tcpcrypt

Stanford University and University College London

November 4, 2015
What’s new?

• Integrated with ENO.

• Simplified spec - cut it in half (25 pages). No more RSA, no more SYNCOOKIE TCP option, basic TLV (no more keep-alive sync-req & other app-layer messages).

• Updated Windows, OSX and Linux code.
Goals

• Simple: what’s the simplest change needed to TCP to add encryption?

• Self-contained: no dependencies, be amenable to implementations in kernels and embedded systems.

• Minimal: tailored for the task at hand (opportunistic encryption) with no unnecessary crypto.
Overview

- Use ENO to negotiate key exchange mechanism.
- Use first two TCP data segments to exchange keys.
- Wrap application data in a basic Type-Length-Value (TLV) record and apply authenticated encryption on it.
Handshake

SYN - ENO(tcpcrypt-P256, tcpcrypt-P512)

SYN ACK - ENO(tcpcrypt-P256)

ACK - ENO

INIT1 (nonce, DH param, sym-cipher list)

ACK

INIT2 (nonce, DH param, sym-cipher)
Key scheduling

eno-transcript, init1, init2, DH-output

HKDF-Extract

PRK

CPRF

SS[0]

Session ID
MK[0]

AEAD A->B
AEAD B->A

Rekey

MK[1]

AEAD A->B
AEAD B->A

Session ID
MK[0]

SS[1]

next key (resume session)

AEAD A->B
AEAD B->A
Session resumption

SYN - ENO(tcpcrypt-resume-ID-ABCDE)

SYN ACK - ENO(tcpcrypt-resume)

ACK - ENO

(user data)
Payload protection

Authenticated

| control (rekey bit) | length | flags URG, FIN | urgent pointer (only if URG set) | data | TAG |

Encrypted

AEAD nonce is byte offset in underlying TCP stream
Divert sockets - kernel sends packets to process for modification.

- Pro: can modify 3-way handshake.
- Con: hard to add TLV to TCP stream (need to map sequence and ack numbers).

Redirect (transparent proxy) - kernel redirects connection (stream) to process.

- Pro: easy to inject TLV and modify payload.
- Con: cannot modify handshake.
- Con: don’t know destination until connection is accepted. Destination may not be listening and so we’ll accept() and close() socket instead of connection being refused - different semantics / behavior.
### OS support

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>OSX</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Divert</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Redirect</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Current tcpcrypt implementation supports all these combinations.