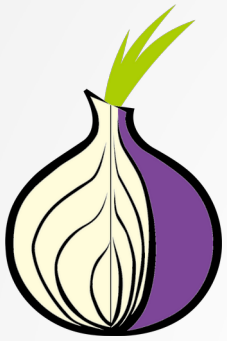


Teil 2: Fortgeschrittenes Tor

Benni Lason

Dr. Christoph Zimmermann

FraLUG, 28. Juli 2020

Übersicht

1. Fortgeschrittene Tor-Konfiguration
2. Monitoring
3. Pluggable Transports
4. Software Qualität & Angriffsoberflächenanalyse
5. Weitere Beobachtungen

Rekapitulation Teil 1

- Tor ist das Project, Operators, User, Software,...
- Anonymisierung durch Onion-Routing
- Offener Code, freie Doku, Diversität bei Operators & Relays (mehr gewünscht)
- Onion Services für jeden Dienst möglich
- Internationales Netzwerk gegen Zensur & Überwachung mit unterschiedlichen Einsatzzwecken & Szenarios
- Repression gegen Operator & User
- Tor nutzen, Tor erklären, Tor verteidigen, Tor machen, (Netz)Politik checken

→ Vortrag Teil 1 https://lug-frankfurt.de/VorTrag

torrc

- Die Konfigurationsdatei für tor
- Steuerung des Verhaltens des tor Prozesses
- Steuerung der Rolle im Netzwerk

/etc/tor/torrc oder tor --defaults-torrc PFAD

/home/user/tor-browser_de/Browser/TorBrowser/Data/Tor (do not edit)

→ tor-manual:

https://2019.www.torproject.org/docs/tor-manual.html.en

torrc

Uncomment this to start the process in the background... or
use

--runasdaemon 1 on the command line. This is ignored on

Windows;

see the FAQ entry if you want Tor to run as an NT service.

RunAsDaemon 1

The directory for keeping all the keys/etc. By default, we store

things in $HOME/.tor on Unix, and in Application Data\tor on
Windows.

#DataDirectory @LOCALSTATEDIR@/lib/tor

torrc

```
## Required: what port to advertise for incoming Tor connections.
ORPort 9001
## If you want to listen on a port other than the one advertised in
## ORPort (e.g. to advertise 443 but bind to 9090), you can do it as
## follows.  You'll need to do ipchains or other port forwarding
## yourself to make this work.
# ORPort 443 NoListen
# ORPort 127.0.0.1:9090 NoAdvertise
## If you want to listen on IPv6 your numeric address must be
## explictly
## between square brackets as follows. You must also listen on IPv4.
ORPort [2001:DB8::1]:9050
```

torrc

```
####### This section is just for location-hidden services #######
## Once you have configured a hidden service, you can look at the
## contents of the file ".../hidden_service/hostname" for the address
## to tell people.
##
## HiddenServicePort x y:z says to redirect requests on port x to the
## address y:z.
#HiddenServiceDir @LOCALSTATEDIR@/lib/tor/hidden_service
#HiddenServicePort 80 127.0.0.1:80
#HiddenServiceDir @LOCALSTATEDIR@/lib/tor/other_hidden_service/
#HiddenServicePort 80 127.0.0.1:80
#HiddenServicePort 22 127.0.0.1:22
```

torrc Beispiel bridge

- /etc/tor/torrc

  BridgeRelay 1

  #Replace "TODO1" with a Tor port of your choice.
  #This port must be externally reachable.
  #Avoid port 9001 because it's commonly associated with Tor and censors #may be
  #scanning the Internet for this port.
  ORPort 443
  ORPort [2001:DB8::1]:80

  ServerTransportPlugin obfs4 exec /usr/bin/obfs4proxy

  #Replace "TODO2" with an obfs4 port of your choice.
  #This port must be externally reachable and must be different from the one #specified for
  #ORPort.
  #Avoid port 9001 because it's commonly associated with Tor and censors #may be
  #scanning the Internet for this port.
  ServerTransportListenAddr obfs4 0.0.0.0:5061

  RelayBandwidthRate 1000 KBytes # Kilobyte/Sekunde
  RelayBandwidthBurst 2000 KBytes # Kilobyte/Sekunde

torrc Beispiel bridge

- */etc/tor/torrc*

  # Local communication port between Tor and obfs4. Always set this to #"auto".
  #"Ext" means "extended", not "external". Don't try to set a specific port #number,
  #nor listen on 0.0.0.0.
  ExtORPort auto

  # Replace "<address@email.com>" with your email address so we can #contact
  #you if there are problems with your bridge.
  # This is optional but encouraged.
  ContactInfo <address@email.com>

  #Logfile
  Log notice file var/log/alle_logs/tor/notices.log
  # Pick a nickname that you like for your bridge. This is optional.
  Nickname MeineTorBoxhatSuperKraefte

torrc Beispiel SSH

- /etc/tor/torrc

  ####### This section is just for location-hidden services #######
  #HiddenServicePort 80 127.0.0.1:80
  HiddenServiceDir /var/lib/tor/ssh_zwiebel/
  #HiddenServicePort 80 127.0.0.1:80
  HiddenServicePort 22 #127.0.0.1:22 (?ungetestet mit IPv6 Port)

- Speichern, Schließen, tor neu starten (systemctl restart tor oder ohne systemd anders)
- Tor hat Keys und notwendige Dateien erstellt
- OnionService Adresse abrufbar mit z.B. cat /var/lib/tor/ssh_zwiebel/hostname

  Output z.B.: 4llc0ps4r3sh1t4ndn33ds0m0th3rj0bs.onion

- Wenn der Dienst läuft, dann einfach mit z.B. torsocks verbinden:

  user:~$ torsocks ssh usermaschine@4llc0ps4r3sh1t4ndn33ds0m0th3rj0bs.onion

torrc Beispiel SSH

- SSH aus dem Clearnet entfernen

/etc/ssh/sshd_config

#Port 22
#AddressFamily any
ListenAddress 127.0.0.1 #(IPv4)
ListenAddress :: #(IPv6 – nicht getestet, sollte aber OK)

Weitere Dateien

- Wo sind die Logs?

  Wo ihr wollt! → torrc | default: /var/log/tor/

- Sonstige Verzeichnisse:

  /var/lib/tor/keys → Keys

  OnionService var/www/ ...

- swapfile: Aufpassen mit Swapping bei Bridge/Relay

  Bitte swapping abschalten!

Monitoring

- Nyx – https://nyx.torproject.org
- Daten kommen von lokalen Diensten (netstat, etc.)
- Nyx kommuniziert mit der tor Anwendung über lokalen Port (Cookie oder hashed Passwort)

```
## The port on which Tor will listen for local connections from Tor
## controller applications, as documented in control-spec.txt.
ControlPort 9051
## If you enable the controlport, be sure to enable one of these
## authentication methods, to prevent attackers from accessing it.
#HashedControlPassword
#16:872860B76453A77D60CA2BB8C1A7042072093276A3D701AD684053EC4
#C
CookieAuthentication 1
```

Monitoring


nyx - caersidi (Linux 4.4.0-78-generic) Tor 0.3.0.10 (recommended) cpu: 13.3% tor, 1.2% nyx mem: 387 MB (39.1%) pid: 6905
caersidi - 208.113.165.162:1443, Dir Port: 1444, Control Port (cookie): 9051 fingerprint: 3EA8E960F6B94CE30062AA8EF02894C00F8D1E66
flags: Fast, Guard, HSDir, Running, Stable, V2Dir, Valid exit policy: reject *:*
page 1 / 5 - m: menu, p: pause, h: page help, q: quit
Bandwidth (limit: 5 MB/s, burst: 1 GB/s, measured: 9.2 KB/s):
Download (2.2 MB/sec - avg: 2.1 MB/sec, total: 1.4 TB):
Upload (2.2 MB/sec - avg: 2.2 MB/sec, total: 1.4 TB):
3 MB
2 MB
1 MB
0 B
10s 20 30 40 50 1m
Events (TOR/NYX NOTICE-ERR):
17:17:20 [NOTICE] Since startup, we have initiated 0 v1 connections, 0 v2 connections, 1 v3 connections, and 330705 v4 connections; and received 1568 v1 connections, 15499 v2 connections, 38193 v3 connections, and 478987 v4 connections.
17:17:20 [NOTICE] Circuit handshake stats since last time: 349862/349862 TAP, 900901/900901 NTor.
17:17:20 [NOTICE] Heartbeat: Tor's uptime is 7 days 11:59 hours, with 7275 circuits open. I've sent 1382.08 GB and received 1345.52 GB. [2 duplicates hidden]
11:17:20 [NOTICE] Since startup, we have initiated 0 v1 connections, 0 v2 connections, 1 v3 connections, and 321652 v4 connections; and received 1500 v1 connections, 14823 v2 connections, 36347 v3 connections, and 460897 v4 connections.
11:17:20 [NOTICE] Circuit handshake stats since last time: 36896/36896 TAP, 912212/912212 NTor.
05:17:20 [NOTICE] Since startup, we have initiated 0 v1 connections, 0 v2 connections, 1 v3 connections, and 312315 v4 connections; and received 1455 v1 connections, 14169 v2 connections, 34219 v3 connections, and 444491 v4 connections.
05:17:20 [NOTICE] Circuit handshake stats since last time: 117335/117335 TAP, 854724/854724 NTor.
November 05, 2017
23:17:20 [NOTICE] Since startup, we have initiated 0 v1 connections, 0 v2 connections, 1 v3 connections, and 303540 v4 connections; and received 1406 v1 connections, 13839 v2 connections, 32923 v3 connections, and 429522 v4 connections.
23:17:20 [NOTICE] Circuit handshake stats since last time: 35622/35622 TAP, 956050/956050 NTor.
23:17:20 [NOTICE] Heartbeat: Tor's uptime is 6 days 17:59 hours, with 11084 circuits open. I've sent 1234.20 GB and received 1203.79 GB. [3

Los geht‘s!

- Tor nutzen
- Tor erklären
- Tor verteidigen
- Tor machen
- (Netz)Politik überprüfen, wählen und machen

Los geht‘s!

Links

Kontakt cryptocation@mailbox.org – ccxmpp@jabber.systemli.org – lugfrankfurt.de Mailingliste

Project

https://www.torproject.org/about/history/
https://community.torproject.org/
https://trac.torproject.org/projects/tor/wiki/TorRelayGuide

Netzwerk/Design

Tor Metrics https://metrics.torproject.org

Riseup Guide https://riseup.net/de/security/network-security/tor/onionservices-best-practices

Applikationen / OS / Tools

Tails OS https://tails.boum.org

Whonix OS https://whonix.org

torsocks – Linux Manual https://linux.die.net/man/8/torsocks

OnionShare https://onionshare.org/ / https://github.com/micahflee/onionshare

Orbot – div. AppStores, Google PlayStore

Briar Messenger – Google PlayStore

*Proxy Einstellungen in div. Apllikationen

Tor Suchmaschine https://ahmia.fi - http://msydqstlz2kzerdg.onion/

Diverses

https://www.privacy-handbuch.de/handbuch_24f.htm

https://www.telekom.com/de/verantwortung/datenschutz-und-datensicherheit/archiv-datenschutznews/news/brauchen-wir-das-darknet-501090

https://medium.com/beyond-install-tor-signal/case-file-jeremy-hammond-514facc780b8

Pluggable transports

- Motivation: DPI durch ISPs & Zensoren
- Idee: "Harmlose Daten" zwischen Client und Eingangsknoten (Bridge)
- Beispiel obfs4:
  - Genereller Ansatz:
    - Modifiziertes TLS-Protokoll zwischen Client und Bridge
    - Voraussetzung: obfs4-fähige Bridge
- Weitere Beispiele: meek, FTE, ScrambleSuit
- Integration in TBB:
  - Während Verbindungskonfigurationsphase

Umfang

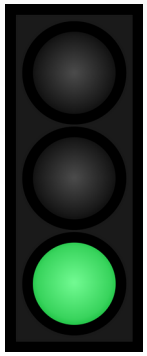
- Analyse der tor-server Code-Basis:
  - Im Hinblick auf Wartbarkeit (ISO 9126)
  - Identifikation der generellen Angriffsoberfläche
  - Weitere Resultate aufgrund von zusätzlicher Analyse
  - Ausblick: obfs4proxy / torsocks

Bewertung

- Werkzeuge:
  - SonarQube (inkl. Community Plugins)
  - RATS (Rough Auditing Tool for Security)
  - Cppcheck
  - Gesunder Menschenverstand + 30 Jahre SDLC-Erfahrung
  - Andere Formen schwarzer Magie ☺
- Code-Basis:
  - Tor 0.4.3.5 (Scope: C CB)

Summary

- Gesamtqualität:



- Keine Sicherheitsrisiken im analysierten Code

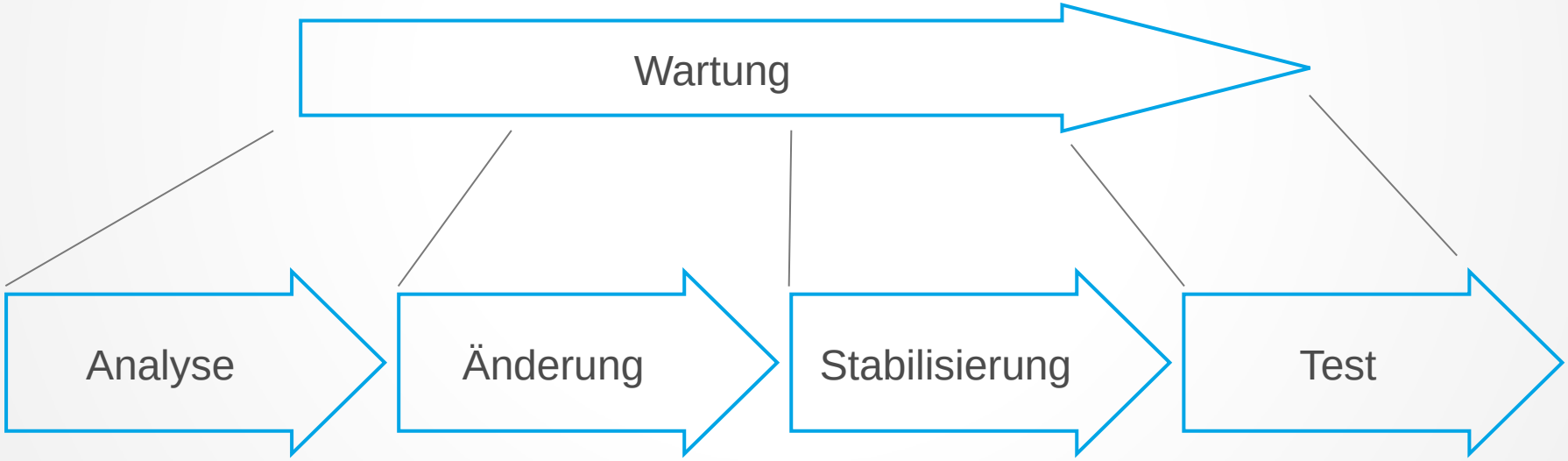
ISO 9126

- Internatonaler Standard für S/W-Qualitätsanalyse

Wartbarkeitsattribute

Wartbarkeit =

- *Analysierbakeit*: Wie und wo zu ändern?
- *Änderbarkeit*: Wie einfach ist eine Änderung?
- *Stabilität*: Code-Kohärenz wärend der Änderung?
- *Testbarkeit*: Validierung der Änderung?



Vereinfachtes Analyse-Model

	Volume	Duplizierung	Einheit-Komplexität
Analysierbarkeit	X	X	
Änderbarkeit		X	X
Stabilität			
Testabarkeit			X

Volumen

- Software Produktivität:
  - xLOC
  - Function points (FPs)
  - …
- Herausforderung:
  - Ausdruckskraft von unterschiedlichen Programmiersprachen
  - Ansatz: Gewichtung xLOC mit Standard-Produktivitätsfaktor
  - ➢ Programming Languages Table

Volumen (ff.)

- Programming Languages Table:

Sprache	Level	Durchschn. # LOC pro FP
Perl	15	21
Smalltalk/V	15	21
Objective C	12	27
Haskell	8.5	38
C++	6	53
Basic	3	107
C	2.5	128
Macro assembler	1.5	213

Volumen (ff.)

- Warum ist das wichtig:
  - Gesamtkosten
  - Aufwand für Neuerstellung
  - tor Volumenmetriken:

Einheit	#
Gesamt-LOCs	250.004
Dateien	1.020
Funktionen	9.545
Klassen	n/a

Duplizierung

- Code-Duplizierung reduziert Wartbarkeit
  - Hohe Wartungskosten
  - Fehlerbehebung
  - Reduzierte Testbarkeit

Duplizierung (ff.)

- tor Duplizierungsmetriken:

Einheit	Duplizierung
Gesamt	1 %
Blöcke	280
Dateien	46

Einheits-Komplexität

- Wird gemessen mittels McCabe’s zyklischer Komplexität:
  - Anzahl von Entscheidungspunkten (decision points / DPs) pro Einheit (Methode / Funktion / Datei)
  - McCabe, IEEE Transactions on Software Engineering, 1976
  - Höhere Komplexität bedingt höherer Aufwand bei Änderungen und Test
  - Für C/C++/Objective C, Erhöhung von DPs für:

Funktionsdefinitionen, while, do while, for, throw statements, return (Ausnahme: letzte Anweisung einer Funktion), switch, case, default, &&, ||, ?, catch, break, continue, goto

Einheits-Komplexität (ff.)

- Übersicht:

Zyklische Komplexität	Risiko-Einschätzung
1 - 10	Klarer Code, geringes Risiko
11 - 20	Komplex, mittleres Risiko
21 - 50	Sehr komplex, hohes Risiko
> 50	Nicht verständlich, sehr hohes Risiko

Einheits-Komplexität (ff.)

- tor Komplexitätsmetrik:

Einheit	Komplexität
Anteil Funktion (zK > 20)	5.2 %
Klasse	n/a

Zusammenfassung

- Ergebnis der Code-Analyse: sehr gut
  - SQALE Rating: A
  - Geschätzte technische Schuld: 2 Stunden
  - Wenige Sicherheitsprobleme:

Einheit	Vorkommen
Schwachstelle	0
Kleinere Probleme	0
Smells	1

(SQALE: Software Quality Assessment based on Lifecycle Expectations)

Angriffsoberflächenanalyse

- Ergebnis:
  - Kein Refactoring nötig
  - Angriffsoberflächenanalyse:
    - Keine nennenswerten Vorkommnisse

Angriffsoberflächenanalyse (ff.)

- Angriffsoberflächenanalyse (ff.):
  - C-Code: 1 Smell
  - Python (jenseits Scope): Coding Standards
- Abhilfe:
  - Erweiterte Code-Review

Zusammenfassung

- Solide Code-Basis trotz Alter
- Kleine Angriffsoberfläche: kein generelles Refactoring notwendig
- Kleinere Probleme können ohne großen Aufwand behoben werden

➢ Stabile Code-Basis mit minimaler Angriffsoberfläche

Ausblick

- obfs4proxy:
  - Ähnlich sichere CB (19 Smells)
  - Etwas größere Angriffsoberfläche durch erhöhe Komplexität
  - Metriken: 14 kLOC, durchschn. Komplexität: 15
- torsocks:
  - Sicherste analysierte CB
  - Keine Smells / Vulnerabilities
  - Metriken: 15 kLOC, durchschn. Komplexität: 11

Quellen

- Tor source code: gitweb.torproject.org/tor.git
- obfs4: github.com/Yawning/obfs4
- torsocks: github.com/dgoulet/torsocks
- Sonarqube: www.sonarqube.org/downloads
- Cppcheck: cppcheck.sourceforge.net
- RATS: code.google.com/archive/p/rough-auditing-tool-for-security/downloads

Diskussion / Fragen

Vielen Dank!

Benni Lason

Dr. Christoph Zimmermann

benlason at <ignore>space</ignore>disroot<dot></dot>org

monochromec at <ignore>space</ignore>gmail<dot></dot>com