

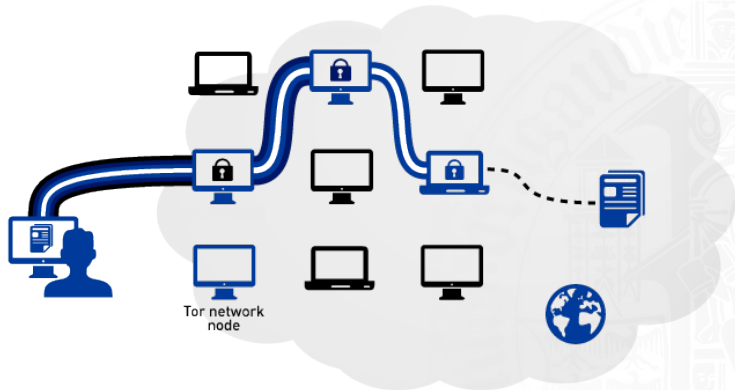
Per-Circuit TCP-over-IPsec Transport for Anonymous Communication Overlay Networks

Manuel Schneider

Albert Ludwigs Universität - Institut für Informatik

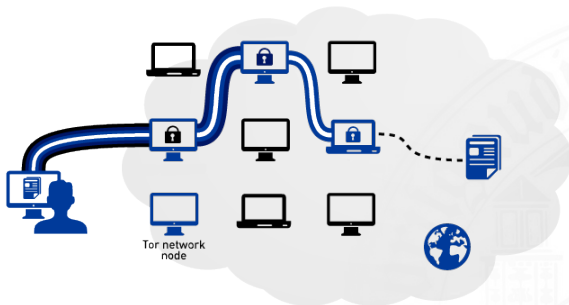
23. Januar 2015

TOR - Architektur/Terminologie



Circuits

TOR



- Werden Hop-by-hop aufgebaut
 - Mit jedem Knoten wird ein Schlüssel ausgehandelt
- Üblicherweise drei: Entry, Middle-, Exit Node
- Werden nach einem Zeitintervall oder Fehler gewechselt
- Werden auf Vorrat angelegt (Performance)

Circuit construction

TOR

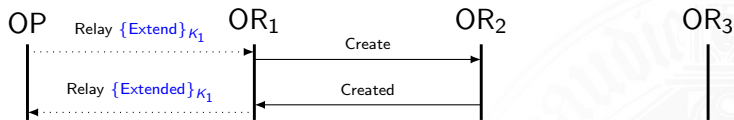


Circuit Erweiterung

- OP sendet **Create Control** Cell an OR₁
- OR₁ bestätigt mit **Created Control** Cell
- Wenn schon eine Verbindung besteht wird diese genutzt
- Schlüsselaustausch: OP und OR₁ teilen Schlüssel K₁
- Beide merken sich Circuit Identifier

Circuit construction

TOR

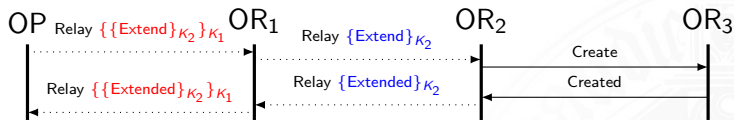


Relay Erweiterung

- OP sendet **Extend Relay** Cell an OR₁
- OR₁ erweitert den Circuit zu OR₂ (**Create**)
- OR₁ leitet die Antwort in einer **Extended Relay** Cell an OP zurück
- OP und OR₂ teilen einen gemeinsamen Schlüssel K₂

Circuit construction

TOR

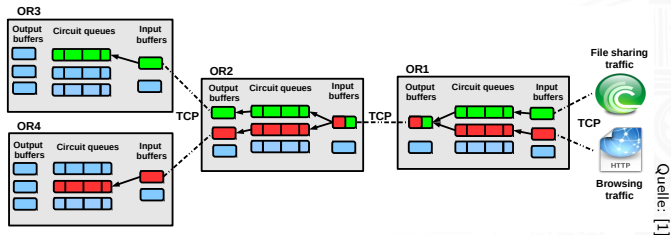


Relay Erweiterung

- OP sendet **Extend Relay** Cell an OR₁
- OR₁ erweitert den Circuit zu OR₂ (**Create**)
- OR₁ leitet die Antwort in einer **Extended Relay** Cell an OP zurück
- OP und OR₂ teilen einen gemeinsamen Schlüssel K_2

Cross Circuit Interference Problem

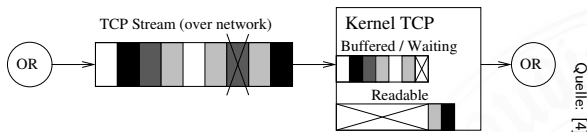
TOR



- Tritt auf wenn sich verschiedene Circuits eine TCP Verbindung teilen
- Circuits teilen selben Buffer / TCP Mechanismus
 - In order delivery
 - Congestion control

Cross Circuit Interference Problem

TOR



Head-On-Line Blocking

- Folge üblicher TCP Arbeitsweise
- TCP garantiert eine fehlerfreie und Ordnung erhaltende Datenübertragung
- Durch eine verlorene Cell werden bereits empfangene Nachfolgecells vom Kernel zurückgehalten
 - Sind diese Cells von einem anderen Circuit, äußert sich diese Zeit als Verzögerung

Kernel-mode per-circuit TCP

PCTCP

Problem:

- Gemultiplexte Circuits unterliegen alle den selben TCP Mechanismen

Lösung in PCTCP:

- Jeder Circuit bekommt eine eigene TCP Verbindung (per-circuit TCP)

Recap - Circuit Construction Vanilla Tor

PCTCP



Circuit Erweiterung

- OP sendet **Create Control** Cell an OR1
- OR1 bestätigt mit **Created Control** Cell
- **Wenn schon eine Verbindung besteht wird diese genutzt**
- Schlüsselaustausch: OP und OR1 teilen Schlüssel
- Beide merken sich Circuit Identifier

Recap - Circuit Construction PCTCP

PCTCP



Circuit Erweiterung

- OP sendet **Create Control** Cell an OR1
- OR1 bestätigt mit **Created Control** Cell
- **Es wird in jedem Fall eine neue Verbindung geöffnet**
- Schlüsselaustausch: OP und OR1 teilen Schlüssel
- Beide merken sich Circuit Identifier

IPSec in PCTCP

PCTCP

Warum IPSec?

- PCTCP: eine TCP Verbindung pro Circuit
- TLS verschlüsselt lediglich die Nutzlast/
TCP-Header bleibt Klartext
 - Ein Angreifer kann Verbindungen unterscheiden
 - Einschränkung der Anonymität

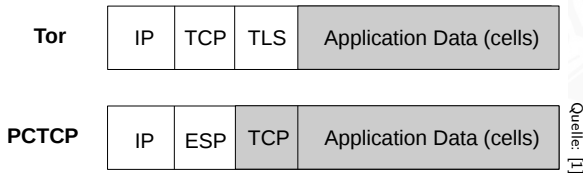
Exkurs - IPSec (Internet Protocol Security)

PCTCP

- Arbeitet auf Vermittlungsschicht
- Stellt Schutzziele sicher
 - Vertraulichkeit
 - Authentizität
 - Integrität
- Umfasst Protokolle für eine gesicherte Kommunikation
 - Authentication Header (AH)
 - Encapsulating Security Payload (ESP)
- Je zwei Betriebsmodi
 - Transport Modus
 - Tunnel Modus

IPSec in PCTCP

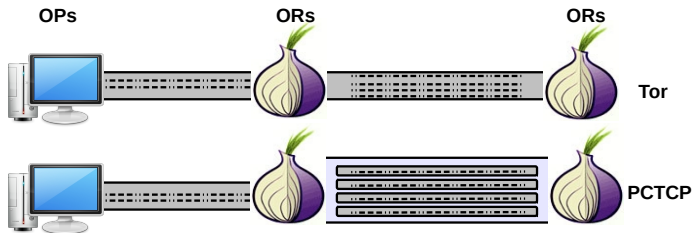
PCTCP



- Encapsulating Security Payload (ESP) im Transport Mode sorgt für die Vertraulichkeit der Verbindungsdaten
- Mit IPSec wird diese Angriffsfläche genommen
- TLS ist redundant und wird ausgelassen

PCTCP Überblick

PCTCP



Quelle: [1]

- “Kernel-mode per-circuit TCP” löst das Cross Circuit Interference Problem
- Anstatt TLS sichert IPsec die Schutzziele und sorgt zusätzlich für die Vertraulichkeit der Verbindungsdaten

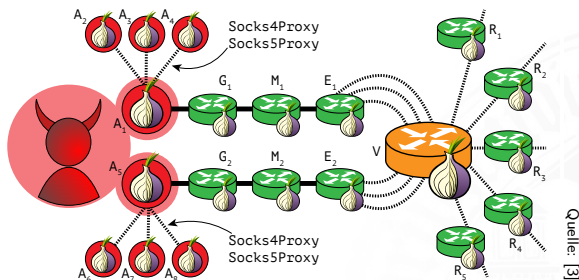
Socket Exhaustion Attacks

Sicherheit

- Sockets sind eine endliche Ressource
- Ziel ist den Onion Router außer Kraft zu setzen
 - Alle verfügbaren Sockets belegen
 - So viele Sockets öffnen bis das System überlastet wird
- Bisher schwierig, wegen Multiplexing der Circuits über eine einzige TCP Verbindung

Socket Exhaustion Attacks

Sicherheit



Sniper Attack

- Angriff auf jede Node möglich
- Nutzt Tor selbst für den Angriff / Angreifer bleibt anonym
- Funktioniert auch ohne PCTCP, aber teuer (1 OP/Socket)
- Mit PCTCP mit einem Onion Proxy möglich

Socket Exhaustion Attacks

Sicherheit

PCTCP Socket Exhaustion

- Basiert auf der Circuit Erweiterung
- Circuits werden über Zielknoten aufgebaut
- Zielknoten soll Middle Node sein
- Angriever bleibt anonym
- Günstig, da nicht mehr ein Onion Proxy pro Circuit gebraucht wird

Fazit

- Einsatz erfordert manuelle Interaktion (IPsec)
 - Hemmschwelle für Teilnahme steigt
 - Gesamtanonymität sinkt
- Alternativ: IPsec im Userspace
 - Lizenzkompatible Implementierung existiert nicht, Eignung vergleichbar mit TCP-over-DLTS
- Sicherheitsprobleme sind nicht auf PCTCP bezogen
 - Kern des Problems liegt in der "Ein Circuit Pro TCP"-Verbindung

Reference

Vielen Dank für die Aufmerksamkeit!



M. AlSabah and I. Goldberg.

PCTCP: Per-Circuit TCP-over-IPSec Transport for Anonymous Communication Overlay Networks.
Technical report, Centre for Applied Cryptographic Research (CACR) at the University of Waterloo,
September 2013.



Roger Dingledine, Nick Mathewson, and Paul Syverson.

Tor: The second-generation onion router.

In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13, SSYM'04*. USENIX
Association, 2004.



John Geddes, Rob Jansen, and Nicholas Hopper.

IMUX: Managing Tor Connections from Two to Infinity, and Beyond.

Technical report, Centre for Applied Cryptographic Research (CACR) at the University of Waterloo,
November 2014.



J. Reardon and I. Goldberg.

Improving Tor Using a TCP-over-DTLS Tunnel.

Technical report, Centre for Applied Cryptographic Research (CACR) at the University of Waterloo,
September 2009.



R. Ringledine and S. Murdoch.

Performance Improvements on Tor or, Why Tor is slow and what we're going to do about it.
März 2009.