Authentication for TCP-based Routing and Management Protocols

Ron Bonica

Andrew Lange

Sriram Viswanathan

Brian Weis

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 so much of an issue for Juniper, Cisco, Alcatel
  - Juniper, Cisco, and Alcatel architectures separates forwarding and control plane hardware
- Key management
  - Keys need to be refreshed periodically
  - Key refresh requires session reset
- Weak cryptography
  - There are many well-know attacks on MD5

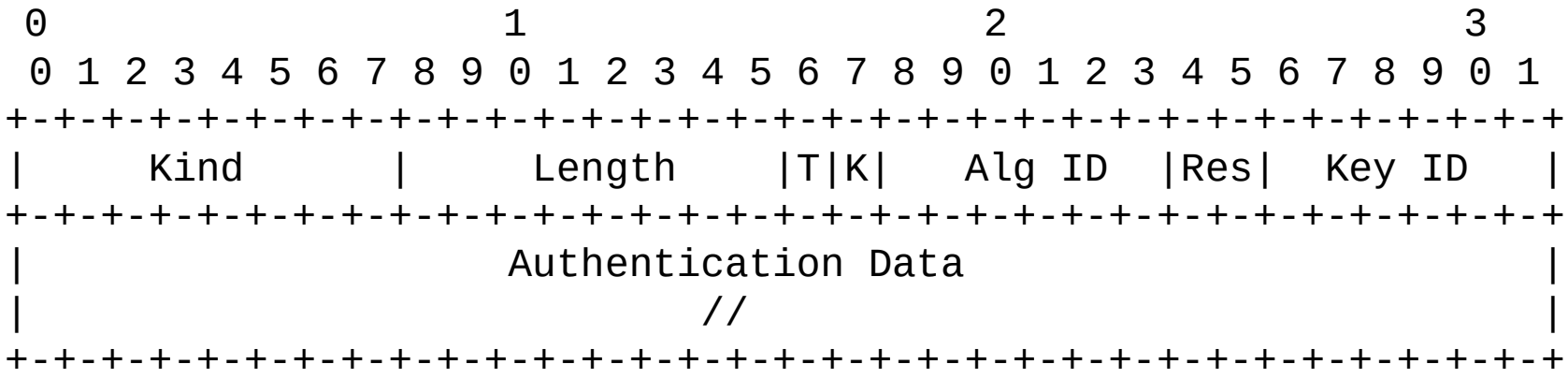
Alternative Approaches

- 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
  - Time based key roll-over
  - Key Identifier
- Stronger cryptography
  - HMAC-SHA-1-96
  - CMAC-AES-128-96

Enhanced Authentication Option



Key Chain

- Contains a tolerance parameter up to 64 keys
- Each key contains
  - Identifier [0..63]
  - Authentication Algorithm
  - Shared secret
  - Start and end time

Sending System Procedure

- Identify active key candidates
  - Start-time <= system-time
  - End-time > 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 Algorithm Identifier

Receiving System Procedure

- Lookup key specified by TCP Option
- Determine whether that key is eligible
  - Start-time <= system time - tolerance
  - End-time > end time + tolerance
- Calculate MAC
- If calculated MAC is equal to received MAC, accept datagram

Authentication Error Procedure

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

Configuration Example - Juniper

```
regress@UI-J6300-1> show configuration protocols bgp
authentication-algorithm hmac-sha-1-96;
authentication-key-chain ibgp;
local-as 65000;
group ibgp {
    type internal;
    neighbor 10.1.1.1;
    neighbor 10.2.2.2;
}
```

Configuration Example - Juniper (continued)

```
regress@UI-J6300-1> show configuration security
authentication-key-chains {
    key-chain ibgp {
        tolerance 200;
        key 1 {
            secret "$9$O.VeBSe7-ws4Z"; ## SECRET-DATA
            start-time 2006-1-1.00:00:00;
        }
        key 2 {
            secret "$9$1BYIrv-VY2aU"; ## SECRET-DATA
            start-time 2007-1-1.00:00:00;
        }
    |
}
```

Configuration Example - Alcatel

```
SR-12:ALA-21>config>router>bgp>group# info
authentication-key keychain ibgp
type internal
neighbor 10.1.1.1
neighbor 10.3.3.3
```

Configuration Example - Alcatel (continued)

```
SR-12:ALA-21>config>system>security# info
keychain “ibgp”
   description “ibgp keychain”
      direction send-receive
      algorithm hmac-sha1-96
      tolerance 200
         entry 1
            key “Vxg.Wea9xlbBIcXskkP00U”
            begin-time 2006-1-1.00:00:00
         entry 2
            key “J/cOv0nfdmHl5Ye2TbfuRk
            begin-time 2007-1-1.00:00:00
```

Configuration Example – Cisco Systems

```
Router>config>#
key chain mykeychain
   accept-tolerance 200
    key 1
          key-string  Vxg.Wea9xlbBIcXskkP00U
          cryptograhic-algorithm  HMAC-SHA-1-12
          send-lifetime    00:00:00 june 1 2006 duration 20000
          accept-lifetime 00:00:00 june 1 2006   23:00:00 july 1 2006
     key 2
          key-string  J/cOv0nfdmHl5Ye2TbfuRk
          cryptographic-algorithm HMAC-SHA-1-12
          send-lifetime    00:00:00 july 1 2006 12:00:00 october 1 2006
          accept-lifetime  00:00:00 july 1 2006  infinite
```

Configuration Example – Cisco Systems Contd..

Applying the keychain under neighbor group

```
Router>config>#
 router bgp 1
     neighbor-group xxx1
         keychain mykeychain
```

Applying the keychain under session group

```
Router> config>#
router bgp 1
    session-group xxx2
         keychain mykeychain
```

Applying the keychain under neighbor

```
Router> config>#
router bgp 1
    neighbor 1.1.1.1
         remote-as 111
          keychain mykeychain
```