The Security Flag for the IPv4 Header

RFC3514
Typical problems

- Distinguishing packets hard for
  - Firewalls
  - Packet filters
  - Intrusion detection systems

- Why
  - Unusual pkts
    vs. pkts of malicious intent

- Solution
  - Security flag in the IPv4 header
Syntax

- Unused bits
  - High-order bit of IP fragment offset field
- Assignment
  - Not left to IANA
Syntax (cont.)

- Bit layout
  0
  +---+
  | E |
  +---+

- Assigned values
  0x0 bit set to 0: no evil intent
    - Hosts, network elements, etc.
      SHOULD assume that the packet is harmless
      SHOULD NOT take defensive measures
      (already implemented by most OSs)
  0x1 bit set to 1: evil intent
    - Secure systems
      SHOULD try to defend themselves
    - Insecure systems
      MAY chose to crash, be penetrated, etc.
IANA consideration

- Document defines behavior of security elements of the 0x0 and 0x1 bit values.
- Behavior for other values of the bit MAY be defined only by IETF consensus [RFC2434].
Setting the security bit

- **Attack applications**
  - *MAY* use suitable API to request it be set
  - System requirements:
    - No other mechanisms for setting
    - *MUST* provide API; *MUST* be used by attack programs

- **Multi-level insecure OS**
  - Special level for attack programs
  - Bit *MUST* be set by default for pkts from this level
  - System *MAY* provide API to clear bit for non-malicious activity by users who normally engage in attack behavior
Setting the security bit (cont.)

- **Fragments**
  - If dangerous => **MUST** set bit
  - Pkt with bit fragmented
    - => **MUST** clear bit in fragments
    - => **MUST** set bit in reassembled pkt

- **Intermediate systems**
  - Used for laundering attack
  - Relayed pkts **SHOULD** have the bit set

- **Hand-craft applications**
  - Part of an attack => **MUST** set bit by themselves
Setting the security bit (cont.)

- Hosts inside firewalls
  - Axiom: no attackers inside \(\Rightarrow\) **MUST NOT** set bit
- NAT
  - Modify packets \(\Rightarrow\) **SHOULD** set evil bits
- Transparent proxies and email proxies
  - **SHOULD** set bit in reply to innocent clients
- Scans of hosts with Intrusion detection systems
  - Benign research
    - \(\Rightarrow\) bit **MUST NOT** be set
  - Ultimate intent evil and destination IDS that alerts
    - \(\Rightarrow\) bit **SHOULD** be set
Processing the security bit

- **Firewalls, etc.**
  - **MUST** drop all inbound packets with bit set
  - **MUST NOT** drop pkts with bit off
  - Dropped pkts **SHOULD** be accounted in MIB

- **IDS**
  - **MUST** apply probabilistic correction factor
    - Known propensity for false negatives/positives
    - Evil bit set  ➞ log attempt probabilistically
    - Evil bit clear ➞ log attempt probabilistically
  - A suitable admin interface **MUST** be provided
Processing the security bit

- **Routers**
  - Not security devices => SHOULD NOT examine bit

- **End-Hosts**
  - System dependent
  - MUST react appropriately according to their nature
Related work

- Only IPv4 evil bit
- IPv6 two options
  - Hop-by-hop option
    - Pkts that damage the network, e.g. DDoS
  - End-to-end option
    - Pkts intended to damage destination hosts
  - Contains a 128-bit strength evilness indicator
- Link layer
  - Bypass routers and hence firewalls
    => link-layer scheme MUST denote evil. E.g.:
    - Evil lambdas
    - Evil polarizations
Security considerations

- Functioning of security mechanisms depends critically on evil bit set properly.
- Faulty components:
  - Inappropriately evil bit = 0
    => firewalls will not function properly.
  - Inappropriately evil bit = 1
    => denial of service condition