Connection Signaling

Mark Handley

UCL
The Problems

- Mobility
- Multihoming
- Firewalls
- NATs
- Address Spoofing
- DoS Attacks
Departures from End-to-End

- Mobility
  - Need to find host, need to re-bind connections.
- Multihoming
  - Need to bind connection to more than one path without affecting global routing.
- Firewalls
  - Middle cares about connections.
- NATs
  - Middle cares about connections, rewrites addresses
- Address Spoofing
  - Prevention involves the middle, detection involves the middle.
- DoS Attacks
  - End can’t defend itself - needs to involve the middle.
Connections

- Perhaps the traditional self-contained TCP model of a session connecting a pair of IP addresses and ports needs revision?
WARNING!

LESS-THAN HALF BAKED IDEAS COMING UP.

IGNORES RELATED WORK.

MAY TREAD ON OTHER PEOPLE’S TURF.

CONTENTS MAY BE HOT.
Philosophy and Assumptions.

- IP Addresses are primarily *addresses*.
  - Identify a location in the network.
  - Should be possible to aggregate routes.

- Transport protocols should be capable of supporting *address and port rebinding*.
  - Before/during connection establishment.
  - In mid connection.

Plenty of work on this - definitely feasible for TCP, SCTP, DCCP. Feasible for UDP *flows*. 
Strawman:
Connection Signaling Protocol (CSP)

- Assume that we use a general purpose Connection Signaling Protocol to signal every transport connection.
  - Intent is not to build virtual circuits.
  - Provide a clean place in the architecture to:
    - Signal the application’s intent to middleboxes.
    - Signal the middleboxes intent to end hosts.
    - Locate mobile end-systems and signal mobility to everyone.
    - Signal alternative path information to end-systems.
    - Handshake between end-systems before trusting them.
    - Signal middleboxes to deny service.
CSP is not strictly layered under or over transport protocols.

- More like alongside.
- Akin to how ICMP is to IP.
- May be able to piggyback first data packet on signaling.
  - Will ignore optimizations for now.
Simple Firewalled UDP Connection

Setup(UDP, A, p_1 ↔ B, p_2)

OK(Timeout=10s)

Change(Timeout = 300s)

Detach

CSP aware Firewall

UDP Connection

OK

OK
Firewalled Incoming UDP Connection

Setup(UDP, B,p₁→A,p₂)

CSP aware Firewall

Unidirectional UDP Connection

Detach

OK

OK
Firewall redirect to offpath proxy

Setup(A,p₁ ↔ B,80)

Redirect(A,p₁ ↔ P:B,80)

Setup(A,p₁ ↔ P:B,80)

Setup(P,p₁ ↔ B,80)

HTTP Connection

CSP aware Firewall

Proxy
Firewall rejection

Setup(A,p₁ ↔ B,p₂)

CSP aware Firewall

Reject(explanation)
NAT Traversal (1)

Setup(UDP, A, p₁ ↔ B, p₂)

Redirect(A, p₁:N, p₃ ↔ B, p₂)

Setup(A:N, p₃ ↔ B, p₂)

UDP(A, p₃ ↔ B, p₂)

OK(A:N, p₃ ↔ B, p₂)

UDP(N, p₃ ↔ B, p₂)
NAT Traversal (2)

CSP-aware NAT

Setup(UDP, A:N, p₁ ↔ B, p₂)

OK(UDP, A:N, p₁ ↔ B, p₂)

UDP(A, p₃ ↔ B, p₂)

Setup(UDP, N, N coffee, p₁ ↔ B, p₂, N = www.example.com)

OK(UDP, A:N, p₁ ↔ B, p₂)

UDP(N, p₃ ↔ B, p₂)

Note: requires change to sockets API and app support on B
Mobile Client

Setup($A, p_1 \leftrightarrow S, p_2$) + Nonce + Sig

Data Transfer ($A, p_1 \leftrightarrow S, p_2$)

Detach($A, p_1 \leftrightarrow S, p_2$),
Attach($B, p_1 \leftrightarrow S, p_2$) + Nonce + Sig

Data Transfer ($B, p_1 \leftrightarrow S, p_2$)

Moves
Mobile Server

Setup(A, p₁ ↔ S, p₂)

Redirect(A, p₁ ↔ S:B, p₂)

Register(S at B)

OK

OK+nonce+sig

Data Transfer (A, p₁ ↔ B, p₂)

Attach(A, p₁ ↔ S:C, p₂),
Detach(A, p₁ ↔ S:B, p₂)+Nonce+Sig

OK

Data Transfer (A, p₁ ↔ C, p₂)

At B

At C
Hidden Mobile Server

Setup($A,p_1 \leftrightarrow S,p_2$)

Redirect($A,p_1 \leftrightarrow S:B,p_2$)+nonce+sig

Setup($A,p_1 \leftrightarrow S:B,p_2$)

Data Transfer ($A,p_1 \leftrightarrow B,p_2$)

OK

Data Transfer ($A,p_1 \leftrightarrow C,p_2$)

OK

Register($S$ at $B$)

At $B$

At $C$

Detach+ Attach+Nonce+Sig

moves
Offpath Firewall for Mobile Host

A

Setup(A,p₁ ↔ S,p₂)
+ auth token

Redirect(A,p₁:p₃ ↔ S:B,p₂:p₄)

Setup(A,p₃ ↔ S:B,p₂:p₄)

Data Transfer (A,p₃ ↔ B,p₄)

F/W Firewall at home site

OK

At B

Binds port p₄ for service from A,p₃

Setup(A,p₁:p₃ ↔ S:B,p₂:p₄)
+ private auth

OK
Simple Multihoming

Setup(C, p₁ ↔ Sₓ, p₂,)

OK, Attach(C, p₁ ↔ Sᵧ, p₂,)

Data

Sₓ, Sᵧ
Simple Multihoming

Data

Change($C, p_1 \leftrightarrow S_x, p_2$, low pref)

S_x, S_y
Spoofing

Setup($A, p_1 \leftrightarrow S, p_2$)

Ack, send nonce

Setup($A, p_1 \leftrightarrow S, p_2$) + nonce echo

Setup($A, p_1 \leftrightarrow S, p_2$)

OK

Data Transfer ($A, p_1 \leftrightarrow B, p_2$)
DoS Prevention

Setup(A, p₁ ↔ S, p₂)

Reject(A, * via G ↔ S, *)

Data(A → S)

CSP aware Gateway

Setup(A, p₁ via G ↔ S, p₂)

Ack, send nonce

Reject(A, * via G ↔ S, *) + nonce
Connection Signaling: Summary

**Assertion:**
- Many of the architectural problems we currently face can be solved using connection signaling.
- Lots of questions.
  - Efficiency, simplicity vs flexibility
  - Backward compatibility, existing NATs, related work.
  - Which problems to focus on, which to ignore?
- Real danger of second system syndrome.
  - Unless it’s simple, no chance of success.