Accountable Internet Protocol

Andersen et. al

Presented by:
Virajith Jalaparti

Securing the Internet

- S-BGP, so-BGP, PG-BGP, StopIt, Listen & Whisper…
- Fundamental Problem
  - No Accountability
- Use CRYPTO!!!
  - source spoofing
  - DOS
  - route hijacking
  - route forgery
- Can we do this without loosing aggregation?
- How can we get anonymity?

AIP

- Self-certifying addresses
- Use my public key as my address
- How to scale to size of Internet?
  - Network identifier
    - Accountability Domains (ADs)
  - End Host Identifier
- AD : EID : iface
- Other ways?
  - DHT of mapping from addresses to keys?
- EID associated with user rather than host

AIP

- stack of src and dest AD’s

Crypto vers (8)	Public key hash (144)	Interface (8)

Figure 1: The structure of an AIP address. For AD addresses, the interface bits are set to zero.

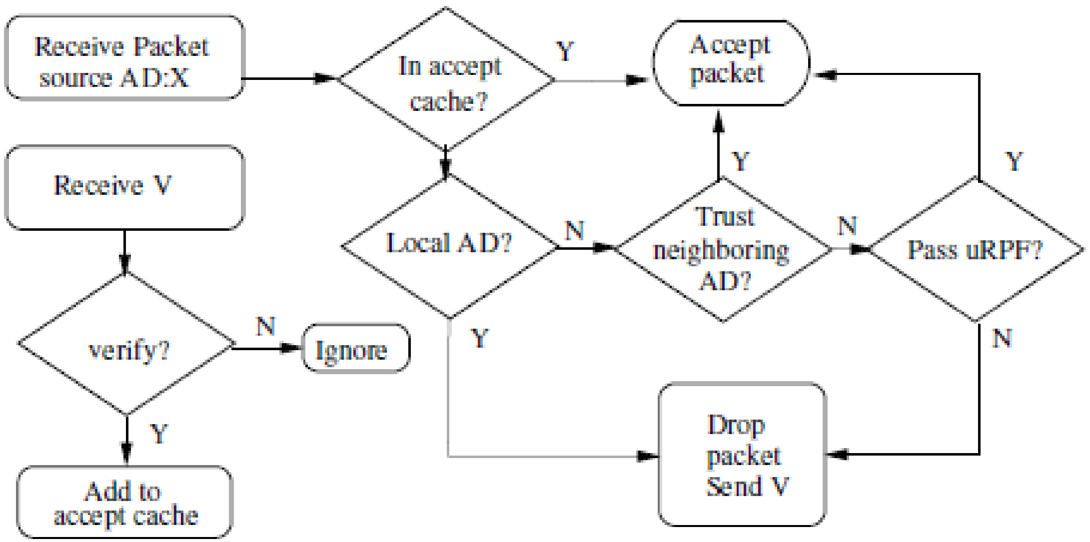
<table>
<tr><td>Vers (4)</td><td colspan="5">... standard IP headers ...</td></tr>
<tr><td colspan="2">...</td><td>random pkt id (32)</td><td>#dests (4)</td><td>next-dest (4)</td><td>#srcs (4)</td></tr>
<tr><td colspan="6">Source EID (160 bits)</td></tr>
<tr><td colspan="6">Source AD (top-level) (160 bits)</td></tr>
<tr><td colspan="6">Dest EID (160 bits)</td></tr>
<tr><td colspan="6">Dest AD (next hop) (160 bits)</td></tr>
<tr><td colspan="6">Dest AD stack (N*160 bits)</td></tr>
<tr><td colspan="6">Source AD stack (M*160 bits)</td></tr>
</table>

Routing

- Inter-domain
  - At AD level rather than AS level
    - Practical? Contracts between ASes
- Intra-domain
  - Use EIDs
  - Probably lots of entries in tables?

Source Spoofing

- First hop router verifies



- Should be done at switch level

Source Spoofing

Let:

$$
\begin{aligned}
rs &= \text{Per-router secret, rotated once per minute} \\
\text{HMAC}_{\text{key}}\langle \text{M} \rangle &= \text{Message authentication code of M} \\
H\langle P \rangle &= \text{Hash of } P \\
\text{iface} &= \text{Interface on which packet arrived}
\end{aligned}
$$

Source $S_{AD} : S_{EID} \rightarrow$ **Dest** $D_{AD} : D_{EID}$

Packet P.

Router R1 $\rightarrow$ **Source**:

Verification packet V =

$$\text{HMAC}_{rs}\langle S_{AD} : S_{EID} \rightarrow D_{AD} : D_{EID}, H\langle P \rangle, \text{iface} \rangle$$

Source $\rightarrow$ **R1**:

$$\{\text{accept}, K_{S_{EID}}, V\}_{K^{-1}_{S_{EID}}}$$

- Is this sufficient?
  - What happens after verification is passed?
  - First packet is a TCP-SYN, replay possible – use a nonce
  - Explicit tear down of connection

Inter-domain verification

- B -> A
  - A trusts B
  - uRPF check
  - Send a verification packet
- Border routers verify src addresses and add to *accept cache*
  - Wildcard AD:* to bound number of entries
  - can be exploited, it the checks in src AD does not perform proper checks

Minting of addresses

- Start connections with arbitrary EID
- Easy
- Solution
  - Limit number of EIDs per
    - interface on switches/routers
    - AD
- Is this sufficient?
- Cant prevent a DOS using minting
  - Using Bots

Shut-off Protocol

- Prevent DOS
- Use smart NIC
  - require physical access to modify the firmware
- cache packets sent
- Protects against replays

Zombie → Victim: Packet P.

Victim → Zombie:
$$\{\text{key} = K_{\text{victim } \mathtt{EID}}, \text{TTL}, \text{hash} = H\langle P \rangle\}_{K^{-1}_{\text{victim } \mathtt{EID}}}$$

- Is this sufficient?
  - Flooding attacks with bot-nets

Key Management

- Discovery
  - DNS - Secure
- Detect compromise
  - Use of global registries
    - Keys
    - Revoked Keys
    - Peerings
    - ADs of EID
    - First hop routes
- Dealing with compromise
  - Change DNS record, insert new key
- Will this work? Requires out of band techniques to fix

Discussion

- Does it work?
- Is it practical?
- Distribution of keys
- Possibility of creating optimal ADs?
- Probably run in combination with IP
- Application (Routing!) level security