Authentication for TCP-based Routing and Management Protocols

draft-bonica-tcp-auth-04

Motivation

- Operators need to authenticate TCP based routing protocols
  - BGP, LDP
- RFC 2385 does not fulfill operator requirement
- Many operators do not authenticate

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

Threats

- Operators are very concerned about keys that have been compromised due to employee turnover
  - It’s easy to revoke account when employee leaves
  - It’s hard to re-key every BGP session
- Operators are not so worried about cryptographic attack in which key is guessed

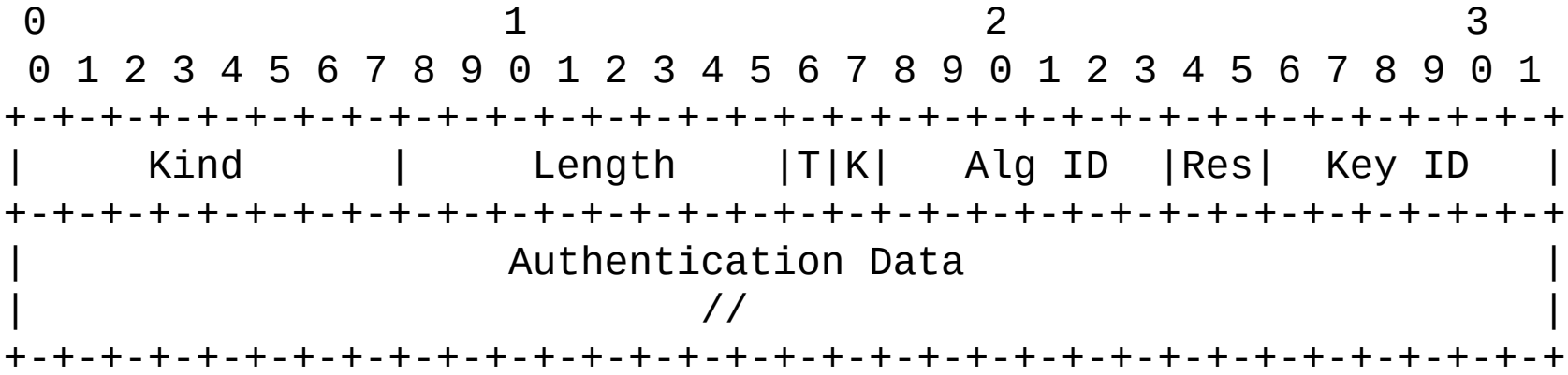
Alternative Approaches

- Application
  - In the Protocols (BGP, LDP, etc.)
  - TLS
- Transport
  - TCP
- Network
  - IKE/IPsec

Chosen 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

Enhanced Authentication Option



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

Why Did We Choose This Approach

- Operator Direction
  - Simplicity
  - Does not require third party certificates
  - Deals well with scenario in which long term key is compromised by employee turn-over
- Protects TCP control information
- Reasonable short term solution until a better mechanism is available

Co-authors and Contributors

- Ron Bonica (Juniper)
- Brian Weis (Cisco)
- Sriram Viswanathan (Cisco)
- Andrew Lange (Alcatel)
- Owen Wheeler (BT)
- Chandrashekhar Appanna (Cisco)
- Andy Heffernan (Juniper)
- Kapil Jain (Juniper)
- David McGrew (Cisco)
- Satish Mynam (mynam@cisco.com)
- Anantha Ramaiah (ananth@cisco.com)