Scalable Analysis for Network Monitoring and Forensics Purposes

Jérôme François

UNIVERSITÉ DU LUXEMBOURG

SNT

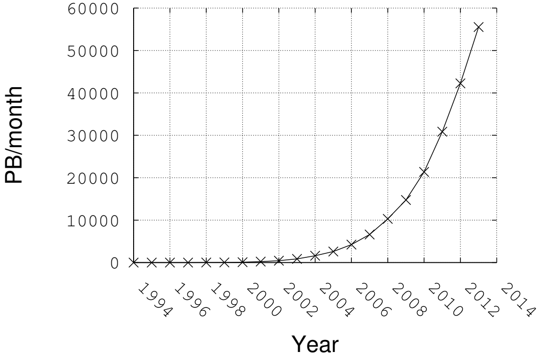
Outline

Outline

Outline

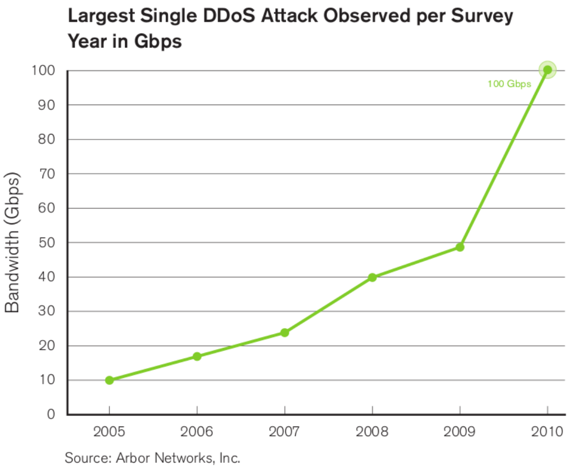
Internet is big

- 2,2 billions users, 200 millions servers
  - Cisco measured and forecasted Internet traffic (1000 PB/day)



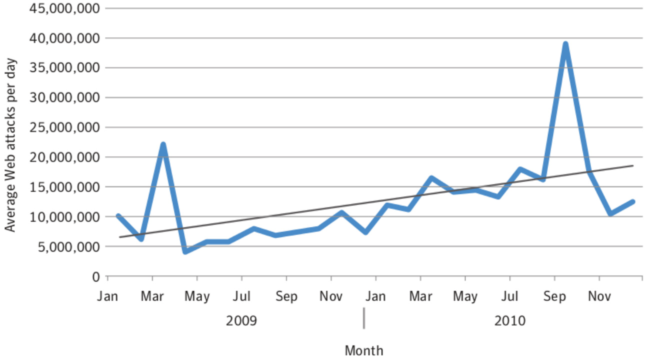
Attack aggressiveness increases

- DDoS Attacks



Some facts

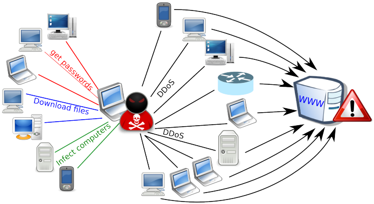
- Web based attacks



Average Web-based attacks per day, by month, 2009–2010
Source: Symantec Corporation

Some facts about botnets

- Botnets



- Botnet monitoring (*Measurement, Detection, Disinfection and Defence* , ENISA report 2011):
  - Shadowserver Foundation: 5000-6000 alive botnets (100000-250000 bots) simultaneously in 2005
  - Conficker working group: 1 000 000 - 3 000 000 alive zombies (2009)
  - Securelist.com: 3 600 000 zombies within US only (2009)

Outline

1. Introduction
   - Some facts
   - Motivation
2. Traffic analysis
   - Anomaly detection
   - Botnet detection
3. Topology analysis
   - Bad behaviors in Internet
   - Detection
   - Evaluation
4. Conclusion

Why attacks are powerful?

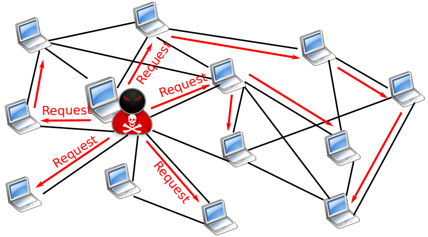
- Motivation
  - challenging aspects / attacker competitiveness... past trend, too risky today
  - win money!
    - abuse (spam, click fraud)
    - attack the competitors (steal information, disrupt services)
    - 15\$ = 10 000 bots (source: Symantec)
    - Zeus botnet: 70\$ million stolen from victim bank accounts
    - $\rightarrow$ costs: 388 billions \$ (source: Symantec 2010)
- And also:
  - more complex attack mechanisms
  - more available bandwidth
  - more users
  - more devices (Internet everywhere)
  - more on-line services

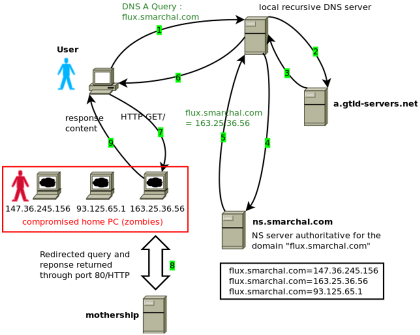
Network security

- Context
  - Growth of Internet / network sizes, heterogeneity, mobility
  - Continuous arising new threats, high sophistication
  - Cyber criminality = new motivations
- Network security:
  1. prevention / proaction
  2. detection
  3. reaction
- Network security $\rightarrow$ observations $\rightarrow$ network monitoring

Attack mechanisms

- Multiple infection vectors: direct attack, email, pdf, instant messaging, social networks
- Distributed attacks (botnet → DDoS, spam,...)
  - Multi-hops attacks
  - Enhancement of malware robustness: fastflux, double-flux





Problematic

- Challenges:
  - local view inefficient against distributed attacks $\rightarrow$ collect global and multiple information (network traffic, DNS domains, used applications, etc)
    - detect attacks at the operator levels
    - collect global data about the network from individual location
  - scalability: storage and analyze large volume of data (60,000 flows/second, millions of hosts, etc)
    - aggregate information
    - combine individual information = collaborative security
    - distributed computing
  - privacy:
    - sensitive infomation to analyze (user tracking)
    - multiple sources / information sharing

Outline

Outline

(Net)flow records

- Condensed information about a traffic "instance"
  timestamp, Ip src, Ip dst, protocol, #bytes, #pkts, etc
- Advantages:
  - Widely available at ISP level
  - No payload $\rightarrow\sim$ privacy preserving
- Challenges:
  - Few information
  - Huge volume of data (100 000 flows/second)
- $\rightarrow$ combine multiple flow records to highlight malicious activities

Aggregation

Aggregation

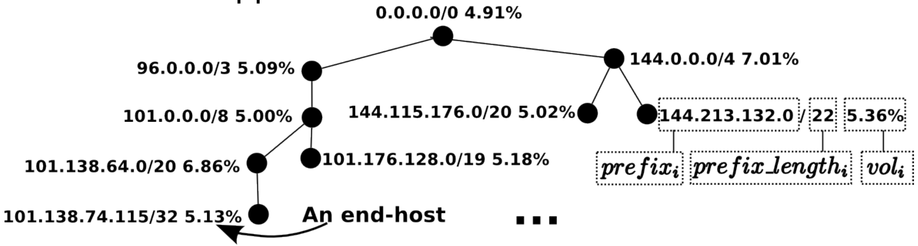
- Scalable way to represent information
  - Outline relevant correlated facts
  - reduce storage needs and post processing time
- Temporal and Spatial aggregation
  - temporal: time windows split (β)
  - spatial: keep nodes with activity $> \alpha$ *e.g. traffic volume*, aggregate the others into their parents $\rightarrow$ needs hierarchical relationships
- Heterogeneous Data
  - No specific order
    - ~~1st Source IP@, 2nd Destination IP@~~
  - Auto adjust to Information Granularity
    - ~~/18 /24 /27 subnetworks...~~

Mutidimensional Aggregation Example

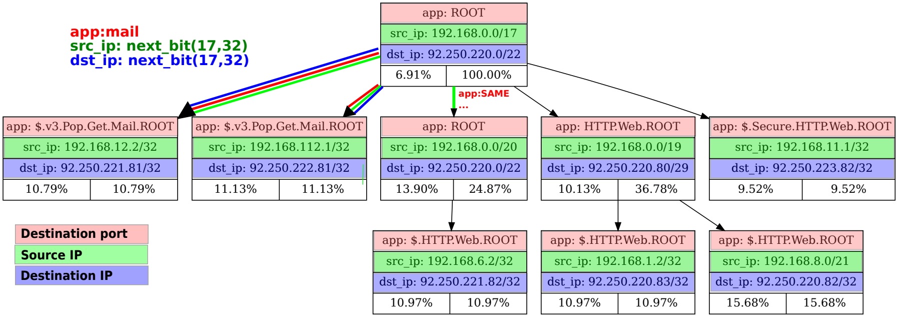
PORT	PROTO	KB	TIME	SOURCE	DEST
80	TCP	1491	2010−02−24 02:20:15	192.168.6.2	92.250.221.82
110	TCP	988	2010−02−24 02:20:19	192.168.8.2	92.250.223.87
443	TCP	902	2010−02−24 02:20:27	192.168.11.2	92.250.220.82
110	TCP	1513	2010−02−24 02:20:29	192.168.112.1	92.250.222.81
80	TCP	1205	2010−02−24 02:20:29	192.168.11.1	92.250.220.82
80	TCP	1491	2010−02−24 02:20:31	192.168.1.2	92.250.220.83
110	TCP	1467	2010−02−24 02:20:39	192.168.12.2	92.250.221.81
80	TCP	927	2010−02−24 02:20:39	192.168.12.2	92.250.220.82
443	TCP	1294	2010−02−24 02:20:39	192.168.11.1	92.250.223.82
110	TCP	940	2010−02−24 02:20:49	192.168.21.2	92.250.221.81
80	TCP	917	2010−02−24 02:20:49	192.168.23.1	92.250.220.82
443	TCP	460	2010−02−24 02:20:59	192.168.26.2	92.250.220.85

Mutidimensional Aggregation Example

- Previous approach:



Mutidimensional Aggregation Example



Data Structure

Tree based structure: Root node and multiple children

Directions

- How to find the right path to insert a node within a tree?
- Direction function
  - Most specific ancestor common ancestor between two nodes
  - Longest common prefix match
- IPv4: binary function (0,1) as next bit value
- DNS: every level name is a direction
- ports: service taxonomy

Optimization

Aggregation

- From leafs to root node
- On a complete tree of a time window
- → Large data structures in memory before aggregation

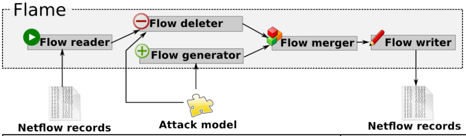
Online Strategies (before the end of the time window)

- Tree size > MAX_NODES → aggregation

	Root	LRU
	Aggregation is triggered from root node	Aggregation is triggered in the least recently used node
RAM	+	+
Performance	- -	-

Datasets

- Real ISP data + attack injection



# Flows	3 907 859
# IP Addresses	source addresses : 250 314 destination addresses: 235 120
# bytes	24.1 GB
Avg. bytes/Flow	6 829
# Packets	38 132 130
Avg. Packets/Flow	9.76
# UDP Flows	2 756 321
# TCP Flows	1 097 030
# ICMP Flows	50 914
# Other Protocol Flows	3 594

Results

- Source and destination IP address + distance → decision tree
- average tree size = 3288, 90 (after aggr.)

Type of Attack	Results	
	TPR	FPR
Nachi scan	0.912	0.222
Netbios scan	0.941	0.185
Popup Spam	0.882	0.361
SSh scan + TCP flood	0.882	0.028
DDoS UDP flood	0.923	0.077
DDoS TCP flood	0.887	0.027
DDoS UDP flood + traffic deletion	0.932	0.072

- False positive reduction → compare Netflow without aggregation (Networking'11)
- Aggregation → better to detect large scale attacks

Outcome

- Anomaly detection in ISP network
  - privacy preserving → Netflow data
  - low complexity:
    - LRU algorithm (Least Recently Used) → maximal size fixed to 128
    - usually lower in practice
  - Dynamic granularity over the IP address space
    - granularity is guided by the events to monitor...
    - ...not by the size of space to monitor
- tool: `https://github.com/jfrancois/mam`
- Publications: Networking'11, LISA'12

Outline

1. Introduction
   - Some facts
   - Motivation
2. Traffic analysis
   - Anomaly detection
   - Botnet detection
3. Topology analysis
   - Bad behaviors in Internet
   - Detection
   - Evaluation
4. Conclusion

Overview

- Botnet architecture: Command & Control (C&C) to propagate orders
  - centralized approach (IRC, HTTP)
  - structured P2P botnet: high performance
- Detection (state of the art)
  - detect large volumes of related attacks
  - centralized botnets: detect central component
  - P2P botnets: active participation
- Objective: passive detection of P2P botnets which do not generate high volume of traffic (data stealing / espionage, stealthy infection)

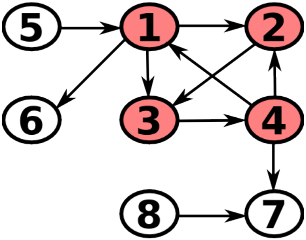
Contribution

- Discover the C&C channel at the ISP level:
  - NetFlow monitoring → who talks to whom ? (dependency graph)
  - linkage analysis + clustering techniques → identify groups of hosts sharing similar behaviors
  - MapReduce implementation
  - experiments using real NetFlow data

Dependency graph

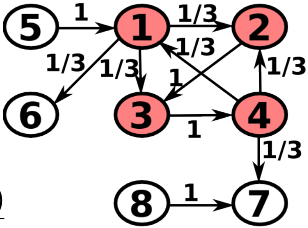
- Who talks to whom ?
  - bots have a distinguishable communication patterns
  - bots are well interconnected together

- Trivial example: bots = 1, 2, 3, 4
- automatic analysis:
  - local view: node adjacency, benign hosts well interconnected (server)
  - global view: a bot may be connected to few others which are connected to few others and so one + loops → they are globally well interconnected together



PageRank

- Global link analysis
  - Google web page ranking algorithm
  - a page/host is highly scored if it is well pointed by others especially if these latter have high scores
- Iterative computation
  - equal score at the begin
  - stop when stable
  - score propagation
  - weighted nodes (bot knowledge)

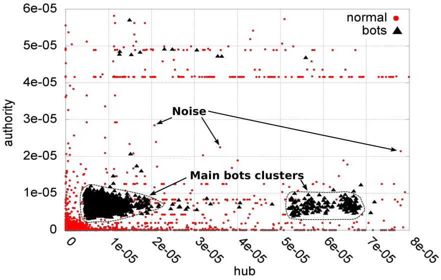


$$P_t(i) = (1-d)\sum_{k=1}^{n} W(k) + d \sum_{(j,i)\in E} \frac{P_{t-1}(j)}{O_j}$$

- Both communication directions are important → invert arrows → two values per node: hub, authority

Clustering

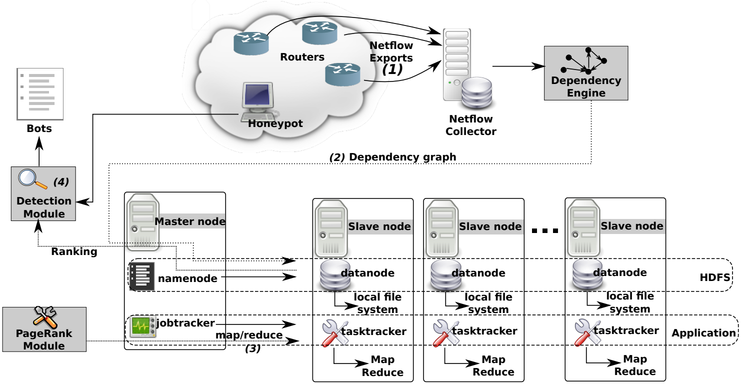
- Inefficiency of pure link analysis
  - benign hosts may be highly scored (popular services)
  - bots $\rightarrow$ similar communication patterns
  - botnet might be partitioned (randomness of connection, disruption)
  - simple thresholds not well fitted



- Clustering
  - find similarly scored hosts
  - unsupervised algorithm + few parameters
  - $\rightarrow$ DBSCAN: density based

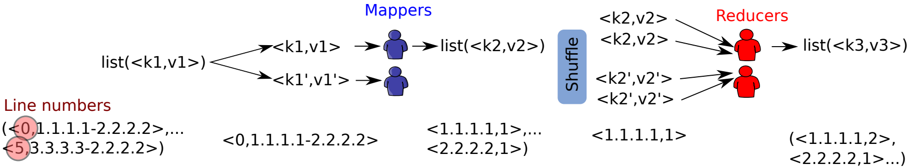
Cluster distinction

- A cluster can be composed of benign hosts → necessary prior knowledge about the botnet:
  - one bot per cluster → all the hosts of the clusters are bots
  - additional tool: honeypot, blacklists, IDS, etc.



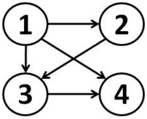
Formal Map-Reduce

- Map-Reduce:
  - data-intensive processing
  - shift the the network transfer from the data to the code
  - approach based on $\langle key, value \rangle$ pairs:
    - map input: $\langle k1, v1 \rangle$ ($k1$: line number, filename... but rarely used for further usage)
    - intermediate between mappers and reducers: $\langle k2, v2 \rangle$
    - reduce output: $\langle k3, v3 \rangle$
  - partitioner: $k2 \rightarrow Reducers$

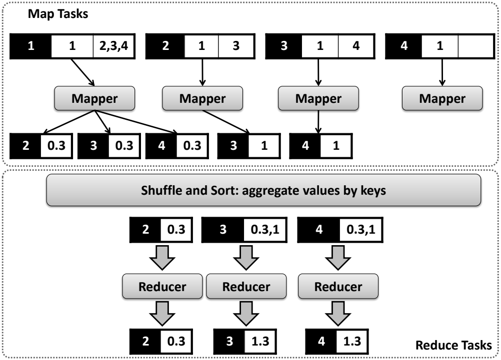


PageRank on Map-Reduce

- Node = ID [key] + (score + adjacent nodes) [value]



Key	Value	
	Current Score	Adjacency List
1	1	2 3 4
2	1	3
3	1	4
4	1	



SnT

Real data issue

- Netflow ISP Data containing labeled botnet C&C traffic $\rightarrow$ impossible
- Compromise:
  - real data (considered as to being free of botnets)
  - synthetic botnet traffic injected
- P2P botnet traffic $\rightarrow$ define host relationships:
  - id space: $N = 2^{160}$
  - Chord (DHT) $\rightarrow$ theoretical but generic: routing in $log(N)$
  - Kademlia (XOR metric): routing in $O(log(N))$ but with a high redundancy $\rightarrow$ high robustness
  - Koorde (sub-partitioning): routing in $O(log(N)/log(log(N))$ with a low redundancy $\rightarrow$ less robustness

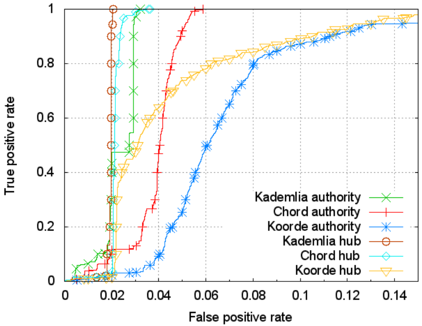
Statistics

- Stealthy botnets: 1% of IP addresses
- Bot IP addresses randomly and uniformly selected

	chord	Kademlia	Koorde
Flow#	2133887	2399032	1997049
Host#	323610	323610	323610
Bytes#	13.7G	13.7G	13.7G
Duration	18min23sec	18min23sec	18min23sec

Pure Link Analysis

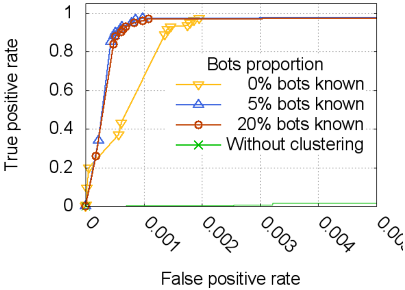
- Without clustering:
  - threshold based method
  - threshold varies to compute both true positive and false positive rates



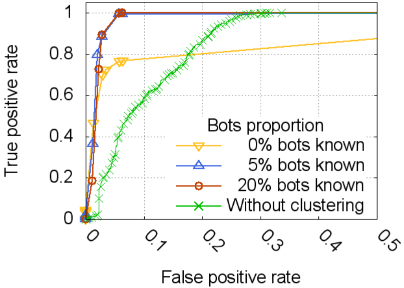
- High redundancy $\rightarrow$ easy detection (Kademlia)
- Hub values are more discriminative
- FPR = 2% = 6400 FPs $\rightarrow$ still needed to improve the accuracy

With Clustering

- Clustering → better accuracy
  - Kademlia: TPR = 99%, FPR = 0.2%
  - Koorde: less redundancy → more noise points with DBSCAN → clustering is better before a certain threshold
- Bot knowledge: significant impact only with Chord



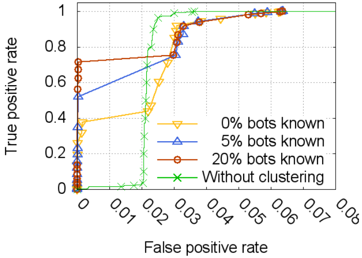
- Kademlia



- Koorde

With Clustering

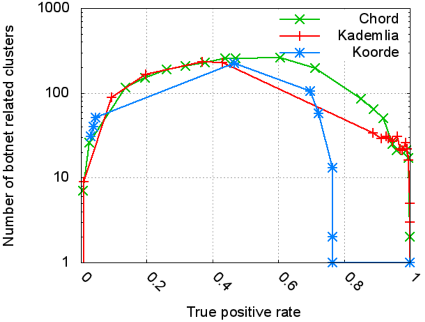
- Clustering → better accuracy
  - Kademlia: TPR = 99%, FPR = 0.2%
  - Koorde: less redundancy → more noise points with DBSCAN → clustering is better before a certain threshold
- Bot knowledge: significant impact only with Chord



- Chord

Cluster analysis

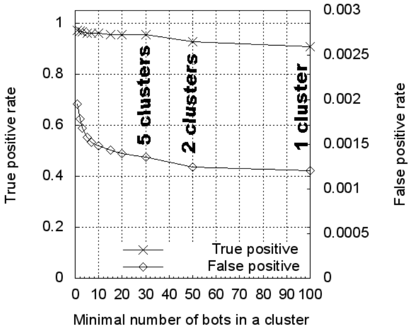
- Unrealistic extrema cases for detecting all botnets
  - one single cluster → huge number of false positives (ROC curves)
  - one cluster per bot → all the botnet monitored by the honeypot



- High TPR without one bot per cluster
- Best tradeoff obtaining with few clusters: worst case (Chord): TPR = 0.96, FPR = 0.04, 21 clusters

Cluster analysis

- Knowledge: 20 bots
- Importance of each cluster ?
  - cluster with few bots
  - only needed to monitor huge clusters → limits the knowledge requirements

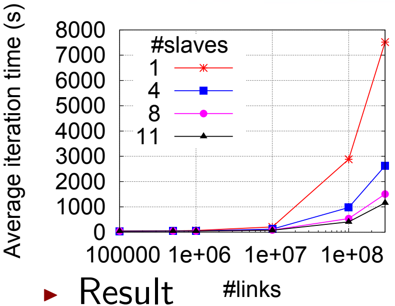


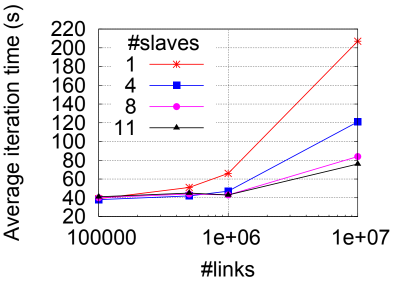
- Kademlia + discard smallest clusters
  - low impact on true positives: 92% with only 2 clusters
  - significant reduction of false positives

Efficiency analysis

- Efficiency analysis
  - score forwarded through the links → number of nodes has no impact + number of intermediate (key,value) pairs depends on the number of links
  - test different size of dataset → subset between 100k and 300M links
  - different Hadoop cluster configurations (number of machines)

Efficiency analysis





- Result
  - linear increase (execution time divided by 7 for a huge dataset)
  - #links x 10 $\rightarrow$ execution time x 6 (8 slaves)
  - few links $\rightarrow$ no improvement due to overhead of Map-Reduce (data split, reduce phase)
  - $<$ 1M #links $\rightarrow$ Hadoop useless
  - $>$ 10M #links $\rightarrow$ 4 slaves are useful

Outcome

- Detection of botnets:
  - structured P2P networks
  - ISP level / IP flow monitoring (passive approach)
  - 2 levels approach: link analysis + clustering
  - Some prior knowledge (additional source of information like honeypot)
  - Scalability: 18min of monitoring handled in 160 seconds
  - publications: Networking 11, WIFS'11
- Future work: how to alleviate the need of a honeypot / relying only on traffic observation $\rightarrow$ service dependency

SnT

Outline

Outline

Malicious AS

- Autonomous Systems
  - BGP routing $\rightarrow$ routing table = AS paths (sequence of AS to reach an IP subnet)
- Malware *providers* needs also hosting (malware, C&C servers, phishing website...)
  - detection: monitoring, complains, reports,...
  - Operators can disconnect/blacklist malware hosters
- $\rightarrow$ some AS are not blocking their malicious users
  - some AS are more *tolerant* for hosting services (money-driven, political-driven...)
  - malicious entities are their own operators

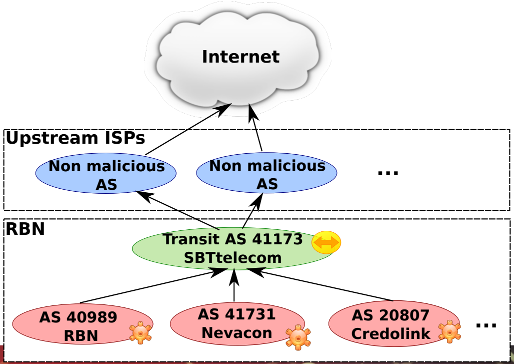
BGP ranking

- How to detect AS hosting malware → BGP ranking (`http://bgpranking.circl.lu/`)
  - ∼ ASs administrated by cyber-criminal organization = malicious AS
  - blacklists of IP addresses involved in malicious activities
  - map IP addresses to ASs → compute a score for each AS = detection
  - → neighbor ASs can react (de-peering, complains)

$$AS_{rank}(ASx) = 1 + \frac{\sum_{b \in BL} occ(b, ASx) b_{impact}}{ASx_{size}}$$

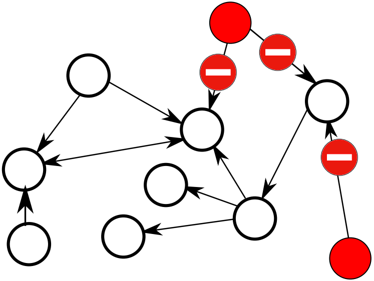
Malware Transit AS

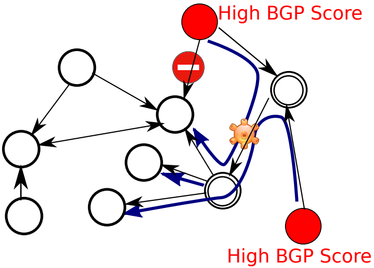
- Avoid detection $\rightarrow$ hide malicious AS behind ASs looking normal (malware transit AS)
- Complex cyber-criminal organization networks $\sim$ long manual investigation
  - Russian Business Network: 3 years before being disrupted



Contribution

- Detection of malware transit AS not filtering their bad neighbors
  - Accurate AS graph based analysis
  - Global
    - investigation not focused on a single AS
    - not only at the first hop
  - Efficiency = real-time (route stability $\sim$ 1 day)





Outline

Theoretical measure

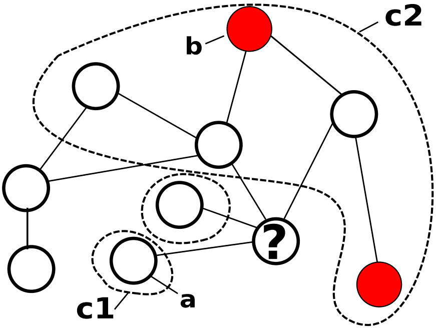
- BGP routes = who provides transit to whom
- Theoretical measure
  - extract pair of ASs which are connected through the evaluated AS
  - evaluate the potential impact of an AS to an another one

$$MT(ASx) = \sum_{\substack{(ASy,ASz) \\ \in\{(a,b)|a \overset{ASx}{\rightarrow} b\}}} \frac{(AS_{rank}(ASy) - AS_{rank}(ASz))^{+}}{card(\{ASu \in V, ASy \overset{ASu}{\rightarrow} ASz\})}$$

- Issues
  - voluminous number of routes → high complexity
  - instability of routes → needs to collect data over long time period to avoid a bias
- → compress routes into an AS graph

Practical measures

- AS graph $\rightarrow$ lost of exact transit information
- Approximation: *a malicious AS A can provide malware to AS B through AS C if all paths from A to B goes by C*
- $\rightarrow$ limit analysis to k hops around AS B
- Normalization regarding the number of neighbors
- Issue: single AS analysis

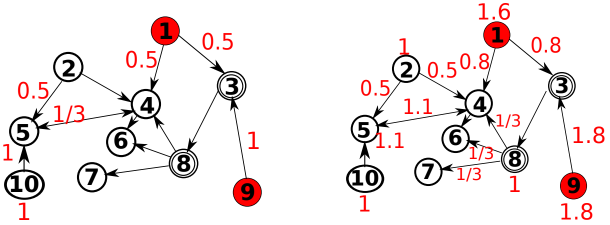


$$MT'_k(ASx) = \frac{\sum_{(c1,c2)\in pairs(C_k)} \left| \sum_{a\in c1} Rank_a - \sum_{b\in c2} Rank_b \right|}{\#neighs(ASx)}$$

PageRank

- Global link analysis
  - Google: a page/host is highly scored if it is well pointed by others especially if these latter have high scores
  - unweighted vs. weighted (BGP ranking)

$$P_t(i) = (1-d)\sum_{k=1}^{n} W(k) + d\sum_{(j,i)\in E} \frac{P_{t-1}(j)}{O_j}$$

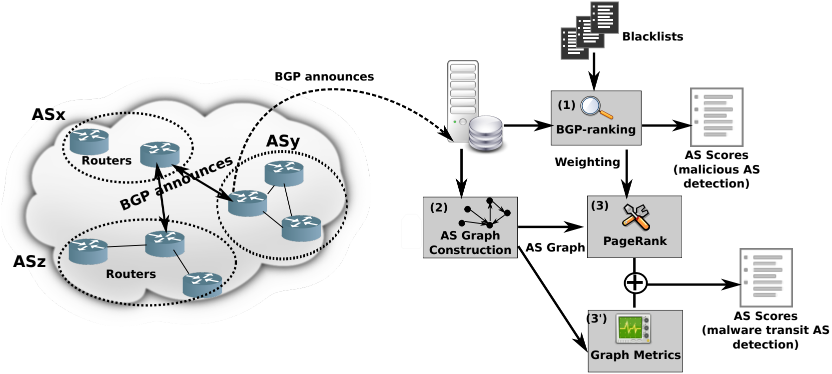


- Normalization → average score of ASs having the same number of neighbors

$$P'_t(i) = P_t(i) - \frac{\sum_{j\in V, \#neighs(j)=\#neighs(i)} P_t(j)}{card(\{j\in V, \#neighs(j)=\#neighs(i)\})}$$

System overview

- Input: BGP announces
  - BGP ranking (additional input/knowledge)
  - AS graph representation → graph analysis



Outline

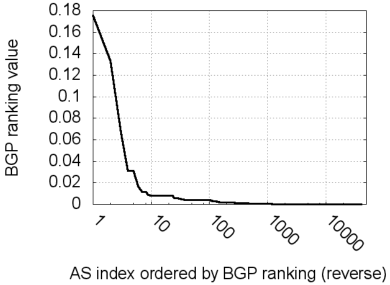
SnT

Dataset and Methodology

- Dataset
  - BGP route announces collected at rrc00.ripe.net (Amsterdam)
  - April 2012, 41k ASs
  - AS paths: 7243k / 1028k (unique)
  - As graph edges: 95k
- Methodology
  - no groundtruth
  - → use theoretical estimation = natural definition of malware transit AS
    - cannot be applied to all ASs
    - → check coherency of the output of PageRank-based approach with the theoretical estimation

Validation

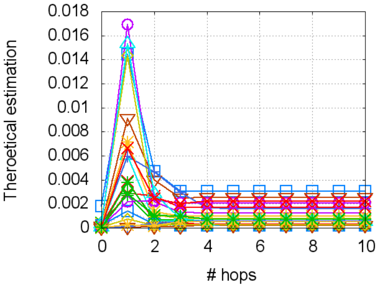
- A minority of malicious AS...
- ... but not blocked



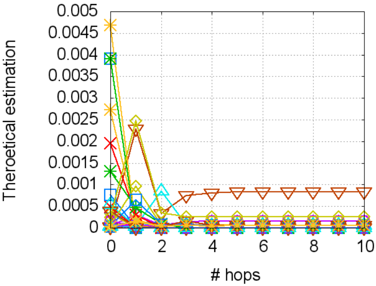
- PageRank-based analysis
  - Damping factor impacts a lot the results
  - variation coefficient $(\sigma/\mu) = 0.41$
  - Criteria
    - Always in top 30 → Malware Transit AS → 23 AS
    - Always out of top 30 → Normal AS
  - Worst case analysis: normal AS in top 30-100 → 30 AS

Validation

- Theoretical estimation (single AS) of selected AS
  - Malware transit AS are clearly distinguishable $\rightarrow$ global analysis is coherent with the natural definition
- First value (index 0) = BGP ranking
  - no correlation between BGP ranking and the malware transit measure
  - $\rightarrow$ the malware hoster are not the malware forwarder
- Malware transit AS

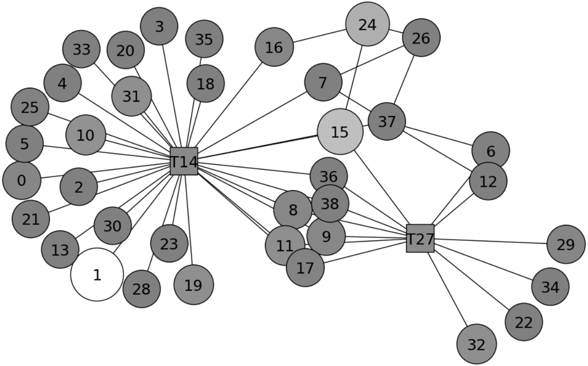


- Normal AS



Example

- Sample topology extraction
  - 2 malware transit ASs: T14, T27
  - High BGP ranking → light color, higher size



Outcome

- Malware transit AS detection
  - domain not well covered until now
  - graph analysis approach → global analysis + low complexity
  - practical validation → *famous* countries
  - publication: IM'13
- Future work
  - enhanced metric / graph analysis
  - time series evaluation

SnT

Outline

Conclusion

- Graph analysis = accurate way to assess security in Internet
  - data selection? → what should a graph represent and highlight?
  - analysis → more sophisticated method ?
- Some issues
  - algorithm tuning → learning
  - datasets
    - real data including various users, services, etc.
    - labeled traffic (attacks)
    - recent
    - → `www.caida.org/data/`

Scalable Analysis for Network Monitoring and Forensics Purposes

Jérôme François

UNIVERSITÉ DU LUXEMBOURG