Authentication for TCP-based Routing and Management Protocols

draft-bonica-tcp-auth-04
Motivation

• Many operators do not authenticate TCP based routing protocols
  – BGP, LDP

• Current BCP (RFC 2385) does not fulfill operator requirement
Concerns Regarding RFC 2385

• CPU utilization
  – Not addressed in the current memo

• Key management
  – Keys need to be refreshed periodically
  – Key refresh (typically) requires session reset

• Weak cryptography
  – There are many well-know attacks on MD5
Approach

• Better TCP-layer authentication
  – Enhanced TCP Authentication Option

• Hitless key rollover
  – Key chains configured on peer systems
  – Key Identifiers

• Stronger cryptography
  – CMAC-AES-128-96
  – HMAC-SHA-1-89
Alternative Approaches

• TLS
  – Does not protect TCP session, itself

• IPSec
  – Perception of operational complexity
  – Coordination issues for pre-shared key rollover
  – Protection of PKI certificates
  – Otherwise, a feasible approach
## Enhanced Authentication Option

<table>
<thead>
<tr>
<th>Kind</th>
<th>Length</th>
<th>T</th>
<th>K</th>
<th>Alg ID</th>
<th>Res</th>
<th>Key ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+---------------------------------+
| Kind | Length | T|K| Alg ID | Res| Key ID |
+---------------------------------+
|      |        |    |    |        |     |        |
|      |        |    |    |        |     |        |
|      |        |    |    |        |     |        |
|      |        |    |    |        |     |        |

---

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+---------------------------------+
| Kind | Length | T|K| Alg ID | Res| Key ID |
+---------------------------------+
|      |        |    |    |        |     |        |
|      |        |    |    |        |     |        |
|      |        |    |    |        |     |        |
|      |        |    |    |        |     |        |

---

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+---------------------------------+
| Kind | Length | T|K| Alg ID | Res| Key ID |
+---------------------------------+
|      |        |    |    |        |     |        |
|      |        |    |    |        |     |        |
|      |        |    |    |        |     |        |
|      |        |    |    |        |     |        |

---
Key Chain

• Contains up to 64 keys
• Each key contains
  – Identifier [0..63]
  – Authentication Algorithm
  – Shared secret
  – Vector [in|out|both]
  – Start and end time for sending
  – Start and end time for receiving
Sending System Procedure

• Identify active key candidates
  – vector == out || vector == both
  – Start-time for sending <= system-time
  – End-time for sending > system time

• If there are no candidates, log event and discard outbound packet

• If there are multiple candidates, select key with most recent start-time for sending
Sending System Procedure (continued)

• Calculate MAC using active key
  – Calculate over TCP pseudo-header, TCP header and TCP payload
  – By default, include TCP options

• Format Enhanced Authentication Option
  – Active key identifier
  – Flags
  – Message Authentication Code (MAC)
  – Authentication Identifier
Receiving System Procedure

• Lookup key specified by TCP Option
• Determine whether that key is eligible
  – Vector == in || vector == both
  – Start-time for receiving <= system time
  – End-time for receiving > end time
• Calculate MAC
• If calculated MAC is equal to received MAC, accept datagram
Authentication Error Procedure

- Discard datagram
- Log
- DO NOT send indication to originator
Coming Soon

• Automated session key distribution
  – Draft-weis-tcp-auth-auto-ks
Co-authors and Contributors

• Ron Bonica (Juniper)
• Brian Weis (Cisco)
• Sriram Viswanathan (Cisco)
• Andrew Lange (Alcatel)
• Owen Wheeler (BT)
• Chandrashekar Appanna (Cisco)
• Andy Heffernan (Juniper)
• Kapil Jain (Juniper)
• David McGrew (Cisco)
Next Step

• Accept as WG draft