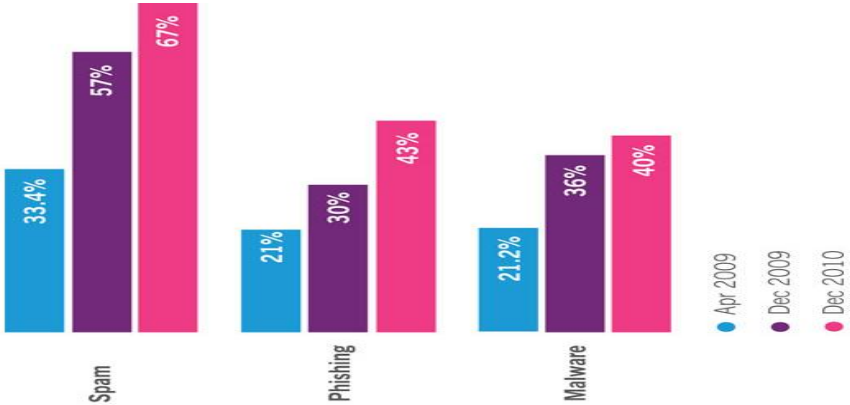
DETECTING MALICIOUS CLIENTS IN ISP NETWORKS USING HTTP CONNECTIVITY GRAPH AND FLOW INFORMATION

Lei Liu[1] Sabyasachi (Saby) Saha[2] Ruben Torres[2] Jianpeng Xu[1] Pang-Ning Tan[1] Antonio Nucci[2] Marco Mellia[3]

[1]Michigan State University, Michigan, USA
[2]Narus, Inc., Sunnyvale, California, USA
[3] Politecnico di Torino, Italy

© 2013 Narus, Inc.

Introduction



- Malware is …
  - Malicious software
  - Virus, Phishing, Spam, …
- Increasing threats
  - 4500 new Web attacks launched per day (Symantec Security Report)
  - Continuous and increased attacks on infrastructure
  - Threats to business, national security
    - Huge financial stake (Conficker: 10 million machines, loss $9.1 Billion)
    - Zeus: 3.6 million machines [HTML Injection]
    - Koobface: 2.9 million machines [Social Networking Sites]
    - TidServ: 1.5 million machines [Email spam attachment]
- Attacks are becoming more advanced and sophisticated!

Introduction

- Limitation of existing techniques
  - Signature-based approach
    - Fails to detect zero-day attacks.
    - Fails to detect threats with evolving capabilities such as metamorphic and polymorphic malwares.
  - Anomaly-based approach
    - Producing high false alarm rate.
  - Supervised Learning based approach
    - Poor performance on novel malware

There is no Silver Bullet

© 2013 Narus, Inc.

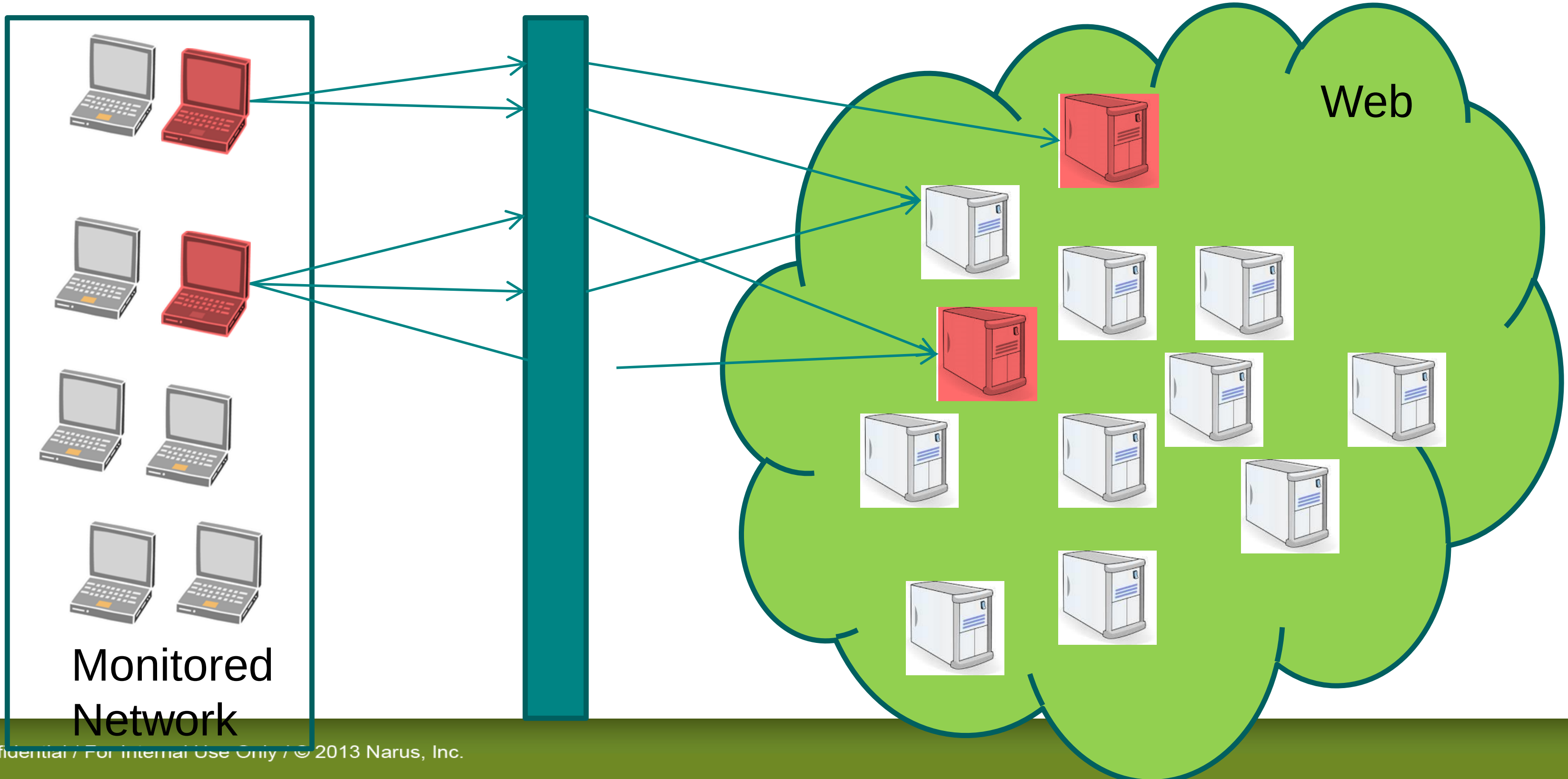
Introduction

- However, the malware cannot hide the communication
  - We know who talks to whom (Connectivity Graph)
  - We can extract some information about what has been communicated

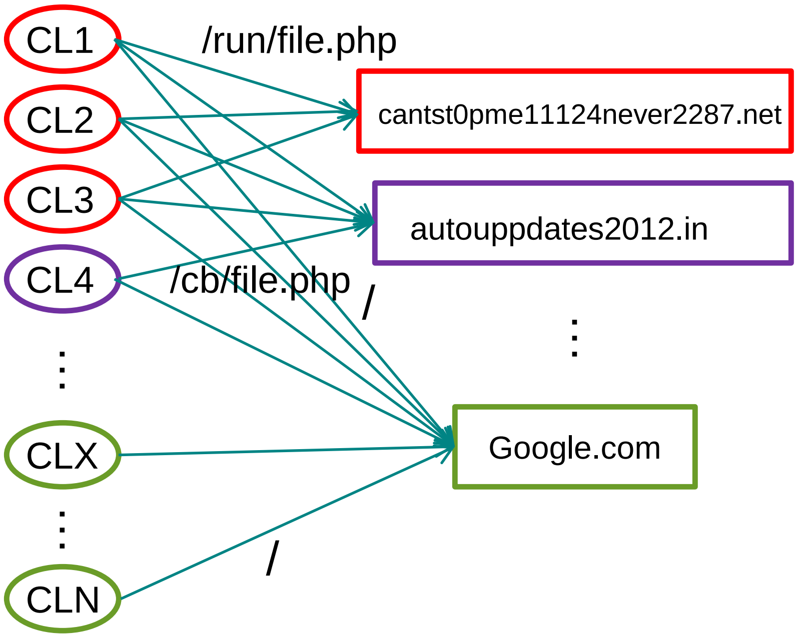
- Goal: Augment current security solutions using connectivity graph and flow information to find hidden malicious nodes

HTTP Connectivity

- We focus on HTTP Traffic
  - Most of the malwares are in HTTP
    - Difficult to block



One Example



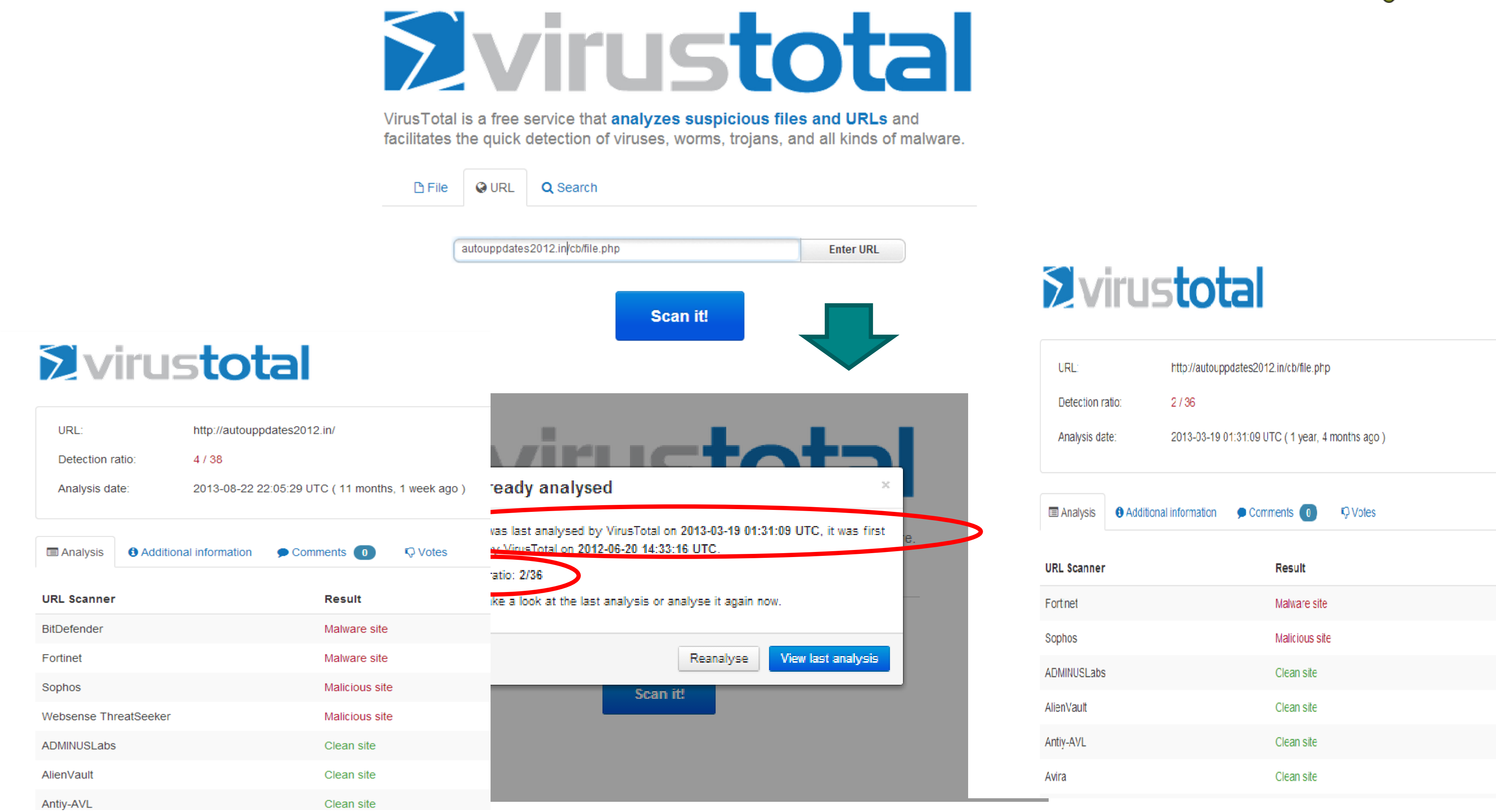
- CL1|cantst0pme11124never2287.net|/run/file.php
- CL1|google.com|/
- CL1|autouppdates2012.in|/cb/file.php
- CL2|cantst0pme11124never2287.net|/run/file.php
- CL2|google.com|/
- CL2|autouppdates2012.in|/cb/file.php
- CL3|cantst0pme11124never2287.net|/run/file.php
- CL3|google.com|/
- CL3|autouppdates2012.in|/cb/file.php
- CL4|google.com|/
- CL4|autouppdates2012.in|/cb/file.php
- CL5|google.com|/
- CLX|google.com|/
- CLN|google.com|/

Our Approach

- We propose a two step malicious score propagation approach to identify other malicious nodes in the network
  - Initialize with the malicious nodes having a non-zero score and others having a zero score

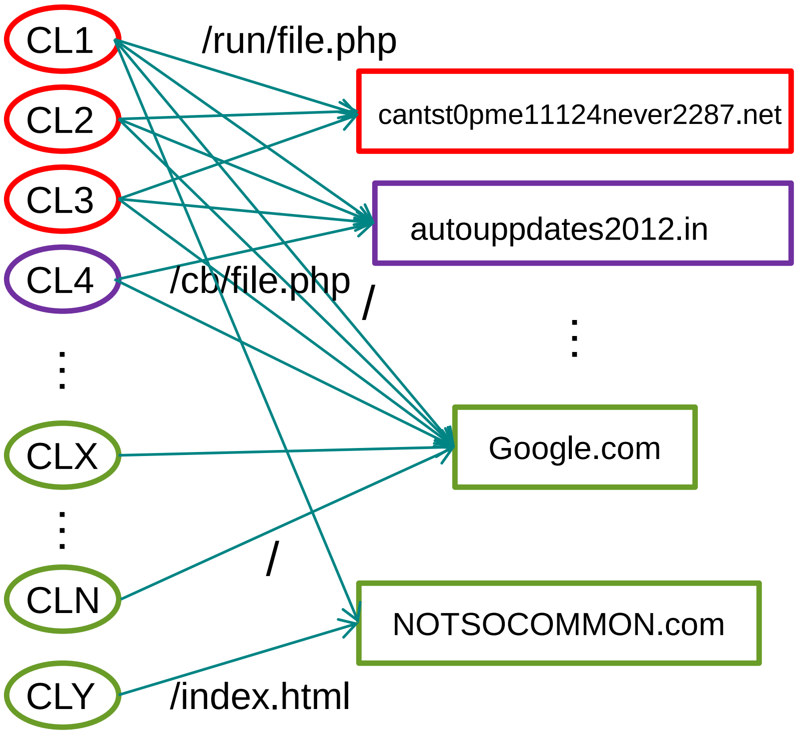
Confidential / For Internal Use Only / © 2013 Narus, Inc.

NARUS


virustotal
VirusTotal is a free service that analyzes suspicious files and URLs and facilitates the quick detection of viruses, worms, trojans, and all kinds of malware.
File
URL
Search
autouppdates2012.in/cb/file.php
Enter URL
Scan it!
virustotal
URL: http://autouppdates2012.in/
Detection ratio: 4 / 38
Analysis date: 2013-08-22 22:05:29 UTC (11 months, 1 week ago)
Analysis
Additional information
Comments 0
Votes
URL Scanner
Result
BitDefender
Malware site
Fortinet
Malware site
Sophos
Malicious site
Websense ThreatSeeker
Malicious site
ADMINUSLabs
Clean site
AlienVault
Clean site
Antiy-AVL
Clean site
ready analysed
was last analysed by VirusTotal on 2013-03-19 01:31:09 UTC, it was first
by VirusTotal on 2012-06-20 14:33:16 UTC.
ratio: 2/36
ke a look at the last analysis or analyse it again now.
Reanalyse
View last analysis
Scan it!
virustotal
URL: http://autouppdates2012.in/cb/file.php
Detection ratio: 2 / 36
Analysis date: 2013-03-19 01:31:09 UTC (1 year, 4 months ago)
Analysis
Additional information
Comments 0
Votes
URL Scanner
Result
Fortinet
Malware site
Sophos
Malicious site
ADMINUSLabs
Clean site
AlienVault
Clean site
Antiy-AVL
Clean site
Avira
Clean site

Example Contd.

NARUS



- CL1|cantst0pme11124never2287.net|/run/file.php
- CL1|google.com|/
- CL1|autouppdates2012.in|/cb/file.php
- CL2|cantst0pme11124never2287.net|/run/file.php
- CL2|google.com|/
- CL2|autouppdates2012.in|/cb/file.php
- CL3|cantst0pme11124never2287.net|/run/file.php
- CL3|google.com|/
- CL3|autouppdates2012.in|/cb/file.php
- CL4|google.com|/
- CL4|autouppdates2012.in|/cb/file.php
- CL5|google.com|/
- CLX|google.com|/
- CLN|google.com|/

- CL1|NOTSOCOMMON.com|index.html
- CLY|NOTSOCOMMON.com|index.html

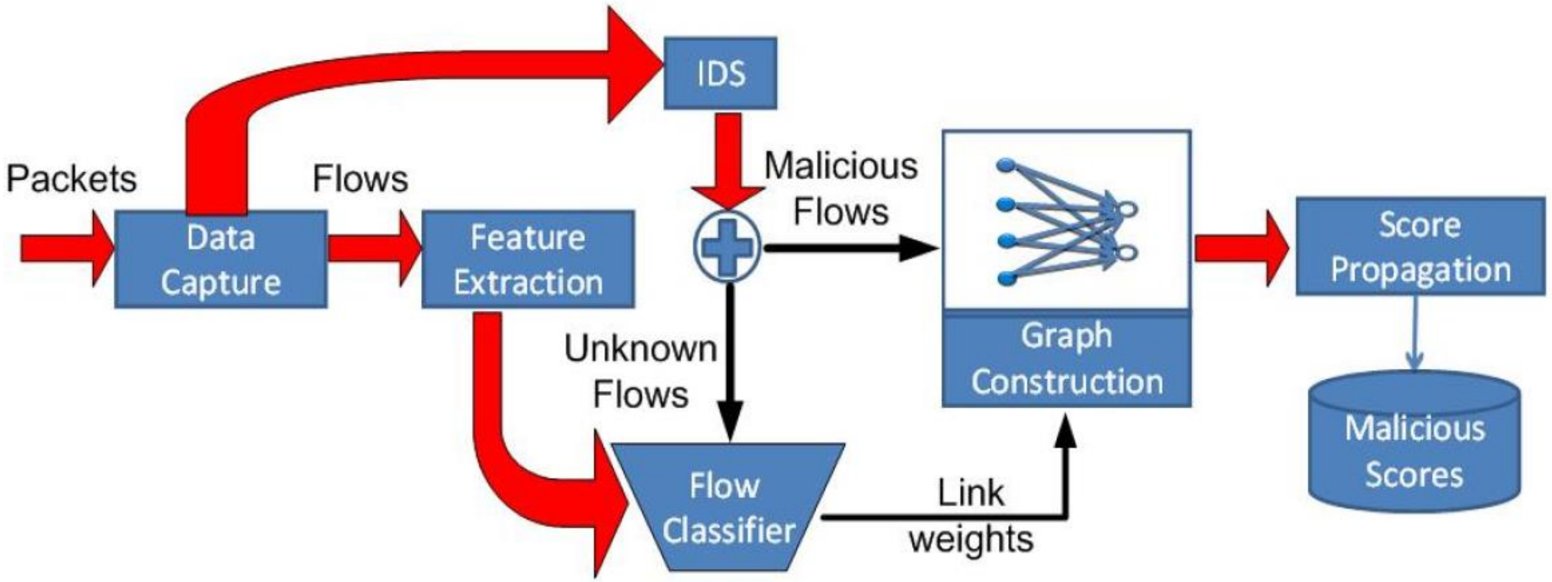
Our Approach

- We weigh the edges of the graph using malicious flow similarity
  - If the flow through the edge has any similarity with the malicious flows in the data
  - Used a SVM based classifier with 270 flow-based features

- Then use the same two step malicious score propagation approach to identify other malicious nodes in the network

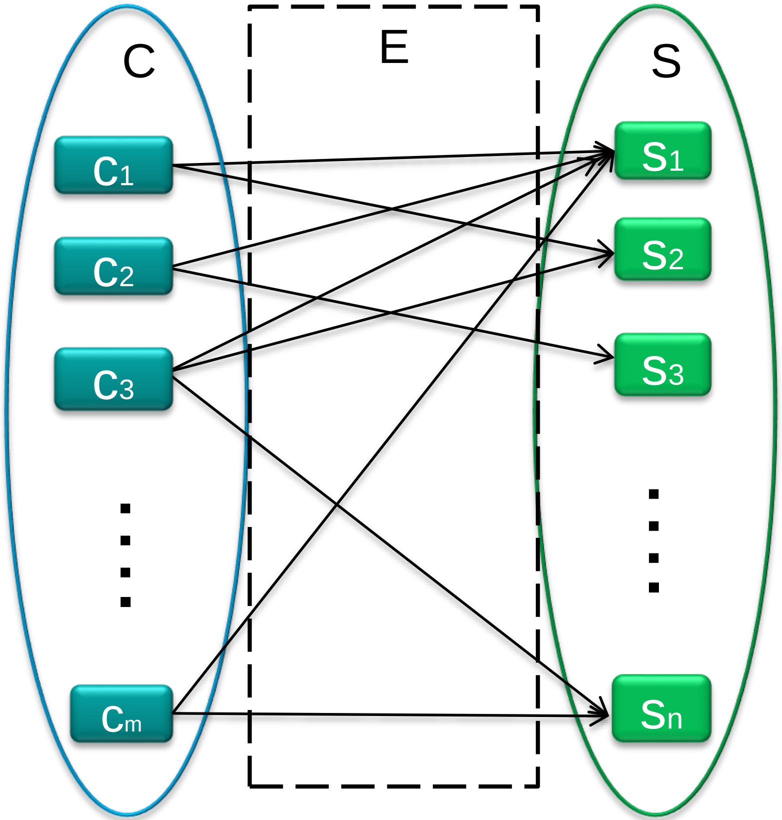
Confidential / For Internal Use Only / © 2013 Narus, Inc.

System Architecture



Confidential / For Internal Use Only / © 2013 Narus, Inc.

HTTP Graph Construction



- HTTP bipartite graph
  - G = ((C,S),E) denote a directed graph constructed from the HTTP connections
  - C, set of client IP addresses
  - S, set of server IP addresses
  - E, set of directed links

Two-phase Alternating Score propagation

- Objective Function

$$Q(\mathbf{y}) = \frac{1}{2}\sum_{ij} W_{ij}\left[\frac{y_i}{\sqrt{D_{ii}}} - \frac{y_j}{\sqrt{D_{jj}}}\right]^2 + \frac{\mu}{2}\sum_{i}(y_i - y_i^{(0)})^2$$

Where **W** is a adjacency matrix, and **D** is a diagonal matrix whose diagonal elements are given by $D_{ii} = \sum_j W_{ij}$

- Our objective can be reduced to:

$$x_i = (1-\beta_s)x_i^0 + \beta_s \sum_{k \in C} w_{ki}^{cs} y_k \quad (1)$$

$$y_k = (1-\beta_c)y_k^0 + \beta_c \sum_{j \in S} w_{jk}^{sc} x_j \quad (2)$$

- Iteratively updating the malicious score based on its initial value and weighted average of scores for its neighbors until convergence

Two-phase Alternating Score propagation

- **Link-only:** the weight of a link depends on the existence of a flow between the node pair

$$w_{ij}^{(l)} = \begin{cases} 1, & \text{if } (v_i, v_j) \in \mathcal{E} \text{ (or } \pi_{ij} \neq \emptyset); \\ 0, & \text{otherwise.} \end{cases}$$

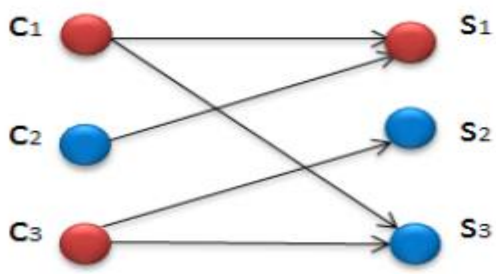
  – Normalizing the link weight by its out-degree:

$$\hat{\mathbf{W}}_{ij} = \frac{w_{ij}}{\sum_j w_{ij}}$$

Confidential / For Internal Use Only / © 2013 Narus, Inc.

Two-phase Alternating Score propagation

- Example



Transition matrix

$$W^{cs} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad W^{sc} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

Score propagation

node	initial score	iteration 1	iteration 2	iteration 3
c_1	1	0.7556	0.7230	0.7220
c_2	0	0.2444	0.2981	0.3222
c_3	1	0.8725	0.8485	0.8253
s_1	1	0.575	0.6165	0.6529
s_2	0	0.425	0.4346	0.4258
s_3	0	0.85	0.8195	0.7908

HTTP Flow Features

NARUS

SessionID|SourceIP|DestIP|1314281856|example.com|mozilla/2.0|||||0|1|1|0|63973|80|477|1628|189|1460|7|4|1|/blog/images/3521.jpg||GET|tq=RA1DQxZBDVlUFQN0AQUDAh|57|200 OK|image/jpeg|190984||

- Hostname
  - Length of the second domain
  - Randomness of the second domain
  - Is it a IP address?
  - Reliability score from .com, .info, etc.

  Etc.
- User agent
- URI
  - Keywords separated by delimiters
  - Length of the URI
  - # of fields
- Referrer
- Method (GET, POST)

- Additional parameters
  - -Length of the first key, value (categorical)
  - - (For each key-value pair) characters, numbers or mix
- Server status
- Content type
- Error on request
- Number of requests
- Number of URLs requested in a session
- Number of pages requested in a session
- Bytes Sent
- Bytes Received
- Data sent, Data received, Pkts sent, Pkt rcvd
- Response time
- Bytes transferred
- Content length

Flow Classification

- Classifying from malicious and unknown flows

$$\begin{aligned} \min_{\omega,b} \quad & \frac{1}{2}\omega^T\omega + C^l\sum_{i=1}^{k-1}\xi_{i^l} + C^u\sum_{j=k}^{m}\xi_{j^u} \\ \text{subject to} \quad & y_{i^l}(\omega^T\phi(x_{i^l})+b) \geq 1-\xi_{i^l} \\ & y_{j^u}(\omega^T\phi(x_{j^u})+b) \geq 1-\xi_{j^u} \\ & \xi_{i^l} \geq 0, i=1,2,\cdots,k-1 \\ & \xi_{j^u} \geq 0, j=k,k+1,\cdots,m \end{aligned}$$

Cost for misclassifying malicious flows

Cost for misclassifying unknown flows

in which, assigning $C^l > C^u$ could guide the classifier towards classifying more accurately flows that belong to malicious class.

Flow-based Graph Construction

- **Combining Flow information and Link structure:**
  - The weight of a link is determined from its malicious score:

$$w_{ij}^{(f)} = \begin{cases} 1, & \text{if } \exists f_{ijk} \in \pi_{ij} : \mathcal{I}(f_{ijk}) = 1; \\ \max\limits_{\sigma_k \in \Sigma_{ij}} \{\sigma_k\}, & \text{otherwise.} \end{cases}$$

- Link is associated with set of flows $\pi_{ij} = \{f_{ij1}, f_{ij2}, \cdots, f_{ij|\pi_{ij}|}\}$

$\Sigma_{ij} = \{\sigma_1, \sigma_2, \cdots, \sigma_{|\pi_{ij}|}\}$ corresponding outputs of flow classifier

- If the malicious score for the link from IDS is 1, we set the weight of the edge as 1
- Otherwise, we set it to the maximum value of the flow classification

Experimental Settings

- Data Sets
  - Four 1-hour data sets, which we name D1~D4, more detailed information in following table.

DATASETS D1 $\sim$ D4

Data set	Time of day	Number of HTTP flows	Number of labeled flows	Number of malicious clients/ Total number of clients	Number of malicious servers/ Total number of servers
D1	4 pm	1926620	1088	26/5856	25/25627
D2	6 pm	973271	1649	30/6533	25/26346
D3	7 pm	1033292	1736	26/7360	22/26216
D4	9 pm	1078765	65	21/7936	23/27518

  - Used a commercial IDS to identify flag malicious flows

Experimental Evaluation

- Validation
  - To validate a predicted client, which receives a high propagation score, we check if this client connects to any malicious web server
  - To validate the web server
    - Google SafeBrowsing, Malware Blacklists
    - WOT (Web of Trust) score

Experimental Evaluation

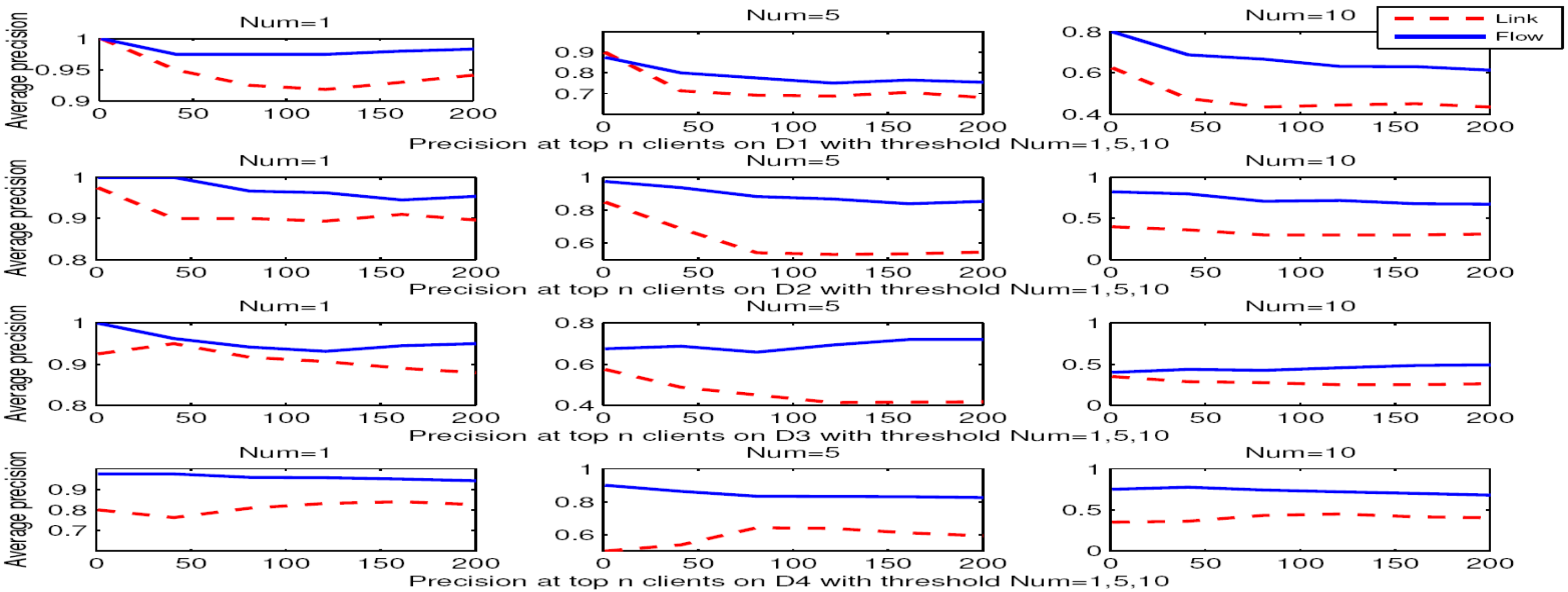
- Metric
  - The precision of top n ranking clients is used to indicate the final performance.

$$P@n = \frac{num(TM)}{n}$$

    - *Num(TM)* is the number of true malicious clients show up in top n clients. We report the precision from p@1 to p@1000.

Results

- Results Comparison for All Clients

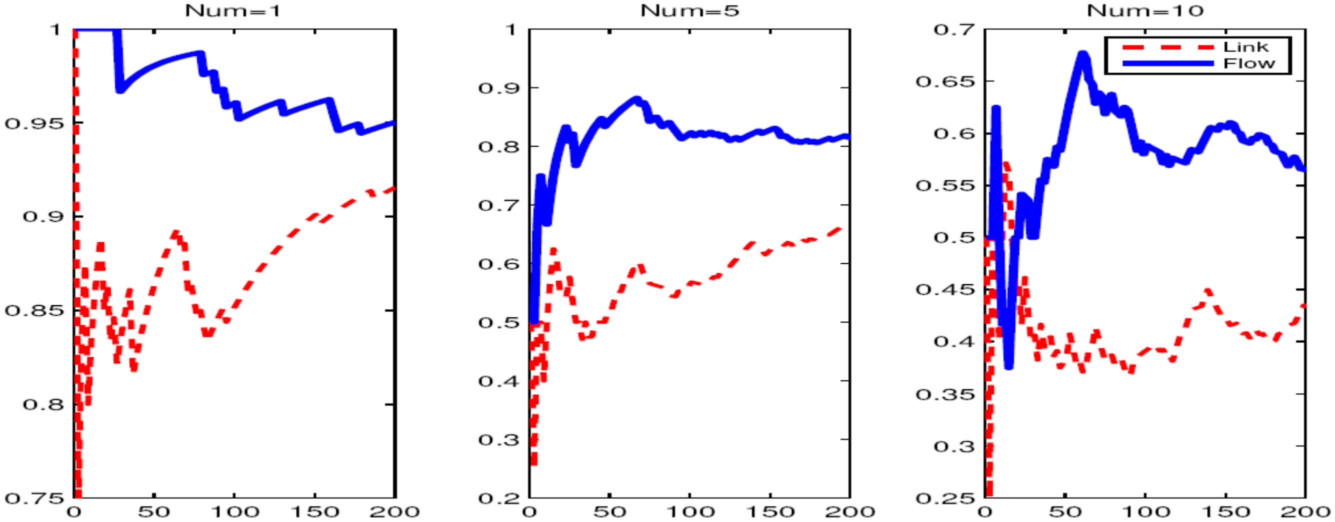


Precision at top n clients on D1 ~ D4 with threshold Num = 1,5,10

Confidential / For Internal Use Only / © 2013 Narus, Inc.

Results

- Results of Clients that are indirectly Connect to Malicious nodes



Precision at top n highest ranked clients that are indirectly connected to malicious hosts

Conclusion

- Proposed a method that combine the links and flow-level information in the HTTP communication graph for malicious clients detection
- Proposed an efficient two-phase score propagation algorithm to identify malicious clients
- Experimental results on large ISP data verified that our proposed method could detect clients that infected with known/new malwares

Confidential / For Internal Use Only / © 2013 Narus, Inc.

Future Work

- Extent the framework beyond HTTP traffic
  - Flow features will change
- To be added

Thank You

Future Work

- Validate with diverse data sets
- Propagating malicious score for server by treating each URL as a node instead of server IP.
- Consider the hidden connections in URL-URL connectivity graph.