Botnet Detection Combining DNS and Honeypot Data

NTT Information Sharing Platform Labs.

Keisuke ISHIBASHI, Tsuyoshi TOYONO, and Makoto IWAMURA

Outline

- Motivation
- Problem of Black domain list
- Graph kernels
- Random walk sampling
- Experimental results
- Conclusion

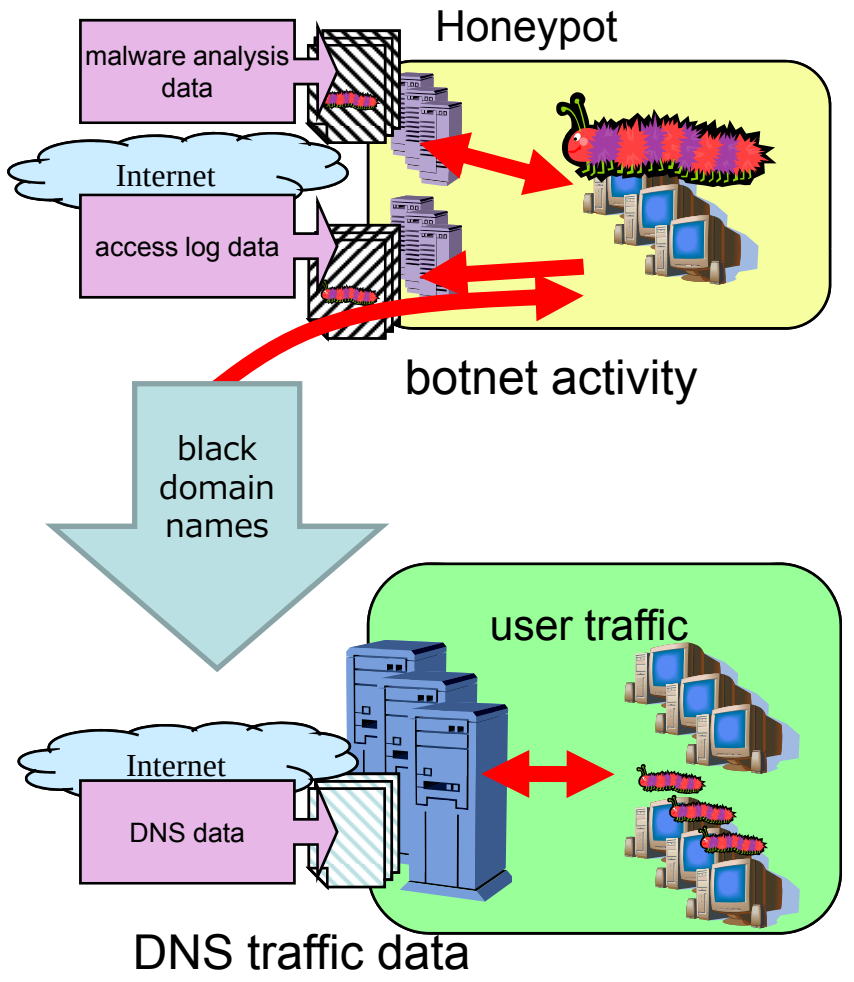
Motivation

- Monitoring DNS traffic is an effective approach for detecting botnet behavior [Kristoff05]
  - By matching DNS traffic data with black domain names that are involved with botnet activities, we can detect botnet-infected hosts.
- Quality of black domain-name list is key for this detection
  - Should not include white domain names
  - Should include black domain names as much as possible
- We talk about how to refine the blacklist

[Kristoff05] J. Kristoff, "Botnets, detection and mitigation: DNS-based techniques," Information Security Day, Northwestern University, July, 2005.

Black Domain-name List

- Get black domain names involved with botnet activity which is monitored in honeypot environment
- Match black domain names with DNS traffic data
- Detect botnet-infected hosts



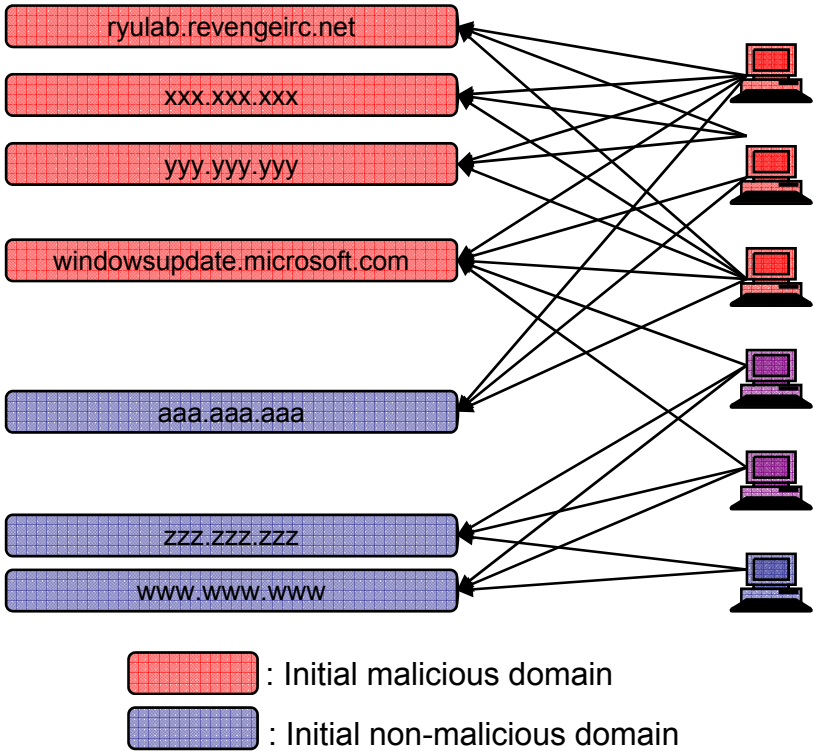
Problems with Black Domain Name List

- Precision (False positive)
  - Some botnets access popular domain names (e.g., www.google.com) to check Internet connectivity
  - Cause misdetection of uninfected host as botnet host (false positives) and should be removed from black domain name list
- Recall (False negative)
  - There may be domain names that are involved with botnet activities but not monitored in honeypot

Use of Query Graph

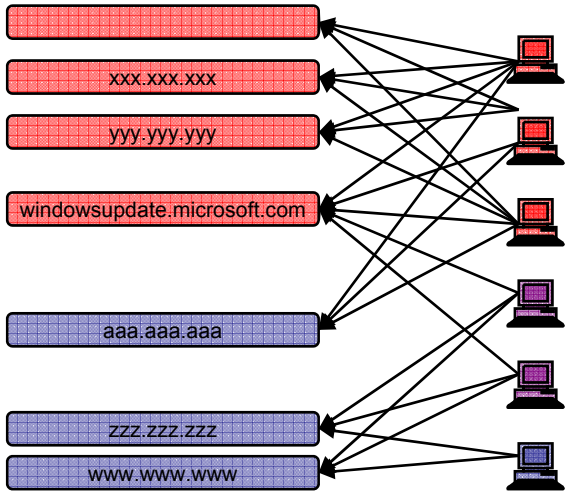
- Query Graph: {queried domain names ↔ quering hosts} bipartite graph
  - Domain names resolved by hosts that resolve black domain names are also expected to be black.
  - Domain names resolved by many hosts that do not resolve black domain names are expected to be white.

- Graph structure indicates us which domain names might be black/white.
  - How can we use the query graph structure?



Graph Kernels

- Graph kernel: gives similarities between nodes in graph
- Developed in machine learning research field[Kondor02, Kandola03, Ito05]
  - Calculate similarities between words using (similar) documents have the both words in common.
  - Calculate similarities between books using customers that those books bought in common.
- In terms of query graph: nodes (domain names) that are similar to nodes (black domains) are also expected to be black, and vice versa.



[Kodor02] R. I. Kondor, “Diffusion Kernels on Graphs and Other Discrete Input Spaces,” *ICML* 2002.
[Kandola03] J. Kandola, et.al., “Learning semantic similarity. *NIPS, 2003.*
[Ito05] T. Ito, et.al., “Application of kernels to link analysis,“ *ACM KDD, 2005*.

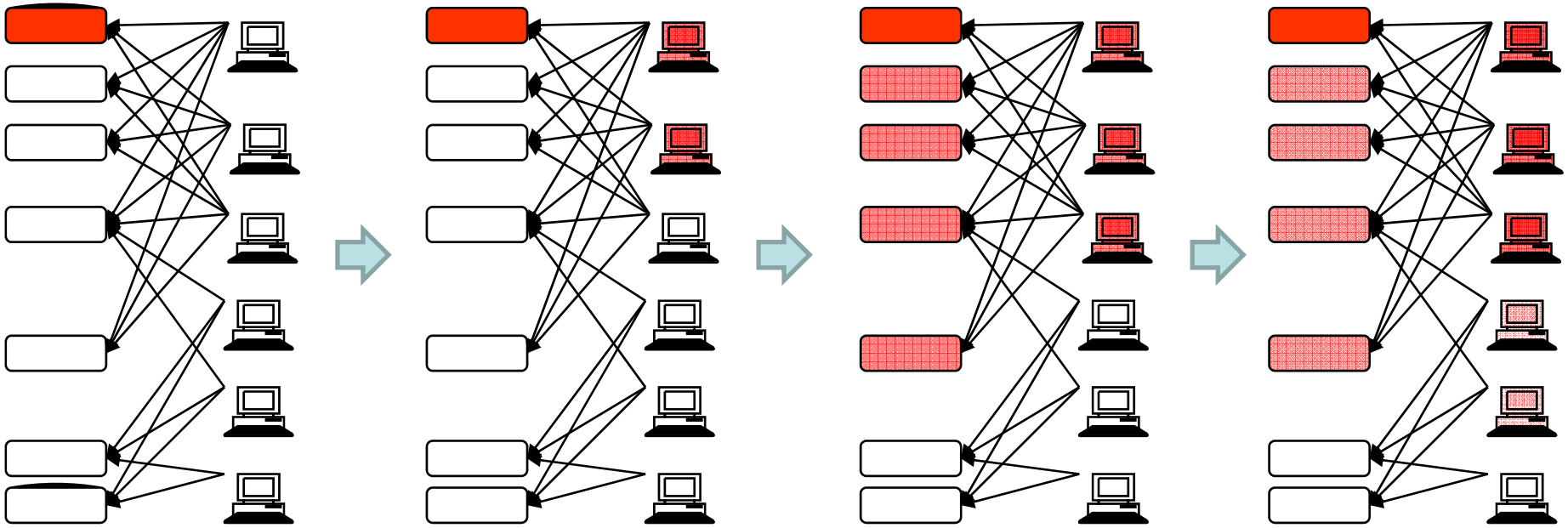
Graph Kernels (cont'd)

- Normalized Neumann Kernel
  - A = $\{a_{ij}\}$: adjacency matrix (a_{ij}=1,if node i and j are connected. i.e. user "i" send query of domain name "j")
  - D: diagonal matrix whose element is sum of rows of A (node degrees)
  - N:=A*D^{-1} :transition probability matrix of discrete-time Markov chain on the graph.)
  - **$K_{NN}(\beta) = N(1-bN)^{-1}$**= $N+\beta N^{2}+ \beta^{2}N^{3} + \cdots$
  - (i,j)-element of N^n: probability that a random walk starting node i is on node j at step n.
  - (i,j)-element of $K_{NN}(b)$: Sum of above probabilities with decay rate β (n step-forward probability weighted with β^n)

- Black score b(j) of domain name j is calculated using (i,j)-element of $K_{nn}(\beta)$, $k_{ij,}$ as:

$$b(j) = \sum_{i:\text{black}} k_{ij}$$

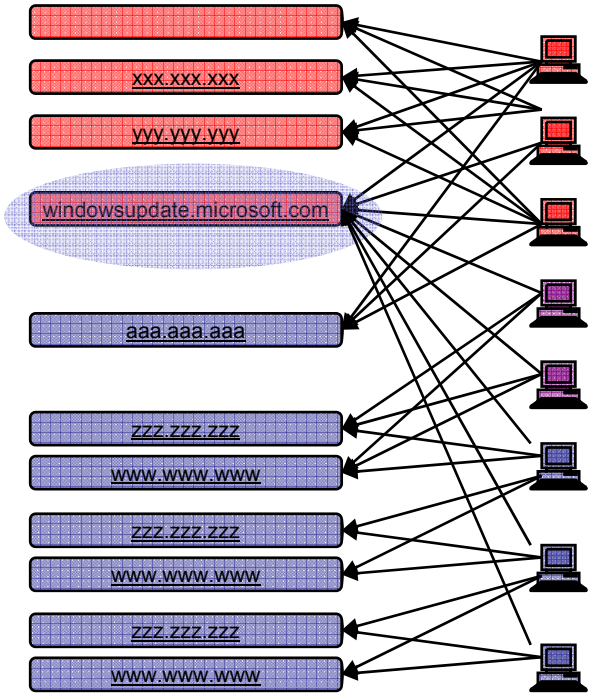
Graph Kernels (cont'd)



n=1 n=2 n=3 n=4

Score Ratio

- By using graph kernel, we can find domain names that are similar to the black domain names but not found in honeypot
  - May improve recall (false negative) ratio
- However, popular domain names that are queried by botnet hosts still have high black score
  - Problem remains in terms of precision (false positive)
- Those popular domain names might have also high similarities among white domain names as well.
- Calculate the white score in addition to black score, and take their ratio



$$b(j) = \sum_{i:\text{black}} k_{ij} \qquad w(j) = \sum_{i:\text{white}} k_{ij} \qquad \Rightarrow \qquad \frac{b(j)}{w(j) + b(j)}$$

Random Walk Sampling

- Calculating graph kernel requires operating adjacency matrix of graph, but it is difficult when applied to a huge graph.

- Construct a random walk sampling to estimate the graph kernel
- Consider a random walk starting at node i with stop probability 1-β
- Normalized Neumann kernel can be estimated as follows:
  - Weight β is automatically applied because long walks have occurred with stop probability 1-β.

$$\widehat{b(j)} = \sum_{i:\text{black}} \frac{1}{K} \sum_{k=1}^{K} \#\text{ of appearace of node } j \text{ in } k\text{-th random walk starting at node } i$$

Preliminary experimental result (1/2)

- Random walk simulation
  - # of walks:10000
  - Stop probability: 0.9
- Ranking of domain names that is initially listed as black domain name
- Successfully extract domain names that are not resolved only by botnets
  - wpad, Google, and Windows, for example.

Ranking	DomainNames	ScoreRatio	Black Score	White Score
1	fix.em3die.com	1.00	1.11	0.00
2	bunghole.mysqld.com	1.00	1.11	0.00
3	fuck.urpal43sourpalhuh.com	1.00	1.08	0.00
4	irc.botstealer.be	1.00	1.06	0.00
5	ypgw.wallloan.com	1.00	1.05	0.00
6	tap.aktash123.com	1.00	1.05	0.00
7	ro.dawnsoul.net	1.00	1.05	0.00
8	cp.dawnsoul.info	1.00	1.05	0.00
9	xt.nad123nad.com	1.00	1.04	0.00
10	xt.nadsamcabran12.com	1.00	1.04	0.00
(snip)				
276	windowsupdate.microsoft.com	0.13	1.02	6.68
277	www.yahoo.com	0.11	1.01	8.23
278	crl.verisign.com	0.10	1.02	9.01
279	google.com	0.10	1.01	9.24
280	csc3-2004-crl.verisign.com	0.09	1.01	10.65
281	download.microsoft.com	0.09	1.01	10.82
282	checkip.dyndns.org	0.06	1.07	17.87
283	crl.microsoft.com	0.05	1.01	19.73
284	www.microsoft.com	0.02	1.18	64.90
285	time.windows.com	0.01	1.17	93.47
286	download.windowsupdate.com	0.01	1.06	153.73
287	www.google.com	0.00	1.10	232.46
288	wpad	0.00	1.26	558.33

Preliminary experimental result (2/2)

- Ranking of domain names that is not initially listed as black domain name
- Many of them seems to be domain names involved with malware activities
- There is room to improve the method because score ratio is not so high.

Ranking	DomainNames	Score Ratio	Black Score	White Score
1	nadsam0.info	0.07	0.09	1.20
2	proxim.ntkrnlpa.info	0.05	0.06	1.11
3	qualitydrug-store.com	0.04	0.04	1.01
4	fbi32.cheapdf.com	0.04	0.04	1.00
5	d.Felony-Productions.net	0.04	0.04	1.04
6	ksacoool.com	0.03	0.04	1.06
7	tassweq.com	0.03	0.04	1.08
8	pop.comcom2.com	0.03	0.03	1.04
9	xx.xx.xxx.xx.in-addr.arpa	0.03	0.03	1.05
10	jns.dns02.com.ar	0.02	0.03	1.15
11	smtp3.google.com	0.02	0.04	1.47
12	smtp2.google.com	0.02	0.04	1.48
13	jns.dns01.com.ar	0.02	0.03	1.16
14	smtp1.google.com	0.02	0.04	1.73
15	ovbijkbqgr.hn.org	0.02	0.02	1.09
16	oegona.hn.org	0.02	0.02	1.12
17	webipcha.cn	0.02	0.02	1.03
18	fumegb.yi.org	0.02	0.02	1.01
19	nnkpuji.hn.org	0.02	0.02	1.10
20	ufyimtttqwh.afraid.org	0.02	0.02	1.02

Conclusion

- Summary
  - We proposed a method to improve quality of black domain-name list for botnet detection.
  - Reduced false positives and negatives of detected black domain names
- Future works
  - Adopt other kernels with various parameters and compare them
  - Physical meaning of kernel should be investigated.

Thank you

Appendix: Other Graph Kernel

- Diffusion Kernel
  - A = $\{a_{ij}\}$: adjacency matrix (a_{ij}=1,if node i and j are connected. i.e. user "i" send query of domain name "j")
  - D: diagonal matrix whose element is sum of rows of A (node degrees)
  - L := D-A: transition rate matrix of continuous-time Markov chain on the graph)

  - **$K_D(\tau) = \exp(\tau L)$**
  - Heat diffusion model used to represent similarities among nodes
  - (i,j) – element of diffusion kernel $K_D(\tau)$ is the heat of node "j" after time τ when initial heat is given to "i".

Appendix: cross validation

- Partition the blacklist into n peaces (say, ten)
- Using nine peaces, estimate the other one piece
  - How domain names in the other one piece can be revealed by other nine piecces.
- Used in evaluations of the machine learning technique
- Seek optimal parameters