A Light-Weight Distributed Scheme for Detecting IP Prefix Hijacks in Real-Time

Lusheng Ji†,
Joint work with Changxi Zheng‡, Dan Pei†, Jia Wang†, Paul Francis‡

† AT&T Labs - Research
‡ Cornell University

Outline

• Background
• Algorithms and Justifications
• Evaluation
• Conclusion
Prefix Hijacking Exploits BGP Authentication Weakness

BGP - the de facto inter-domain routing protocol
- Path vector protocol
- Lacking “authenticity” checking capability

Prefix hijacking: routers falsely advertise routes

Types of Prefix Hijacking
Current Approaches to Prefix Hijacking Prevention and Detection

- **Prevention**
  - Software/configuration changes
  - Public Key Infrastructure or other authentication mechanisms
  - Deployment hurdles

- **Detection**
  - BGP update message and routing table inspection and anomaly/signature detection
    - Limited vantage point locations
    - Difficult to be “real-time”
    - Often requiring privileged access
    - High false positive rates

New Approach: Data Plane Monitoring

**Benefits**
- Can have multiple strategically placed vantage points
  - Gotta have multiple
  - At good locations
  - Distributed work load
  - Distributed traffic load
- Potential of extending to overlay detection architecture
  - Robustness
  - Scalability
- Easily deployable, and anybody can do it.
Monitoring Prefix Network Location

The First observation

*If a prefix is hijacked, the paths observed from certain vantage points to the prefix would likely exhibit significant changes.*

Let’s start from monitoring the vantage point-to-target prefix paths
- What to measure

End-to-end Network Distance Measurement

End-to-end measurements
- Easy to obtain
- Low overhead

- Take one: end-to-end delay
  - Information rich
  - Not a good measurement target
- Take two: hop count
  - Relatively stable
  - Seems promising
Measurement Setup

Monitors (measurement sources)
- 43 Planetlab nodes (25 ASes)

Target prefixes (measurement destinations)
- Identified from RouteView and RIPE BGP tables
- 242 MOASes
- 125 SOASes

1 full month of data
- One hop count measurement for each path every 12 minutes

Hop Count Stability and Change Detection

- Time Series Analysis
  - Short-term moving average differs significantly from long-term moving average

Hop count relative variance
... But ...

Detection only based on hop count change may result in large false-positive ratios
  - Hop count is not that stable
  - How to quantify “significant”
  - Other reasons for “significant” hop count changes
    - MOAS changing entry/exit point
    - Traffic engineering
    - Natural/human disasters causing large Internet topology changes
    - Mis-configurations

Inspiration: Stuck in Traffic
Path Disagreement

Reference point
- As close to the target prefix as possible but outside of the target prefix AS.

Are path to a network and path to the reference point of the network similar?

Experiments on Planet-lab
- L1: longer path (i.e. path to a destination network).
- L2: shorter path (i.e. path to the reference point).
- Compute the “similarity” between L1 and L2:
  - L1’: the sub-path of L1 that starts from the same origin (source), but with length of |L2|.
  - HD: Hamming Distance.
  - S: path similarity.

$$s = 1 - \frac{HD(L_1', L_2)}{|L_2|}$$
Measurement Setup

Use the same set of monitors and target prefixes as before
One reference point for each monitor-to-target prefix path
Run a pair of traceroute probes every 12 minutes
  - Traceroute from monitor to target prefix
  - Traceroute from monitor to the reference point of the target prefix
One week of data
Convert hop by hop paths to AS paths
  - “Holes” in traceroute results
  - IP to AS mapping

AS Path Similarity

![Graph showing AS Path Similarity](image)
Hijacking Detection Scheme in a Nutshell

1. Select a set of monitors for each target prefix
2. Each monitor periodically measures the network distance to each target prefix and detects significant changes in network distance measurements
3. If a significant distance change is detected, the monitor measures the similarity between the path to the target prefix and the path to the reference point of the target prefix

Evaluation Methods

- We are about data plane and “real-time”
  - Difficult to evaluate using historical data
- Catching real hijacking attacks red handedly
  - But......
- Build simulation
  - Construct simulation scenarios based on real Internet topology
Simulating Prefix Hijacking Attacks

Imposture attacks
- One Planetlab node as the monitor \( s \)
- One target prefix as the victim \( t \)
- Another Planetlab node or target prefix as the hijacker \( h \)
- If \( s \) is closer to \( h \) than \( t \), imposture attack affects monitor, then
  \[ p(s,t) = p(s,h). \]
- Repeat for all possible selections of \( s, h, \) and \( t \)

Total of 34K imposture scenarios

Simulating Prefix Hijacking Attacks

Interception attacks
- Planetlab node as the monitor \( s \)
- Target prefix as the victim \( t \)
- Another Planetlab node as the hijacker \( h \)
- If \( s \) is closer to \( h \) than \( t \), interception attack affects monitor \( s \),
  \[ p(s,t) \approx \text{cat}(p(s,h), p(h,t)) \]
- Repeat for all possible selections of \( s, h, \) and \( t \)

Total of 25K interception scenarios
Hop Count Changes Due to Hijacking

AS Path Similarity After Hijacking

Path similarity decreases after hijacking.
## Hijacking Detection Accuracy

<table>
<thead>
<tr>
<th>Thresholds</th>
<th>False positive ratio</th>
<th>False negative ratio (interception)</th>
<th>False negative ratio (interception)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hop count, AS path</td>
<td>Hop count</td>
<td>Hop count</td>
<td>Hop count</td>
</tr>
<tr>
<td>0.04, 1.12</td>
<td>0.797%</td>
<td>0.223%</td>
<td>0.003%</td>
</tr>
<tr>
<td>0.125, 1.20</td>
<td>0.506%</td>
<td>0.199%</td>
<td>0.003%</td>
</tr>
<tr>
<td>0.25, 1.20</td>
<td>0.504%</td>
<td>0.192%</td>
<td>0.003%</td>
</tr>
<tr>
<td>0.25, 1.30</td>
<td>0.194%</td>
<td>0.173%</td>
<td>0.014%</td>
</tr>
</tbody>
</table>

## Discussion and Future Work

- Multiple monitors
  - Location and confidence level
- Granularity of detection
  - Subnet hijacking
- Counter measures
- Deployment
Conclusion

A light-weight distributed scheme for detecting IP prefix hijacks by conducting measurements in the data plane

• Hop count stability
• AS path similarity

Advantages
• Highly accurate
  - low false positive rate and low false negative rate
• Real-time
• Easy deployment
• Highly robust on monitor failure and attacker evasion

Thank You
Stability of Hop Counts

Change ratio over time

Change ratio: ratio of hop count of a later bin to that of an earlier bin

Hijacking Detection Latency

(a) Number of probes for imposture
(b) Number of probes for interception