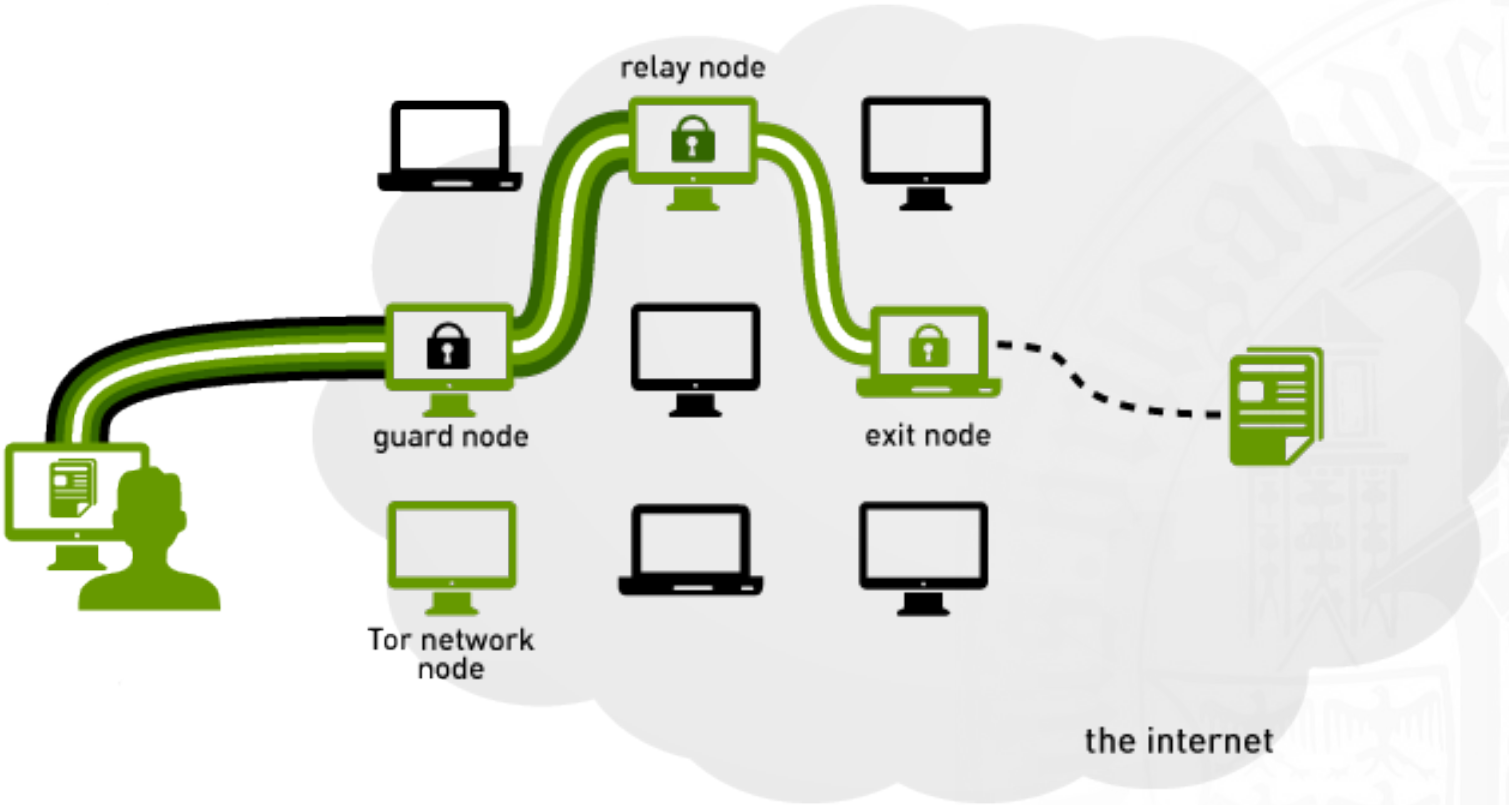
Per-Circuit TCP-over-IPsec Transport for Anonymous Communication Overlay Networks

Manuel Schneider

Albert Ludwigs Universität - Institut für Informatik

24. Januar 2015

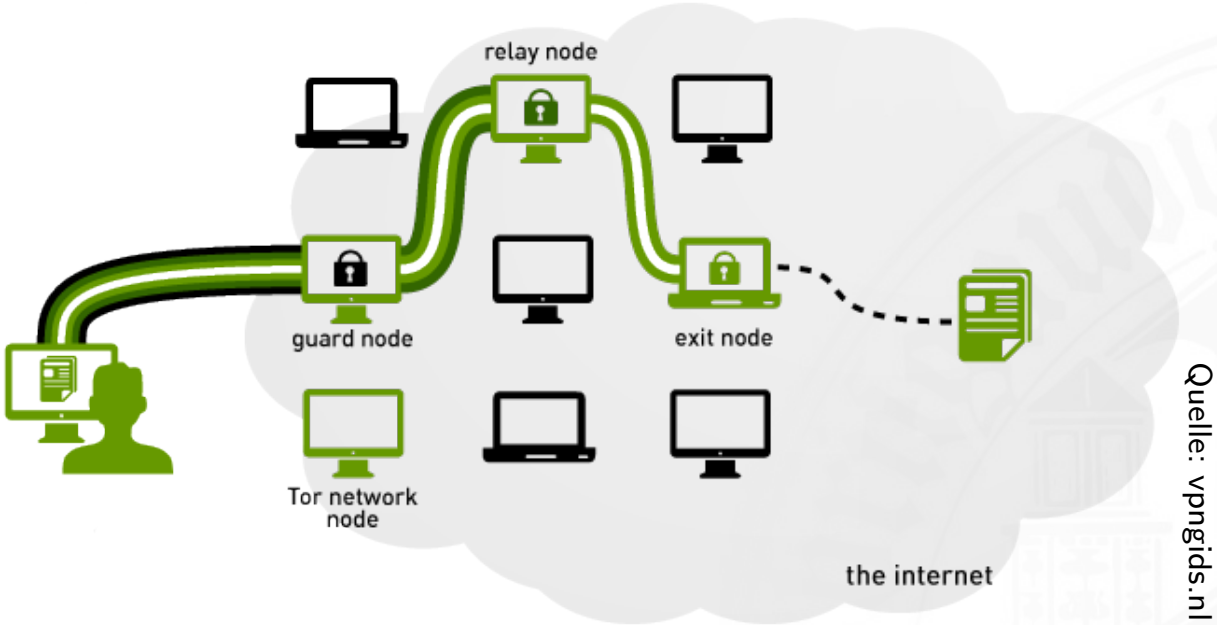
Tor - Architektur/Terminologie



Quelle: vpngids.nl

Circuits

TOR



Quelle: vpngids.nl

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

Circuit construction

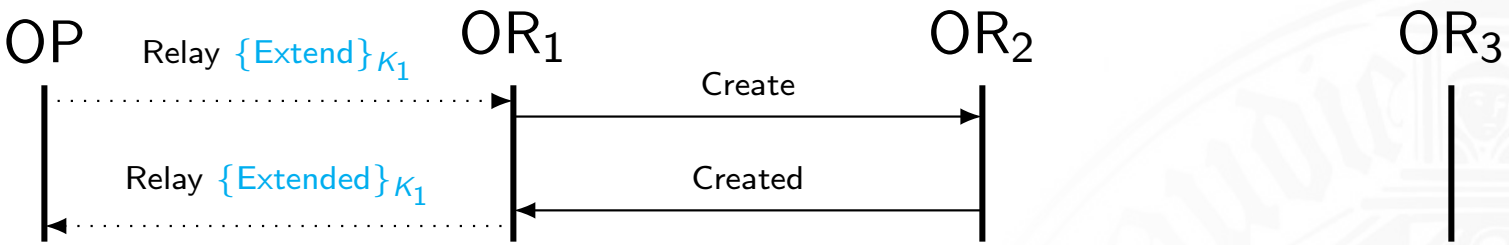
TOR

Circuit Erweiterung

- OP öffnet eine Verbindung zu OR_1
  - Wenn schon eine Verbindung besteht wird diese genutzt
- OP sendet Create Control Cell an OR_1
- OR_1 bestätigt mit Created Control Cell
- Schlüsselaustausch: OP und OR_1 teilen Schlüssel K_1

Circuit construction

TOR

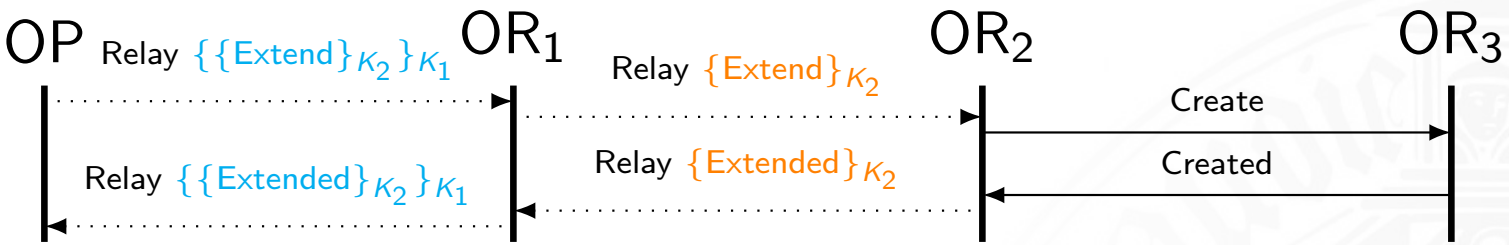


Relay Erweiterung

- OP sendet Extend Relay Cell an OR_1
- OR_1 öffnet eine Verbindung zu OR_2
  - Wenn schon eine Verbindung besteht wird diese genutzt
- OR_1 erweitert den Circuit zu OR_2 (Create)
- OR_1 leitet die Antwort in einer Extended Relay Cell an OP zurück
- OP und OR_2 teilen einen gemeinsamen Schlüssel K_2

Circuit construction

TOR

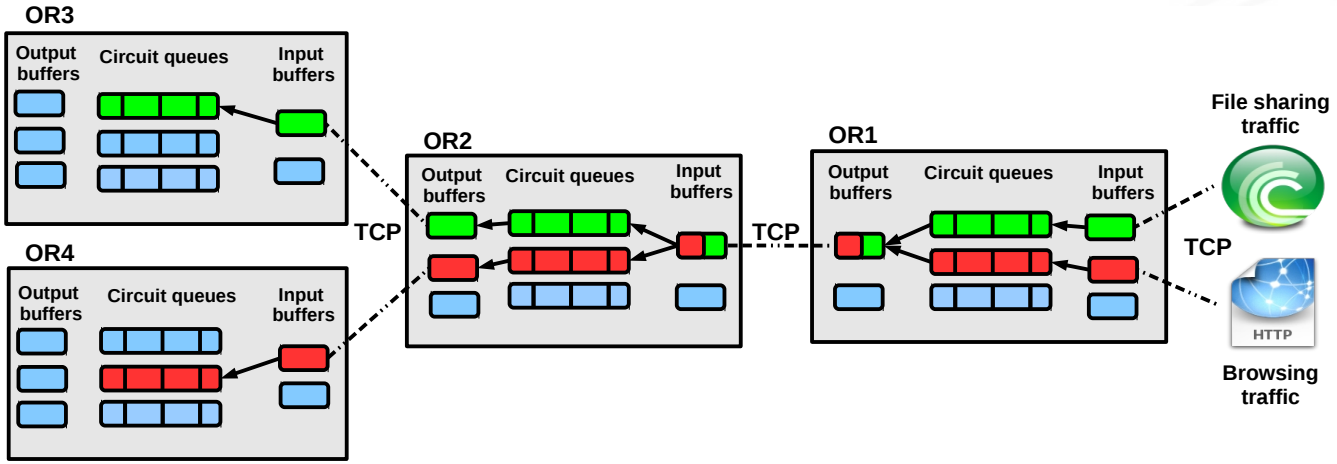


Relay Erweiterung

- OP sendet Extend Relay Cell an OR_2
- OR_2 öffnet eine Verbindung zu OR_3
  - Wenn schon eine Verbindung besteht wird diese genutzt
- OR_2 erweitert den Circuit zu OR_3 (Create)
- OR_2 leitet die Antwort in einer Extended Relay Cell an OP zurück
- OP und OR_3 teilen einen gemeinsamen Schlüssel K_3

Cross Circuit Interference Problem

TOR

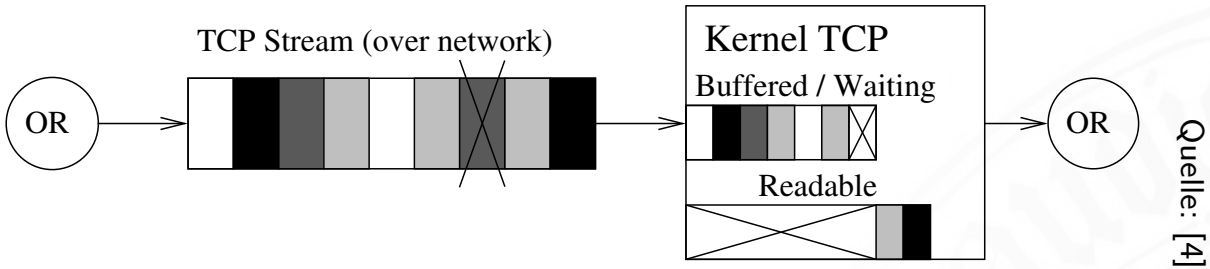


Quelle: [1]

- 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



Quelle: [4]

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
  - $\rightarrow$ Sind diese Cells von einem anderen Circuit, äußert sich diese Zeit als Verzögerung

Kernel-mode per-circuit TCP

PCTCP

Problem:

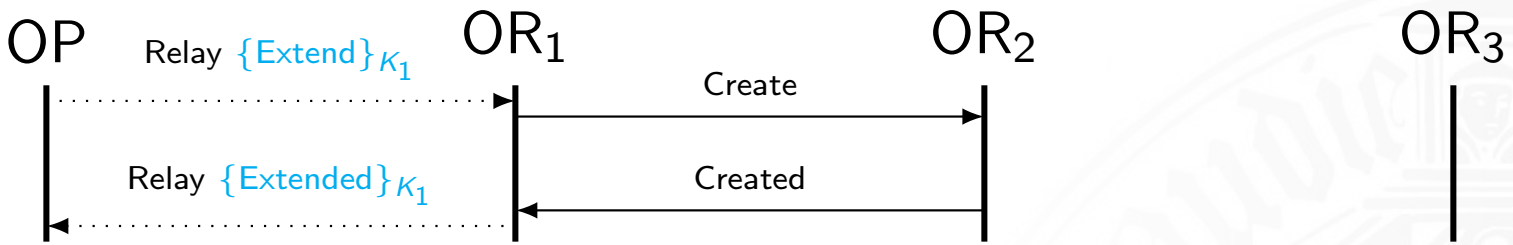
- Gemultiplexte Circuits unterliegen alle den selben TCP Mechanismen

Lösung in PCTCP:

- Jeder Circuit bekommt eine eigene TCP Verbindung (Ausnahme OP zu OR)

Recap - Circuit Construction PCTCP

PCTCP

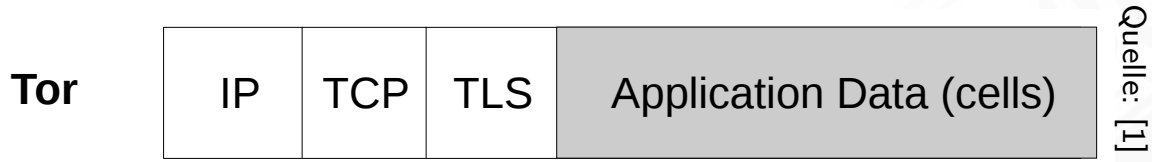


Relay Erweiterung

- OP sendet Extend Relay Cell an OR_1
- OR_1 öffnet eine Verbindung zu OR_2
  - ~~Wenn schon eine Verbindung besteht wird diese genutzt~~
- OR_1 erweitert den Circuit zu OR_2 (Create)
- OR_1 leitet die Antwort in einer Extended Relay Cell an OP zurück
- OP und OR_2 teilen einen gemeinsamen Schlüssel K_2

IPSec in PCTCP

PCTCP



Quelle: [1]

Warum IPSec?

- TLS verschlüsselt lediglich die Nutzlast/ TCP-Header bleibt Klartext
  - $\rightarrow$ Ein Angreifer kann Verbindungen unterscheiden
  - $\rightarrow$ Einschränkung der Anonymität

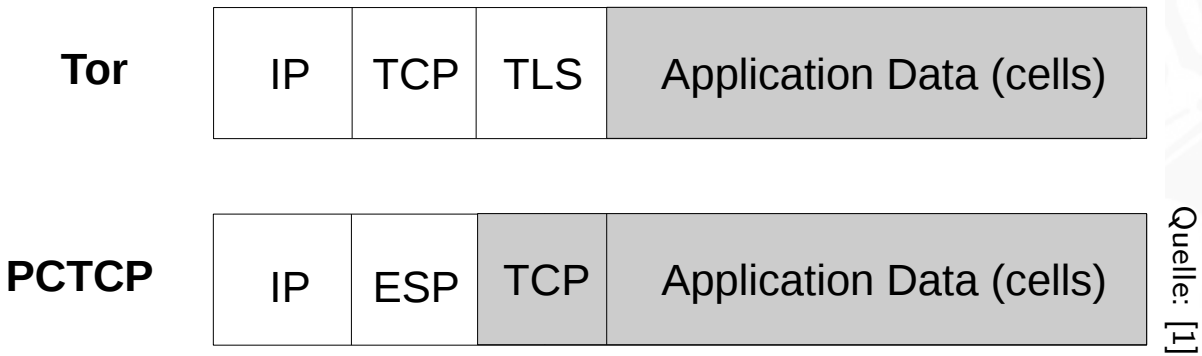
Exkurs - IPSec (Internet Protocol Security)

PCTCP

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

IPSec in PCTCP

PCTCP

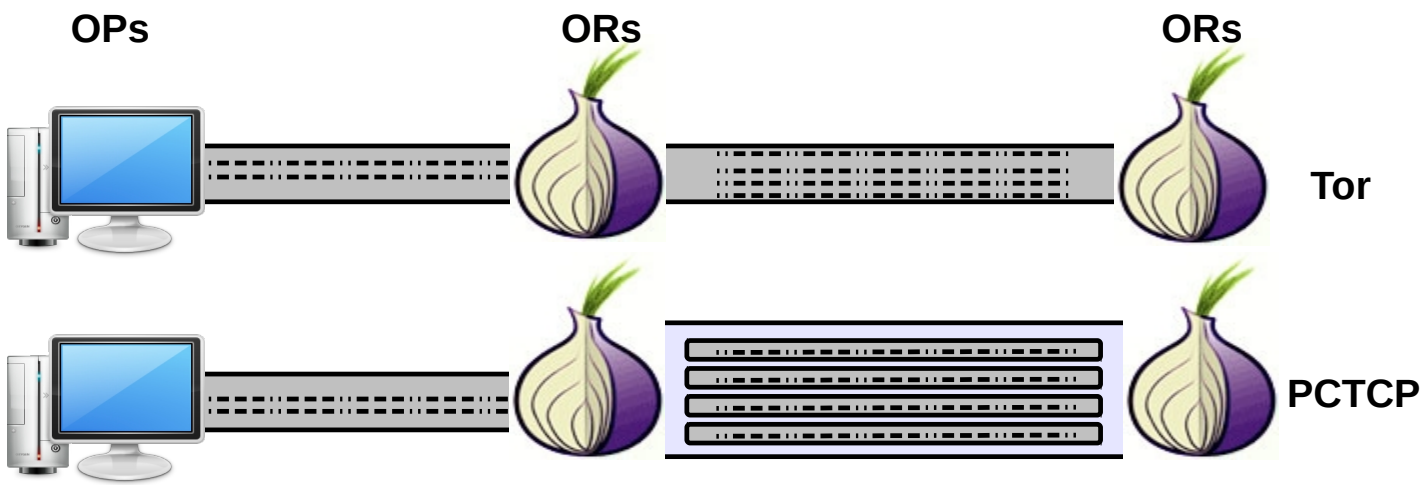


Quelle: [1]

- 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

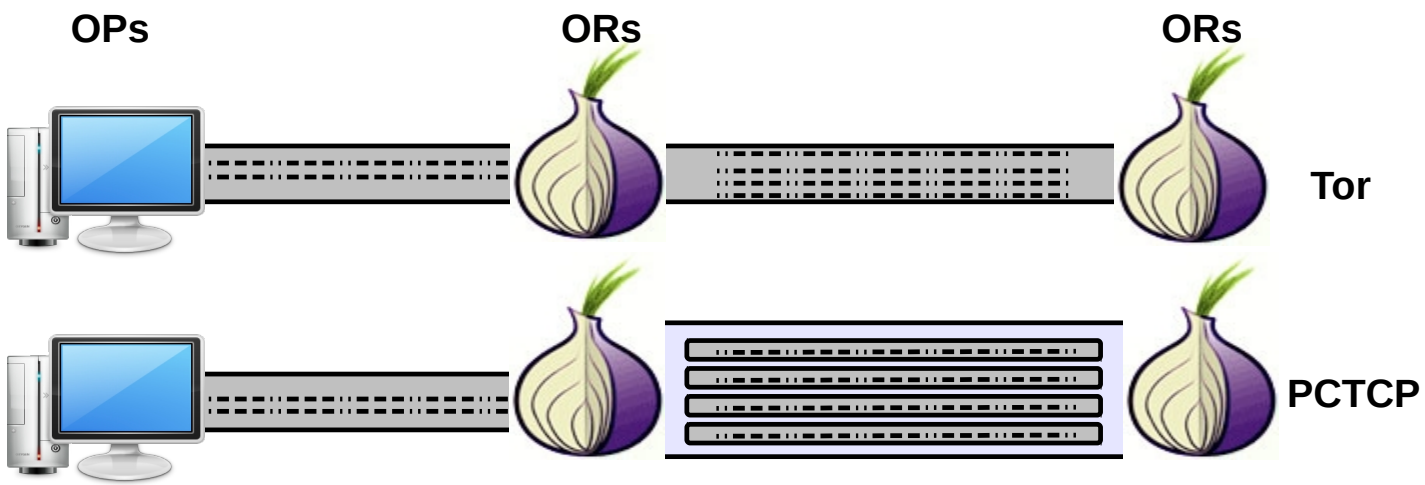
PCTCP



Quelle: [1]

PCTCP Überblick

PCTCP



Quelle: [1]

- Bis zu 60% verringerte Latenzzeiten
- Bis zu 30% verbesserter Durchsatz

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 öffen bis das System überlastet wird
- Bisher schwierig, wegen Multiplexing der Circuits über eine einzige TCP Verbindung

Socket Exhaustion Attacks

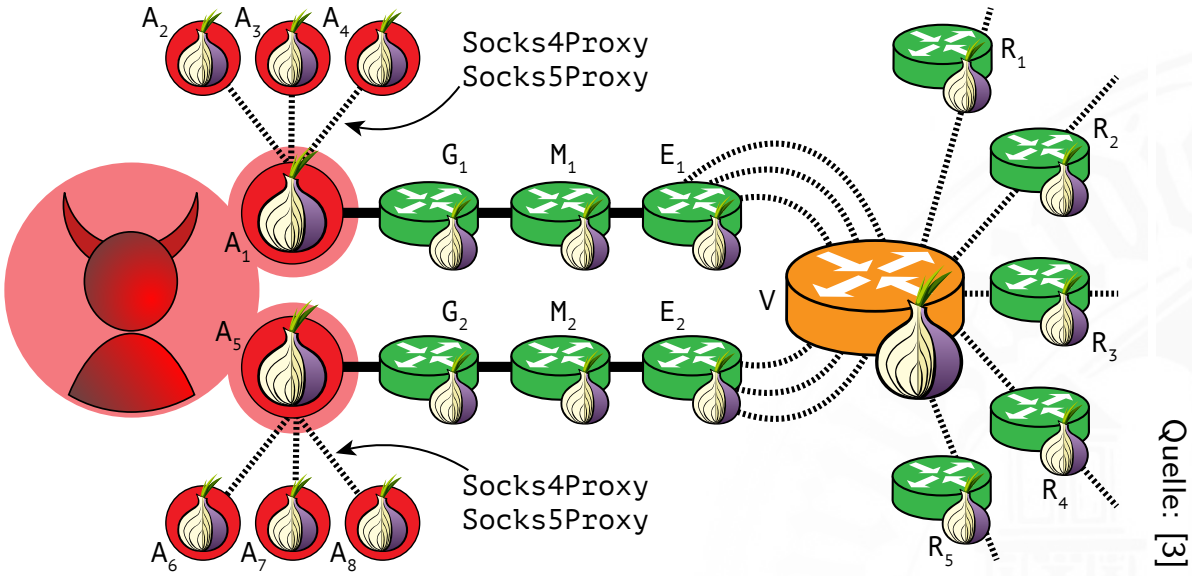
Sicherheit

Angriff auf Exit Nodes

- Jeder Stream öffnet einen Socket an der Exit Node
- Angreifer öffnet Streams über selbe Exit Node,
  bis diese weitere Streams ablehnt
- Funktioniert auch ohne PCTCP
- Angriff erreicht nicht das komplette Netzwerk

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
- Angreifer bleibt anonym
- Günstig, da nur ein Onion Proxy gebraucht wird

Fazit

- Einsatz erfordert manuelle Interaktion (IPsec)
  - $\rightarrow$ Hemmschwelle für Teilnahme steigt
  - $\rightarrow$ 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

Referenzen

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.