magneto": sends MAC and real IP as HTTP request

Onion Services

“Operation Torpedo”:

- not the first exploit used against hidden services
- at the beginning of 2012 another
- monitored 3 hidden services over 5 months
- identified 25 US-user
- among others a Cybersecurity director of an US government agency
- less impact, thus unnoticed[10]

[10] See also this link

Onion Services

AlphaBay & Hansa takedowns:

- Operation Bayonet, joint work of FBI/Europol
- alleged AlphaBay operator was Alexandre Cazes, 26
- killed himself after 1 week in Bangkok prison
- 10,000+ users signed up for Hansa after AlphaBay takedown
- modified and operated by Dutch police weeks before

Onion Services

Playpen:

- Operation Pacifier, 2015
- taken over and operated by FBI for 13 days
- 3229 cases created by Europol, 50 in .at[11]
- creator sentenced to 30 years in prison[12]
- two admins for 20 years, each

[11] See also here

[12] Source here

Onion Services

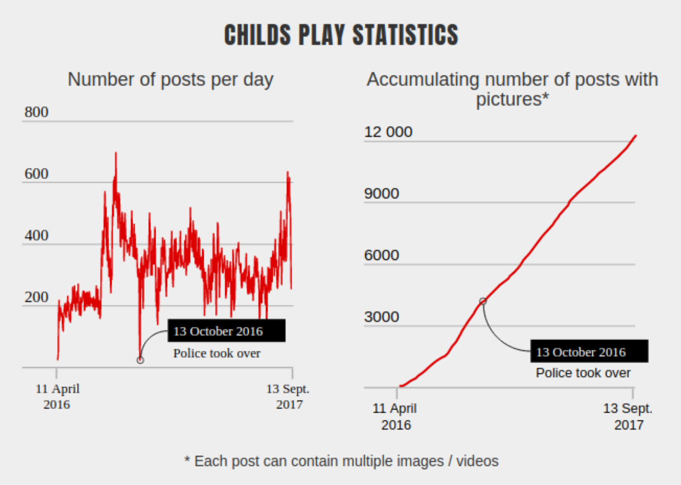
Childs Play:

- Operation Artemis, September 2017
- image upload for avatar leaked IP
- Norwegian newspaper found server in Australia
- operated by Task Force Argos, for almost a year!
- full story[13]

[13] See also here

Onion Services

Ethics, anyone?



Attacking Tor

Attacking Tor

Selection of attacks against:

- … Tor itself
- … Tor users
- … hidden services
- active and passive

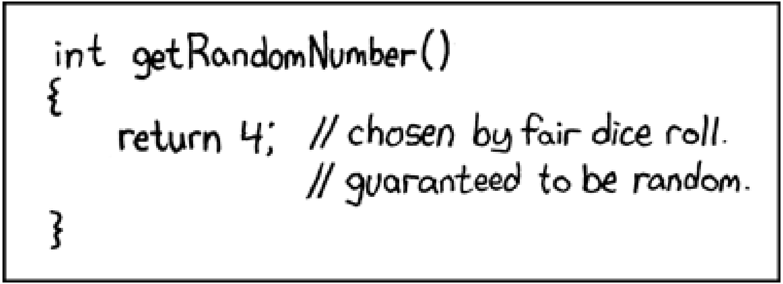
Reference library: Freehaven Anonbib[14]

[14] http://freehaven.net/anonbib/

Attacking Tor

Debian weak keys:

- OpenSSL weakness in May 2008
- back then 6 Directory nodes, 3 running Debian
- attack on consensus would have required 4



```
int getRandomNumber()
{
     return 4;  // chosen by fair dice roll.
                // guaranteed to be random.
}
```

Attacking Tor

Heartbleed, CVE-2014-0160[15]:

- browser not affected (libnss)
- Tor client could leak info to Guard Node
- relays and bridges: Onion keys
- hidden service: identity key
- DirAuthority medium key, Orbot, ...
- Directory Authority use offline long-term key

[15] Source: here

Attacking Tor

BitTorrent & Tor:

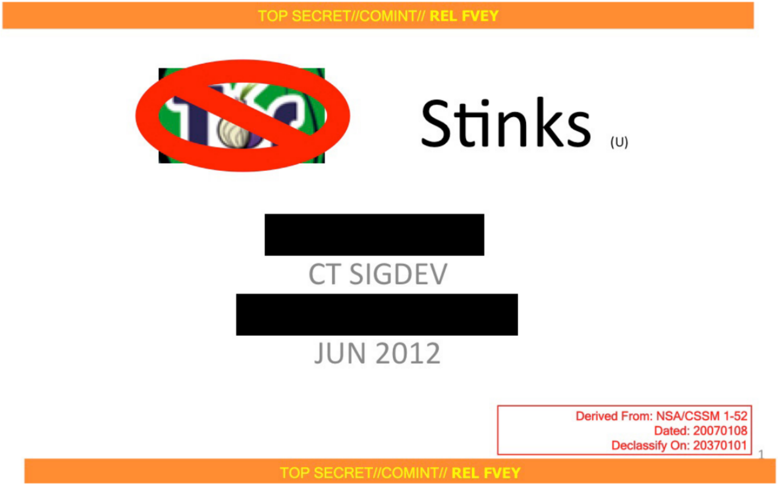
- “One Bad Apple Spoils the Bunch: Exploiting P2P Applications to Trace and Profile Tor Users”
- paper at LEET 2011
- bad idea, and not just because it can overflow networks
- both tracker-only and tracker & content over Tor
- can lead to deanonymization
- can also deanonymize other streams, e.g. HTTPS

Attacking Tor

zmap and *masscan*:

- 10 gbit port scanners, scan IPv4 in few minutes
- Tor used to listen on port 9001 and 443
- zmap could enumerate 86% of all bridges
- countermeasure: random ports with bridge announcement

Attacking Tor



TOP SECRET//COMINT// **REL FVEY**

Stinks (U)

[redacted]

CT SIGDEV

[redacted]

JUN 2012

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20370101

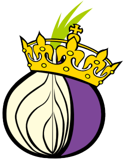
1

TOP SECRET//COMINT// **REL FVEY**

Attacking Tor

NSA: Tor = King of Anonymity[16]

- “Still the King of high secure, low latency Internet Anonymity”
- no new attack found in these files
- but: Tor users can be tagged
- some of the files by the NSA on Tor: http://media.encrypted.cc/files/nsa/



[16] http://www.theguardian.com/world/2013/oct/04/nsa-gchq-attack-tor-network-encryption

Attacking Tor

One of our papers:

- “Spoiled Onions: Exposing Malicious Tor Exit Relays”
- published at PETS Symposium 2014

Two scanners:

- *exitmap*: SSL & HTTPS MITM
- *HoneyConnector*: plaintext credetials in FTP, IMAP
- 6+ months runtime, starting fall 2013
- identified 65 malicious Exit Relays

Attacking Tor

Problems:

- Exit relay can read or modify traffic
- if in plain: FTP, IMAP, SMTP, DNS
- can also change TLS certificate (MITM)
- again and again a problem in Tor (embassy email accounts, ...)

Attacking Tor

exitmap:

- extremely fast!
- uses 2-hop path
- scan all exit relays in less then 1 minute
- Python & STEM
- identified 40 malicious exit relays
- most of them did HTTPS MITM
- but also SSH MITM, *sslstrip*, DNS

Attacking Tor

HoneyConnector:

- uses unique credentials for FTP and IMAP
- conducted more than 54,000 connections
- identified 27 sniffing exit relays
- 255 login attempts, with 128 credentials
- up to 2 months after the connection

Attacking Tor

Spoiled Onions aftermath:

- notified Tor, relays got BadExit flag
- identified 3 groups of cooperating exit relays
- paper and sources here

Countermeasures:

- Firefox extensions on about:certerror
- compares TLS certificates over another Tor path
- user education, no plaintext protocols
- pinning, HSTS, DANE

Attacking Tor

Attack on hidden services[17]:

- announced (but not delivered) presentation at Blackhat 2014
- deanonymised users for hidden service (traffic confirmation attack)
- mixture of relay & relay-early cells
- sybil attack with 115 fast-non-exit relays
- 3,000$ per month, for 6 months

[17] Source: here

Attacking Tor

Tor and the structure of the Internet:

- assumptions in Tor are sometimes simplified
- Internet made of Autonomous Systems (AS) and Internet Exchange Points (IXP)
- attackers can control both
- traffic correlation is easy for the first and last relay

Attacking Tor

Tor circuit constraints:

- do not use two relays in the same “family” of nodes
- do not use two relays in the same /16 network
- and exit bandwidth varies per port

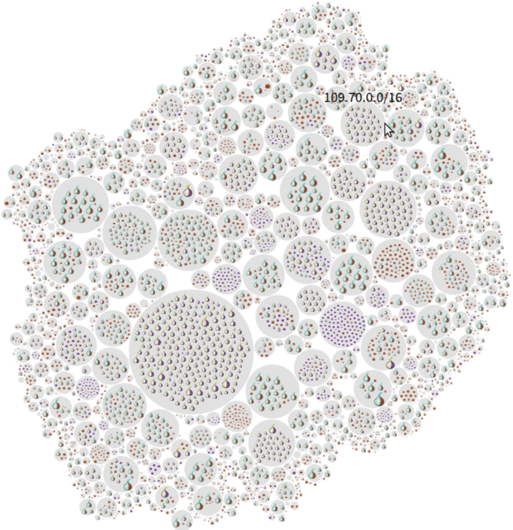
Attacking Tor

But how easy?

- 80% of users deanonymisable within 6 months[18]
- 100% of users within 3 months (for large AS)
- 95% of users in 3 months (for IXP)
- two AS: 1 day instead of 3 months
- evaluated against five different usage types (time and port)

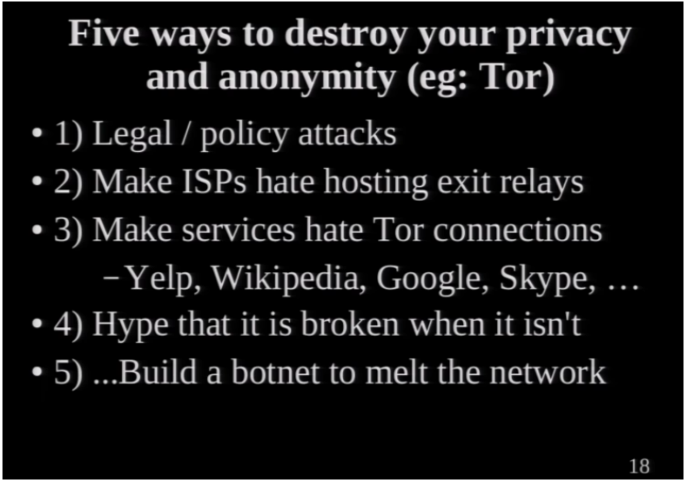
[18] Source: “Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries”, CCS 2013

Attacking Tor



2385 network families (/16) with 6757 relays (4250 visible)

Attacking Tor (30c3)



Five ways to destroy your privacy and anonymity (eg: Tor)

- 1) Legal / policy attacks
- 2) Make ISPs hate hosting exit relays
- 3) Make services hate Tor connections
  - –Yelp, Wikipedia, Google, Skype, …
- 4) Hype that it is broken when it isn't
- 5) ...Build a botnet to melt the network

Using Tor

Using Tor

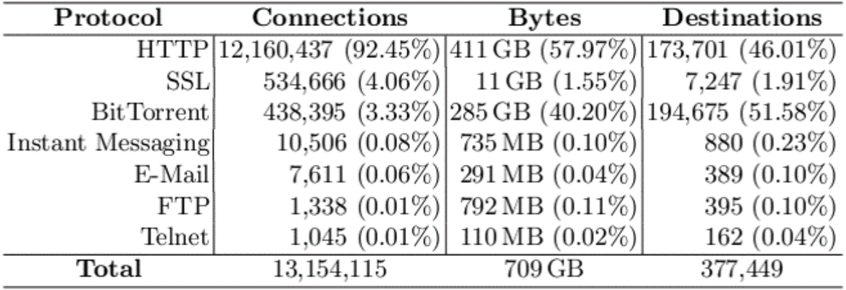
How to count users:

- counting access to network consensus[19]
- only a few relays collect this information
- extrapolated on the entire Tor network
- not exact, but good estimate
- geolocation using GeoIP

[19] Details here und here.

Using Tor

“Shining Light in Dark Places: Understanding the Tor Network”, PETS 2008



Protocol	Connections	Bytes	Destinations
HTTP	12,160,437 (92.45%)	411 GB (57.97%)	173,701 (46.01%)
SSL	534,666 (4.06%)	11 GB (1.55%)	7,247 (1.91%)
BitTorrent	438,395 (3.33%)	285 GB (40.20%)	194,675 (51.58%)
Instant Messaging	10,506 (0.08%)	735 MB (0.10%)	880 (0.23%)
E-Mail	7,611 (0.06%)	291 MB (0.04%)	389 (0.10%)
FTP	1,338 (0.01%)	792 MB (0.11%)	395 (0.10%)
Telnet	1,045 (0.01%)	110 MB (0.02%)	162 (0.04%)
Total	13,154,115	709 GB	377,449

Using Tor

HTTP Usage:

- “Tor HTTP usage and Information Leakage”, IFIP CMS 2010
- analysing HTTP GET requests
- largest group: social networks, file sharing, search engines
- 80% did not use TorButton or Tor Browser
- problematic downloads: .exe, .pdf, .zip, ...

Using Tor

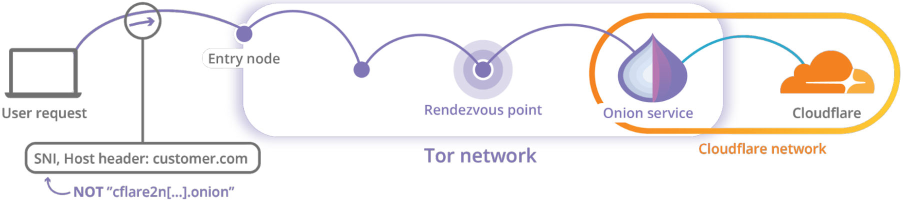
Using Tor correctly:

- use supported software: Tor Browser, Tails, Qubes
- check signatures!
- use separate device or network exclusively for Tor
- air-gapped machine for critical activities

Using Tor

Since 2018:

- Cloudflare now runs on Onions[20]
- new HTTP header, *alt-srv*
- enabled by default
- no changes required for site operators



[20] Source: here

Questions?