Locating Prefix Hijackers using LOCK

USENIX 09

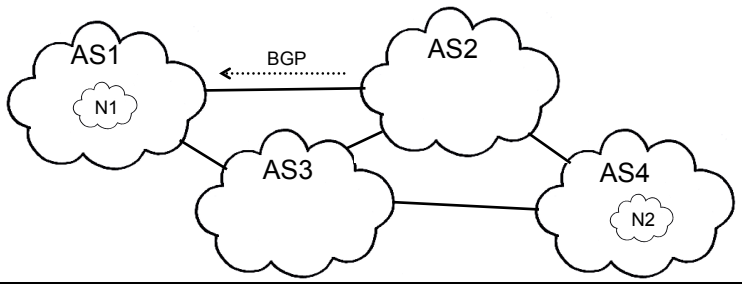
Authors: Tongqing Qiu (Georgia Tech), Lusheng Ji, Dan Pi, Jia Wang (AT&T), Jun Xu (Georgia Tech), and Hitesh Ballani (Cornell)

Presented by: Matthew Tan Creti

1/25

Background: AS

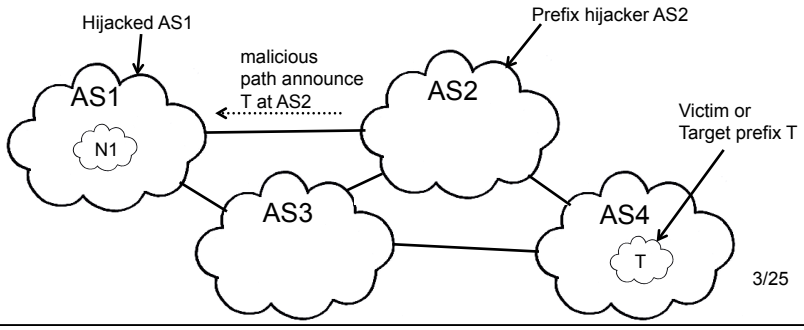
- AS (autonomous system) - the Internet consists of ASes, which consists of smaller networks
- BGP (Boarder Gateway Protocol) - provides interdomain (across AS) routing
- IP Prefix
  - the first n bits of an IP address
  - routes to a single AS
- Prefix hijacking - a misconfigured or malicious AS gives out false information on the route to a prefix



2/25

Background: Prefix Hijacking

- Happens due to misconfigured or malicious AS
- Hijacking AS might drop messages addressed to hijacking target
- Can be be used by attacker to intercept or snoop on traffic



3/25

Problem

- Locate a prefix hijacker AS
- Malicious hijacker may perform countermeasures such as modifying traceroute packets
- Detecting prefix hijacking is solved elsewhere, will not cover

4/25

General Solution Approach

- Observe AS path information from multiple vantage points using monitors
- Limited in what monitors can do and collect (ASes are rivals)
- Path information can be from:
  - Control plane: BGP route tables or messages
  - Data plane: AS-level traceroute

5/25

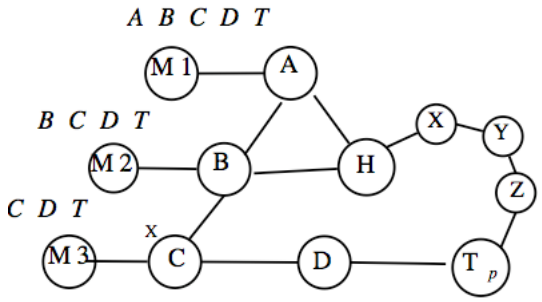
Path Information from Control and Data Plane

- Control plane
  - BPG route tables or messages
  - Difficult to collect in real time because BGP updates typically delayed a few hours
- Data plane
  - AS-level traceroute (map IP addresses in traceroute to ASes)
  - Easy to perform
  - Also easy for malicious AS to modify ICMP packets passing though it

6/25

Control Plane Before Attack

- Monitors M1, M2, M3 are shown with BGP route or traceroute data they have collected
- T is target AS and p is target prefix

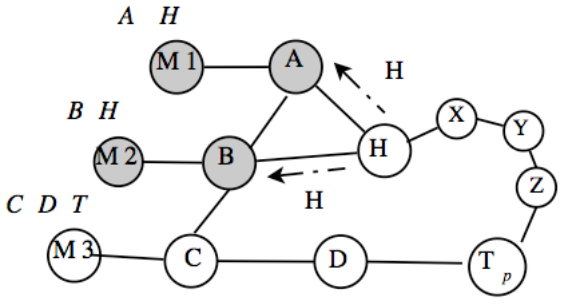


Unaffected AS

7/25

Control Plane: Hijacker as Origin

- Hijacker H claims to be origin of p



Unaffected AS

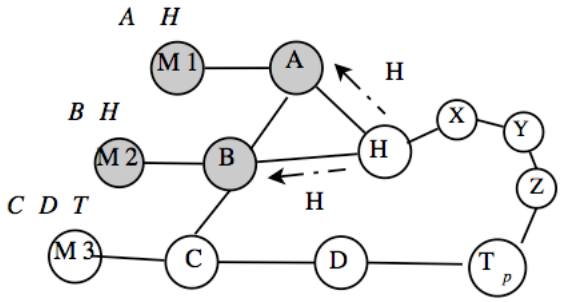
Polluted AS

AS Path Announcement

8/25

Simple Locating Approach

- Use route information at monitor to find origin H
- Declare H is hijacker

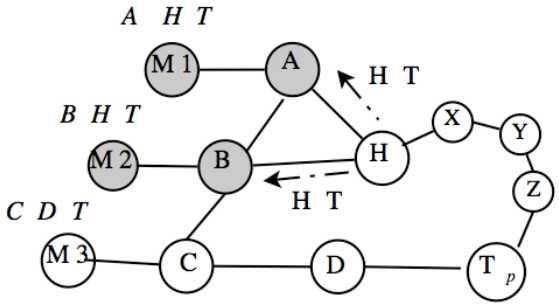


Unaffected AS
Polluted AS
AS Path Announcement

9/25

Control Plane: Hijacker as Neighbor of Target AS

- Hijacker provides correct origin for p
- Claims that it has link to p
- Could be cased by misconfiguration or malicious countermeasure to simple detection

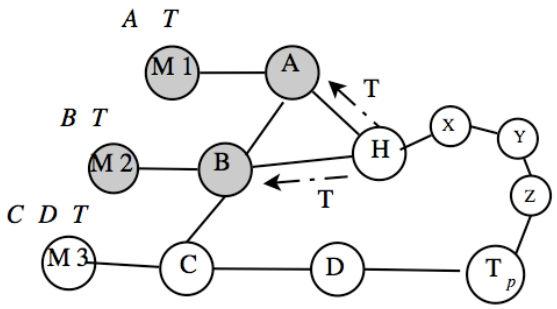


Unaffected AS
Polluted AS
AS Path Announcement

10/25

Control Plane: Hijacker as Target AS

- Hijacker completely removes itself from route
- Even an extension of simple approach to look at all ASes in route will not locate H

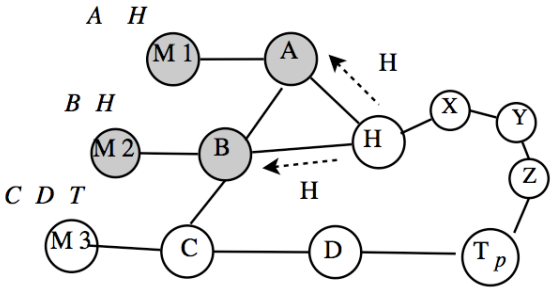


Unaffected AS
Polluted AS
AS Path Announcement

11/25

Traceroute: Blackholing

- In blackholing an AS claims a route to p, but drops traffic to p
- H responds honestly by dropping traceroute from monitors
- Simple approach can locate H in this case

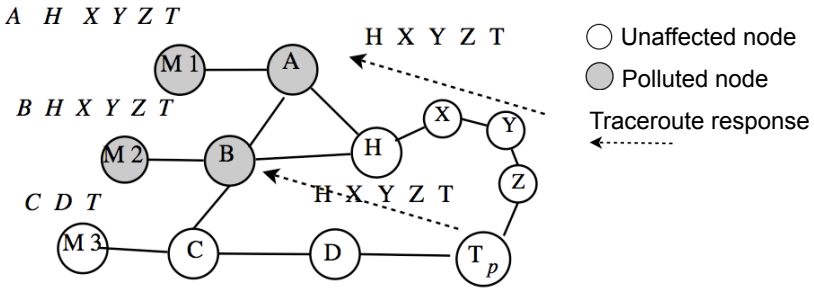


Unaffected node
Polluted node
Traceroute response

12/25

Traceroute: Interception

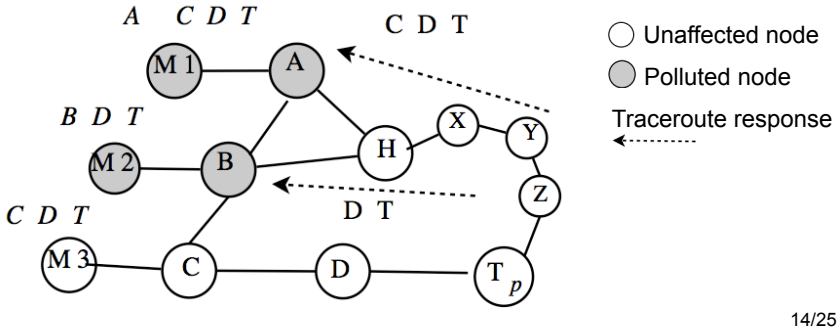
- In intercepting H allows messags to get to p
- Simple approach cannot find H



13/25

Traceroute: Manipulation

- Manipulation is a countermeasure by a malicious hijacker
- Simple approach will not work

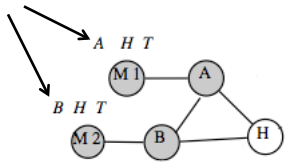


14/25

First Observations: Cannot Manipulate Upstream

- "The hijacker cannot manipulate the portion of the AS path from a polluted vantage point to the upstream neighbor AS of the hijacker AS"

Nodes A and B are upstream of H, so H cannot modify

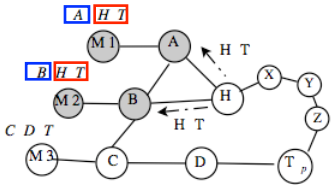


15/25

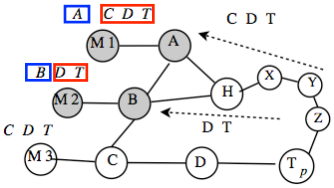
Second Observation: Polluted Paths Converge

- "The trustworthy portion of polluted AS paths from multiple vantage points to a hijacked victim prefix "converge" "around" the hijacker AS"

Control Plane



Data Plane



trustworthy
un-trustworthy

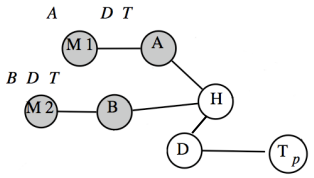
16/25

Basic Algorithm

- M is set of monitors that have detected a prefix hijacking
- P_i is monitor-to-prefix path for monitor m_i
- $N(P_i)$ is neighborhood set of nodes in P_i
  - Includes nodes in P_i
  - Taken from inferred AS topology data, not real time
- $H=U_iN(P_i)$ is the search space, hijacker must be in H
- From the search space rank each node $a_k \in H$
- $C(a_k)$ is covered count (how many P_i contain a_k)
- $D(a_k)$ is the total distance to monitors
- Sort first by C (higher count is higher rank) and then by D (lower distance is higher rank)

17/25

Example



- P1={A,D,T} P2={B,D,T}
- N(P1)={A,H,D,T} N(P2)={B,H,D,T}
- C(A) = 1 C(H) = 2 C(D) = 2
- D(A) = 1 D(H) = 4 D(H) = 4
- H and D are highest rank

18/25

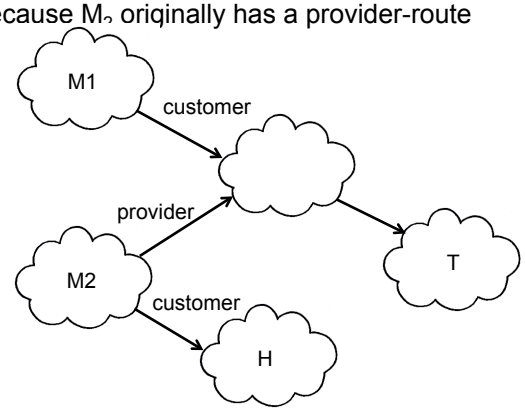
Monitor Selection

- Number and location of monitors have impact on accuracy
- Selection monitors should
  - Have high likelihood of observing hijackings
  - Have high diversity of paths
- Algorithm steps
  1 Clustering: monitors with similar paths to p are clustered together
  2 Ranking: monitors in each cluster are ranked based on probability of their paths being polluted
  3 Selection: select the highest ranked monitor from each cluster

19/25

Ranking Monitors

- Based on probability that monitor's path will be polluted
- Need to take into account nature of BGP routing
- Customers, peers, and providers
- It costs money to use a provider or peer
- M_2 would be more likely than M_1 to accept route offered by H, because M_2 originally has a provider-route



20/25

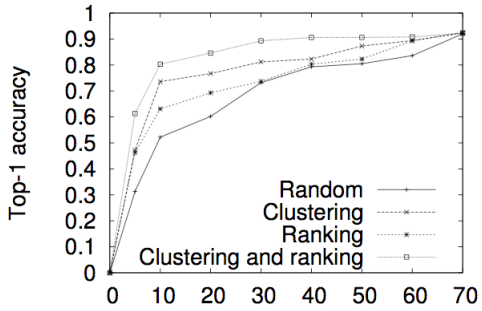
Evaluation

- PlanetLab - lab nodes actually distributed around world
- 73 candidate monitors in 63 ASes
- Setup synthetic prefix hijackings, reconstructed prefix hijackings, and performed actual prefix hijackings with hosts under authors control

21/25

Accuracy Based on Number and Selection of Monitors

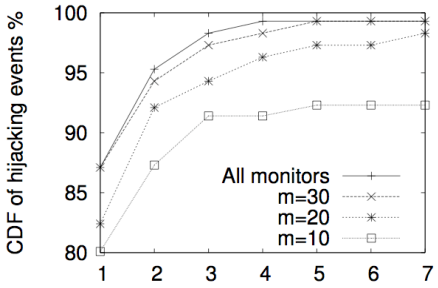
- Shows accuracy of highest ranked candidate
- Even with all monitors accuracy is 92%
- Because some hijacking have little impact



(a) Imposture

22/25

Ranking Accuracy



(a) Imposture

23/25

Results with Real Attacks

Table 5: Locating hijackers in real Internet attacks

Victim Site	Hijacker Site	Launch Time (EST)	Response Time (minutes)	Required monitors
Cornell	Berkeley	May 2 12:01:31	13	12
	Seattle	May 2 16:12:47	7	10
	Pittsburgh	May 2 17:34:39	9	9
Pittsburgh	Cornell	May 2 19:32:09	13	14
	Berkeley	May 2 22:50:25	11	15
	Seattle	May 3 02:26:26	12	15
Seattle	Cornell	May 3 11:20:42	9	8
	Pittsburgh	May 3 13:03:10	12	12
	Berkeley	May 3 19:16:16	8	18
Berkeley	Seattle	May 3 22:35:07	13	14
	Pittsburgh	May 4 00:01:01	12	16
	Cornell	May 4 11:19:20	11	10

24/25

Conclusion

- LOCK can use either control or data plane information
- Unified approach to locating hijacker that uses different countermeasures
- Robust because increasing monitors improves accuracy

25/25