

Multidimensional Traffic Clustering for Digesting, Visualization, Anomaly Detection, and Signature Extraction

J. Wang, D.J. Miller, G. Kesidis
EE & CSE Depts., Penn State



Introduction (1)

➤ Multidimensional Flow Mining:

Frequent item set mining applied to network traffic flows, based on the packet header 5-tuple (source IP, destination IP, source port, destination port, protocol)

➤ Applications

- Data digesting: identify dominant flows sent over a given link, over a specified time window
- Data visualization: identify flow definitions of interest, for subsequent (e.g., time series) visualization

Introduction (2)

➤ Applications

- Anomaly detection: identify dominant flows and classify them as normal/abnormal
 - precise specification of abnormality, for traffic blocking
 - sensible front end for payload-based signature extraction

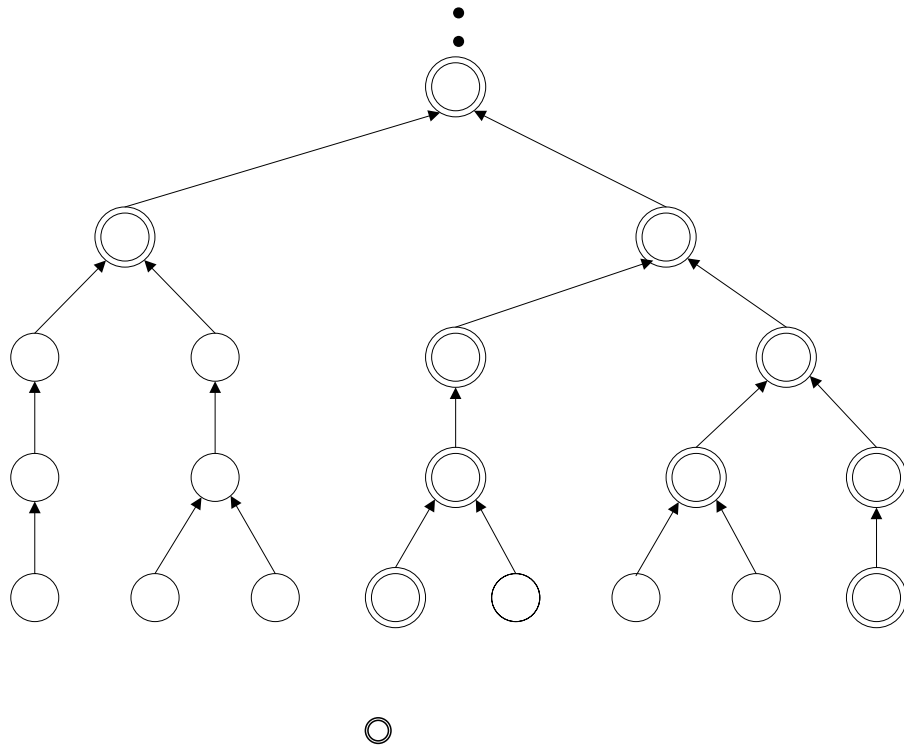
➤ Hierarchy Multidimensional Mining

- captures traffic at different scales
- captures hierarchical nature of IP addresses
- useful for computationally efficient mining

Unidimensional Clustering (1)

- Bottom-Up Tree Construction
- Only Save Significant Nodes in Each Dimension
- Five Tuples (Dimensions):
 - Source IP dimension: binary tree, 32 levels
 - Destination IP dimension: binary tree, 32 levels
 - Source port dimension: 3 levels (low port: < 1024; high port: > 1023)
 - Destination port dimension: 3 levels
 - Protocol dimension: 2 levels (leaf, top)

Unidimensional Clustering (2)



(Source IP Addresses from New Zealand Trace Data)

Clustering Report

Dominant Cluster	Bytes
10.0.0.8/32	961
10.0.0.15/32	576
10.0.0.8/31	1001
10.0.0.12/31	120
10.0.0.14/31	576
10.0.0.8/30	1001
10.0.0.12/30	696
10.0.0.0/29	128
10.0.0.8/29	1697
10.0.0.0/28	1825

Multidimensional Clustering (1)

➤ Top-Down Tree Construction

Only generate children for significant nodes, and stop when there are no more significant nodes generated.

➤ Two Criteria for Judging a Significant Node

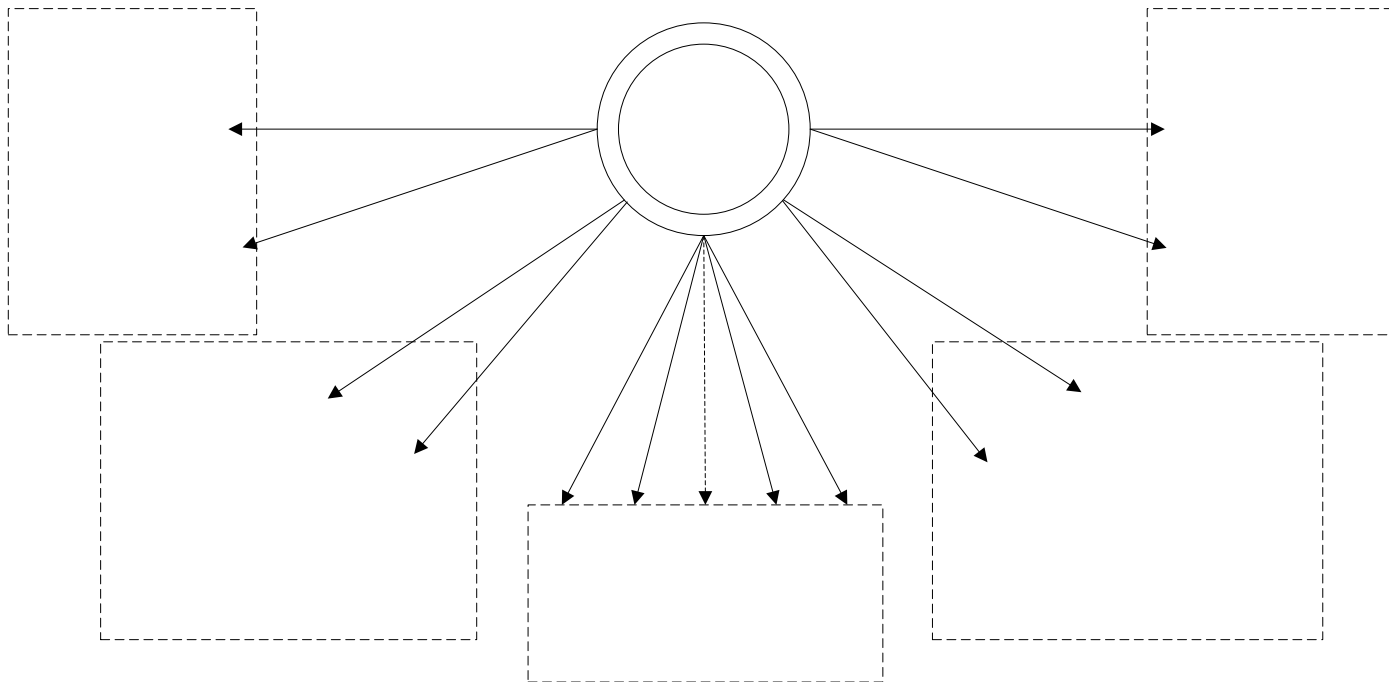
1). All of the node's unidimensional ancestors must be significant

2). All of the node's multidimensional parents must be significant

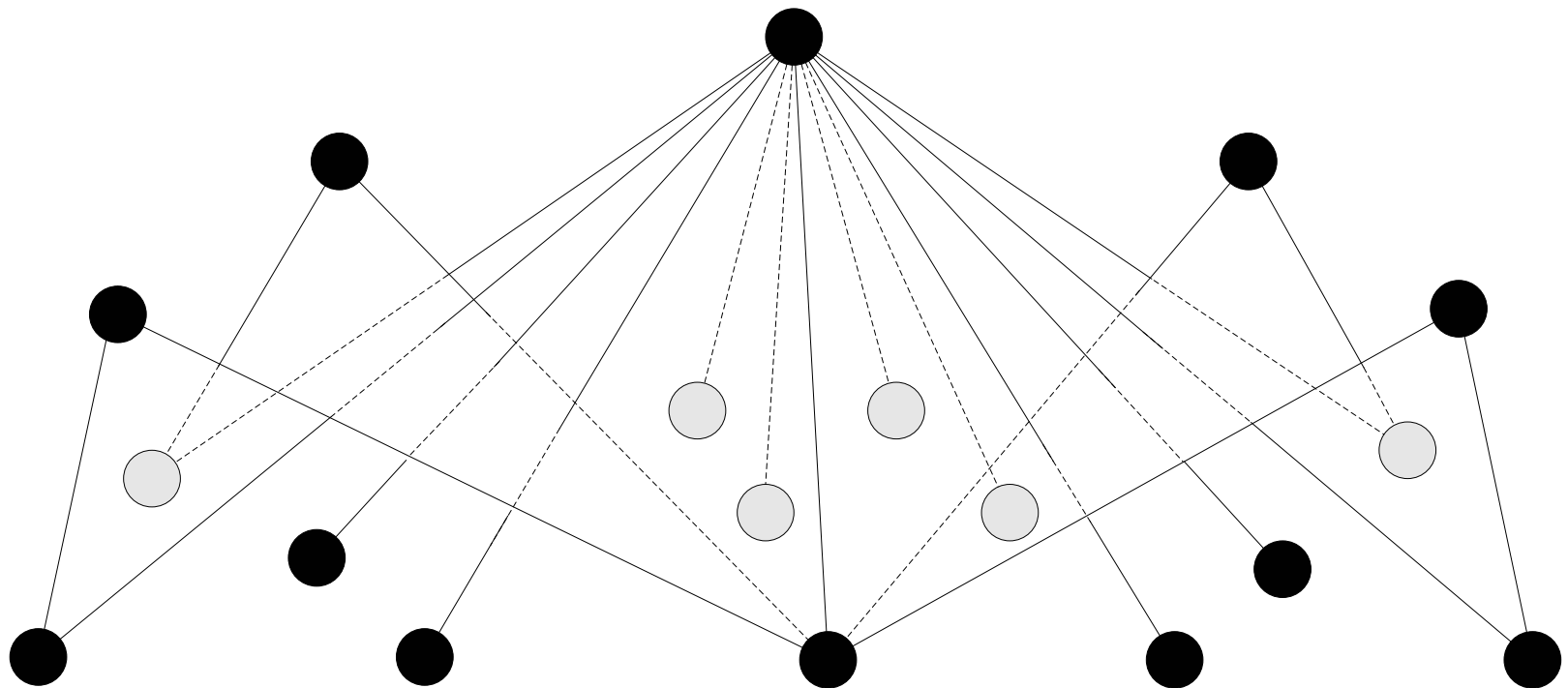
➤ Packet Matching and Counting Operation

Only for the nodes which satisfy two criteria above

Multidimensional Clustering (2)



Our Improvements (1)



Our Improvements (2)

➤ Flow Subset Paradigm:

To perform flow counting for a node at the child level, we can perform the matching operation using the flow subset table of the node's parent, rather than the entire Netflow table.

➤ Minimum Parent Strategy:

Use the parent (there may be at most five parents for each child node) with minimum subset size to count the size of a child level node.

Our Improvements (3)

➤ Complement Strategy:

Make use of the mutually exclusivity and collective exhaustion of the child nodes in the same dimension and on the same level to improve flow counting efficiency.

➤ Top-Down Unidimensional Clustering for IP Dimensions:

Avoid the possible huge memory requirement for IP mapping method.

Avoid possible complexity used to sort and generate leaf nodes for bottom-up unidimensional clustering.

Our Improvements (4)

Processing Time Comparison on New Zealand (NZIX) Traces

Trace Data Length	30 min	1 hour	2 hours	3 hours
New Method in [2]	5.922s	12.766s	23.344s	45.360s
Method in [1]	48.907s	101.329s	260.422s	354.156s
Time Ratio	8.26	7.94	8.84	7.81

➤ Threshold: 5%

Criteria for Anomaly Identification (1)

➤ Absolute IP Difference in Entropy (AIDE):

IP Entropy:
$$H_k(IP) = -\sum_l P[l] \log P[l]$$

l : IP addresses in cluster k that occurred in the current digest interval

$$P[l] = \frac{\text{number of packets (bytes) in cluster } k \text{ with IP} = l}{\text{number of packets (bytes) in cluster } k}$$

AIDE: for cluster k

$$AIDE(k) = |H_k(\text{source IP}) - H_k(\text{destination IP})|$$

Criteria for Anomaly Identification (2)

➤ Unexpectedness:

Discrepancy between the actual traffic volume of a cluster and the volume predicted on a model that assumes statistically independent attributes.

$$\text{unexpectedness}(k) = 100 \times \frac{p(k)}{\prod_{i=1}^5 p_i(k)}$$

$p(k)$: the fraction of traffic that falls in cluster k

$p_i(k)$: the fraction that matches the i^{th} attribute value of cluster k .

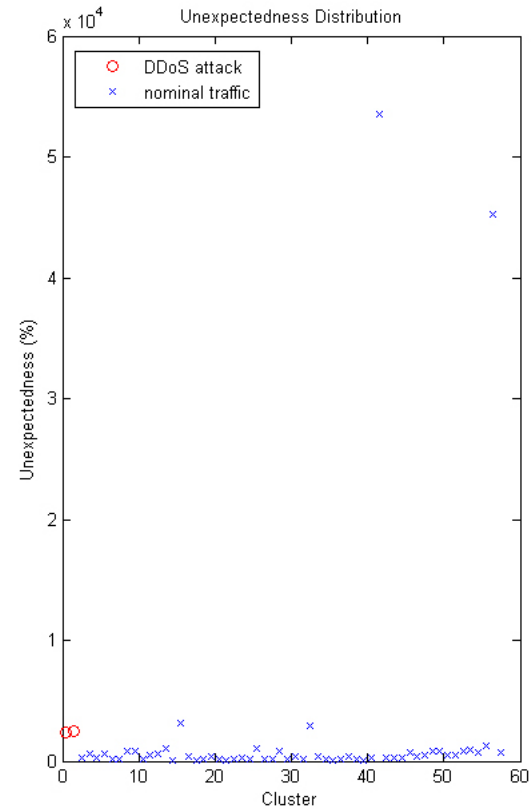
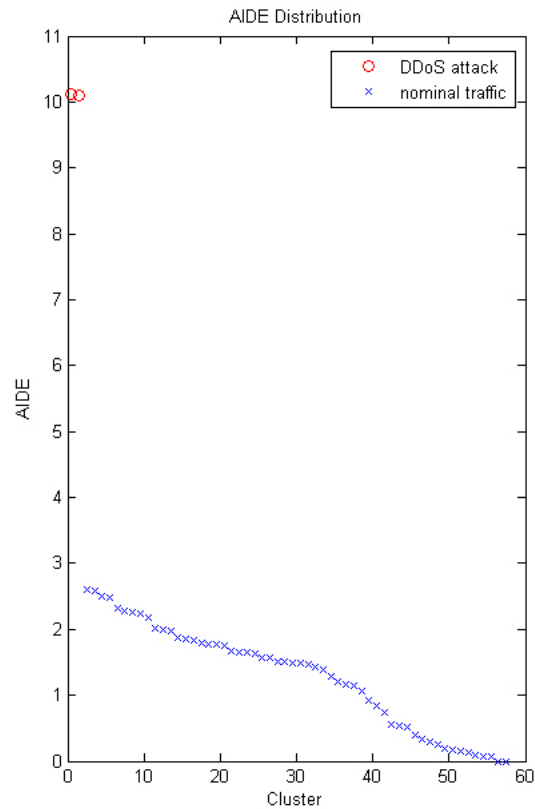
Anomaly Identification – DARPA Trace

Multidimensional Clustering Report of DARPA Trace

No	Src IP	DstIP	SrcPt	DstPt	Pr	Byte	Packet	Perc	AIDE
1	0.0.0.0/2	131.84.1.31/32	high	high	6	0.9M	23.9k	7.1%	10.11
2	64.0.0.0/2	131.84.1.31/32	high	high	6	0.9M	23.8k	7.0%	10.10
3	128.0.0.0/1	172.16.112.0/23	80	high	6	13.5M	18.6k	5.5%	2.61
4	0.0.0.0/0	172.16.116.0/23	low	high	6	10.5M	17.0k	5.0%	2.59
5	172.16.116.0/23	0.0.0.0/0	high	low	*	1.3M	17.8k	5.3%	2.50
6	128.0.0.0/1	172.16.116.0/23	low	high	*	10.1M	18.3k	5.4%	2.48
7	172.16.112.50/32	128.0.0.0/1	*	*	6	0.8M	17.1k	5.1%	2.32
8	172.16.112.50/32	128.0.0.0/2	*	*	*	0.8M	17.2k	5.1%	2.28
9	192.0.0.0/3	172.16.112.0/22	80	high	6	11.9M	18.6k	5.5%	2.26
10	172.16.112.0/22	192.0.0.0/3	high	80	6	1.7M	17.0k	5.0%	2.24

Anomaly Identification – DARPA Trace

AIDE and Unexpectedness Distribution of DARPA Trace



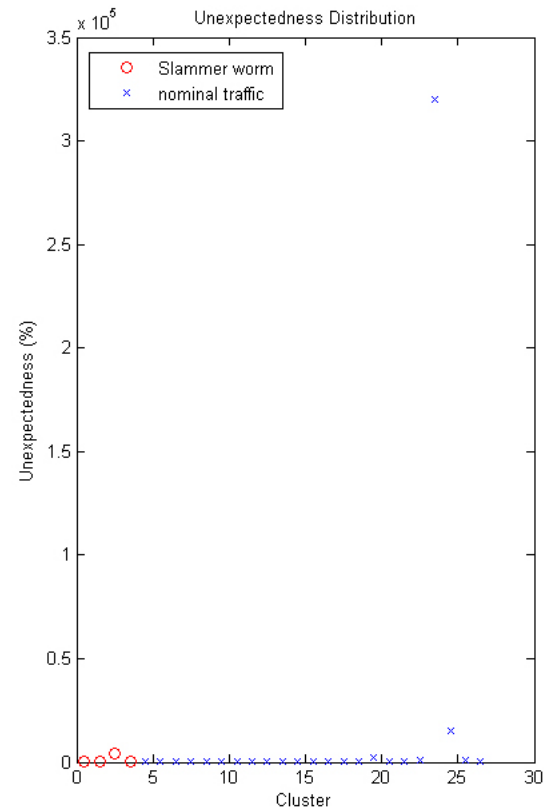
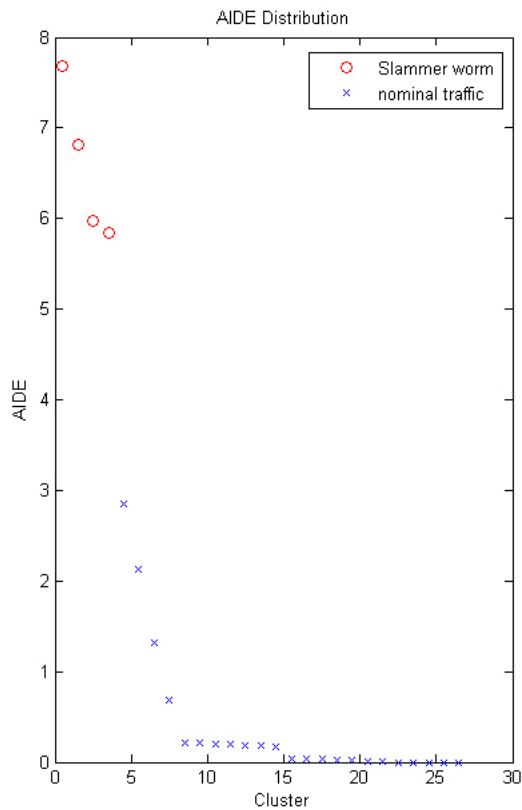
Anomaly Identification – Slammer Worm

Multidimensional Clustering Report of Slammer Worm Trace

No	Src IP	DstIP	SrcPt	DstPt	Pr	Byte	Packet	Per	AIDE
1	10.0.0.0/23	10.0.0.0/16	high	1434	17	11.9M	30.1k	7.0%	7.69
2	10.0.0.0/25	10.0.0.0/16	high	high	17	8.8M	37.1k	5.1%	6.81
3	10.0.0.0/21	10.0.128.0/17	high	1434	17	8.8M	22.2k	5.2%	5.98
4	10.0.0.0/21	10.0.0.0/17	high	1434	17	9.9M	25.2k	5.8%	5.84
5	10.0.0.0/28	10.0.0.0/17	high	high	*	9.6M	14.4k	5.6%	2.85
6	10.0.0.0/23	10.0.0.0/17	high	high	17	11.2M	70.7k	6.6%	2.13
7	10.0.0.0/27	10.0.0.0/18	high	high	*	8.8M	16.0k	5.2%	1.32
8	10.0.0.0/16	10.0.8.0/21	high	high	6	10.5M	10.8k	6.2%	0.69
9	10.0.0.0/21	10.0.0.192/29	high	high	6	8.5M	5.9k	5.0%	0.21
10	10.0.0.32/27	10.0.0.32/27	high	high	6	11.5M	30.5k	6.7%	0.21

Anomaly Identification – Slammer Worm

AIDE and Unexpectedness Distribution of Slammer Worm Trace



Anomaly Identification – Code-Red v2

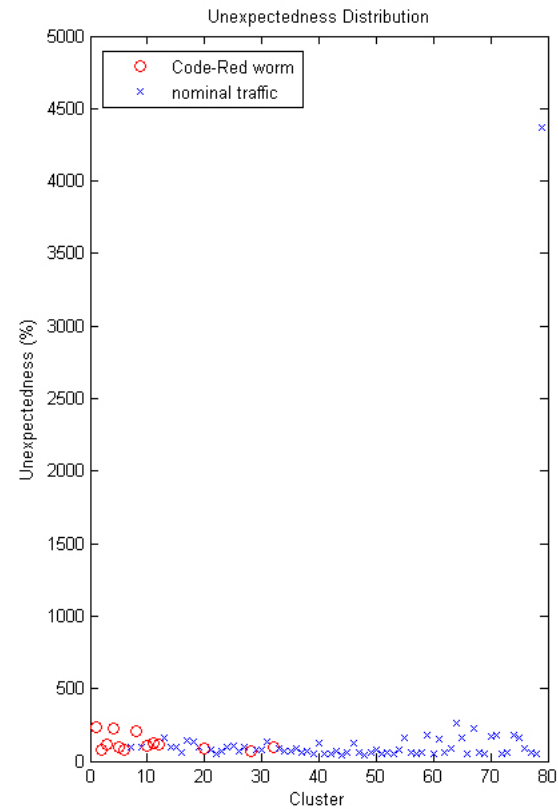
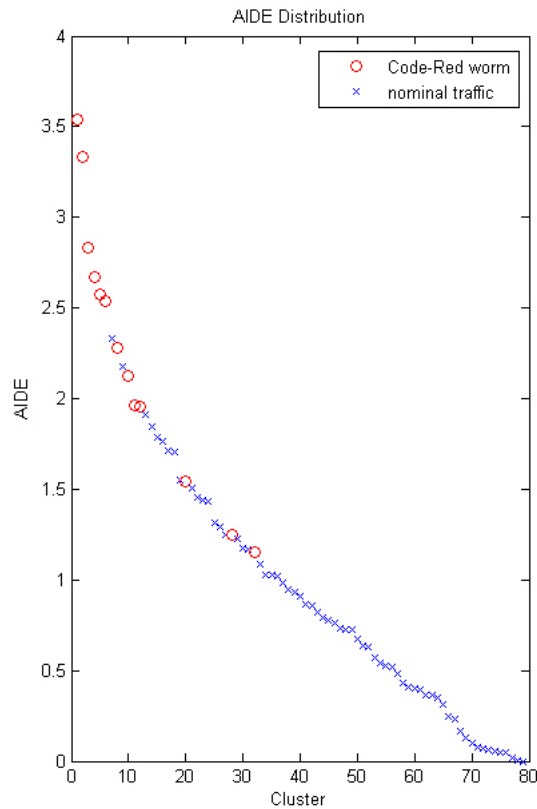
Multidimensional Clustering Report of Code-Red Worm Trace

No	Src IP	DstIP	SrcPt	DstPt	Pr	Byte	Packet	Perc	AIDE	SYNPerc
1	0.1.0.0/22	0.0.0.0/2	high	80	6	9.9M	116.4k	11.3%	3.54	53.7%
2	0.1.0.0/24	0.0.0.0/2	*	low	*	8.4M	103.9k	10.1%	3.33	46.4%
3	0.1.0.0/23	0.0.0.0/3	high	low	6	9.8M	113.1k	11.0%	2.84	42.8%
4	0.1.0.0/21	0.0.0.0/3	high	80	6	11.2M	117.1k	11.4%	2.67	44.1%
5	0.1.0.0/23	0.0.0.0/4	high	low	*	8.8M	103.8k	10.1%	2.58	39.7%
6	0.1.0.0/23	0.0.0.0/4	*	low	6	8.7M	104.2k	10.1%	2.54	39.6%
7	0.1.0.0/26	0.0.0.0/2	high	*	6	50.5M	102.9k	10.0%	2.33	9.4%
8	0.1.0.0/20	0.0.0.0/4	high	80	6	10.5M	106.8k	10.4%	2.28	41.8%
9	0.1.0.0/26	0.0.0.0/3	high	*	*	50.5M	104.5k	10.2%	2.18	7.3%
10	0.1.0.0/22	0.0.0.0/6	high	low	6	9.0M	103.0k	10.0%	2.12	35.7%

- Test on 90-second traces

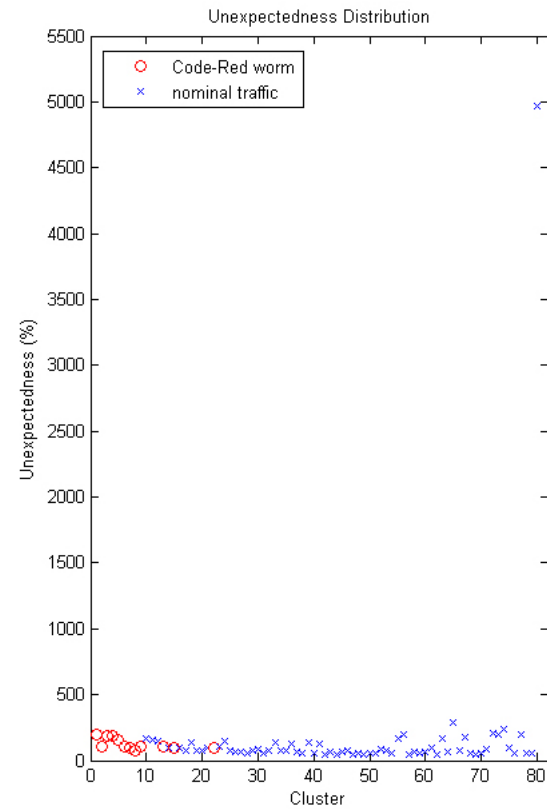
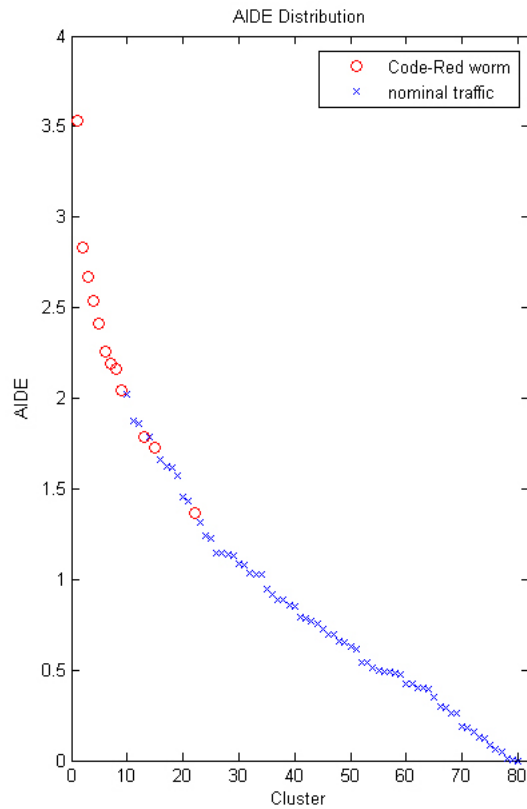
Anomaly Identification – Code-Red v2

AIDE and Unexpectedness Distribution of Code-Red Worm Trace

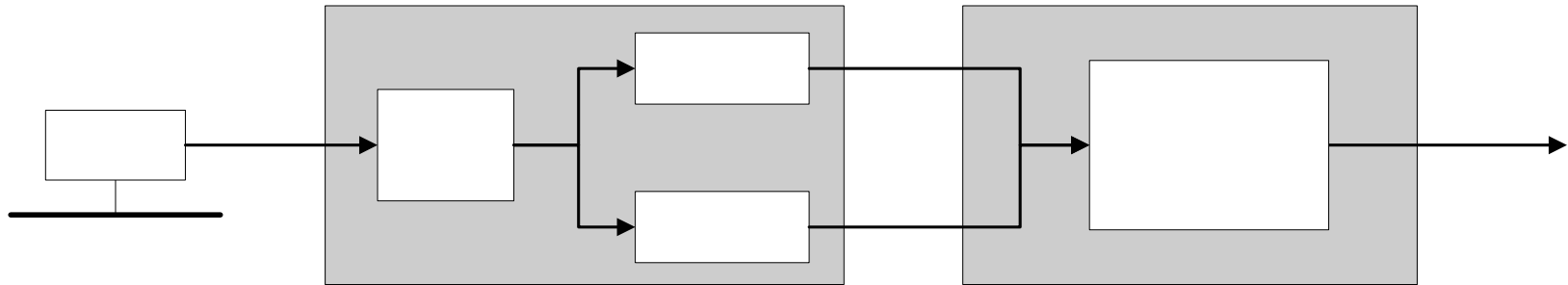


Anomaly Identification – Code-Red v2

AIDE and Unexpectedness Distribution of Merged Code-Red Worm Traces



Content-Based IDS – Structure



Worm Signature Generation:

- Uses Rabin fingerprinting to generate signatures of worms in suspicious pool
- Uses packets in innocuous pool to reduce false positive of worm signatures

Content-Based IDS – Earlybird (1)

➤ System Review

- No packet classifier, works on all incoming packets

- Uses content prevalence to determine worm substrings

Besides payload, destination port number and protocol are also used to partition different worms

- Uses address dispersion to reduce false positive rate

Counts the number of src and dst IP addresses to help determine worm substrings. Address dispersion is very similar to our AIDE idea.

- Rabin fingerprint: fixed length window, overlapping

Generates signatures for all possible substrings of a certain length, but sampling signatures

- Tradeoff between detection accuracy and processing speed

Content-Based IDS – Earlybird (2)

➤ Disadvantages

(1) Can NOT handle polymorphic worms

Earlybird uses long window size (40 bytes) in Rabin fingerprinting, so it can not detect even very simple variation (one-byte insertion and deletion, simply payload reordering) of worms. A worm can easily undermine Earlybird by adding one NOOP every 40 bytes.

(2) Inaccuracy in Detecting Worms

Earlybird handles all incoming packets, so it needs to use sampling and estimation in computing address dispersion and content prevalence, both of which may lead to misdetecting worms.

Moreover, Earlybird samples signatures (taking only 1/64 of all possible substrings), it is possible to miss the worm signatures.

Content-Based IDS – Autograph

➤ System Review

- Packet classifier: identifying packets with suspicious scanning activity (unanswered inbound SYN packets, outbound ICMP unreachable messages ...)
- Rabin fingerprint: variable length window (Content-based Payload Partitioning), non-overlapping

➤ Disadvantages

- Does not work for UDP-based worm (like Slammer) and email borne worms (like MyDoom)
- Non-overlapping Rabin fingerprinting, the partition of packets is too sensitive to the predetermined breakmark.

Content-Based IDS – Polygraph

➤ System Review

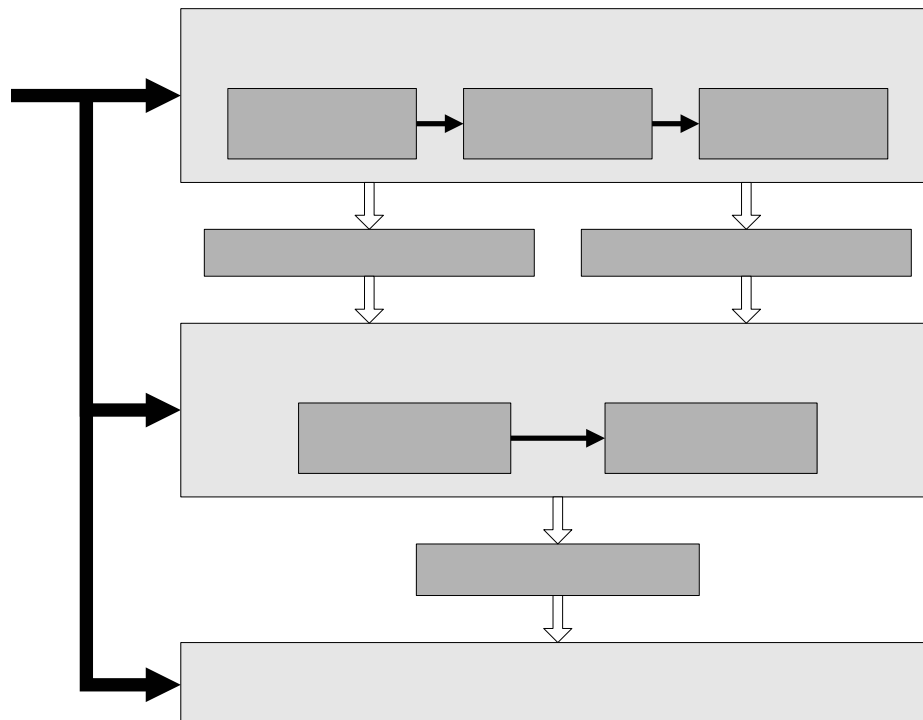
- Focus on detecting and generating signatures for polymorphic worms
- Use Color Set Size (CSS) method to find a set of worm signatures
- Bayes signatures: the probability a packet is a worm, based on the probability of each substring

➤ Disadvantages

- No method developed for classifying packets as innocuous or suspicious; signatures highly relying on the purity of both pools
- High complex computation, impossible for online implementation
- Susceptible to creating “hybrid” worm signatures

Payload-Based IDS (1)

Structure of Payload-Based Intrusion Detection System



- Pipelined implementation, 3 second delay
- Fast speed: suitable to real-time online detection
- Low memory requirement: less than 30 MB

Payload-Based IDS (2)

➤ Multidimensional Clustering

- Obtain any cluster with packet volume larger than threshold (1%)
- Considering both volume and IP cardinality → the tree size is only 1/100 of our previous method

➤ Worm Signature Extraction

- Building suffix tree for each suspicious cluster, and extracting signatures only from a small part of packets
- With time and space linear in the length of each suspicious cluster
- Extracting signatures of any length without complexity increase
- Jointly considering length, frequency and false positive of signatures

Worm Experiment – Basic Information

➤ Trace and Worm Information

Total Normal Packets: 79435 (1 minute)

Total Worm Packets: 898

Worm Packet Information:

Destination Port: 110

Protocol: TCP

WormRate: 15 packets/sec

WormPayloadSize: 100 bytes

WormCommonBytes: 15

WormImages: 20

CommonBytePosition: Random

Worm Experiment – Report

***** NETWORK INTRUSION DTECTION REPORT *****

The network traffic begins at Jun 7, 2004 03:14:39 EST. The traffic length is 1.0 minutes.

The total traffic is 78.5k packets (73.4M bytes). The threshold is 1% = 0.8k packets.

Number	Source IP	Dst IP	SrcPt	DstPt	Prot	PktNo	Perc
1	140.96.114.97/32 (898 Worm Packets + 245 Normal Packets) Total: 21 signatures	0.0.0.0/1	high	110	6	1.1k	1.4%
	1) Leng: 15		Freq: 898		False Pos.: 0		
	2) Leng: 100		Freq: 54		False Pos.: 0		
	3) Leng: 100		Freq: 36		False Pos.: 0		
	4) Leng: 100		Freq: 41		False Pos.: 0		
	5) Leng: 100		Freq: 46		False Pos.: 0		
	6) Leng: 100		Freq: 39		False Pos.: 0		
	7) Leng: 100		Freq: 39		False Pos.: 0		
	8) Leng: 100		Freq: 50		False Pos.: 0		
	9) Leng: 100		Freq: 46		False Pos.: 0		
	10) Leng: 100		Freq: 49		False Pos.: 0		
	11) Leng: 100		Freq: 46		False Pos.: 0		
	12) Leng: 100		Freq: 38		False Pos.: 0		
	13) Leng: 100		Freq: 42		False Pos.: 0		
	14) Leng: 100		Freq: 49		False Pos.: 0		
	15) Leng: 100		Freq: 39		False Pos.: 0		
	16) Leng: 100		Freq: 45		False Pos.: 0		
	17) Leng: 100		Freq: 44		False Pos.: 0		
	18) Leng: 100		Freq: 54		False Pos.: 0		
	19) Leng: 100		Freq: 38		False Pos.: 0		
	20) Leng: 100		Freq: 46		False Pos.: 0		
	21) Leng: 100		Freq: 57		False Pos.: 0		
2	140.96.114.97/32 Total: 2 images	0.0.0.0/1	high	4662	6	1.1k	1.4%
	1) Leng: 101		Freq: 66		False Pos.: 27		
	2) Leng: 68		Freq: 18		False Pos.: 4		

Worm Experiment – Discussion

- Only need to deal with the payloads of 3% packets.
- **1st Cluster:**
 - All worm-related signatures, both 15-byte common string and 20 worm images, are correctly extracted
 - No false signatures even with 21% normal packets
 - Zero false alarm counts for each worm-related signature
- **2nd Cluster:**
 - Peer-to-Peer file transfer
 - Two signatures extracted, but can be easily rejected by monitoring their future occurrence or their represented contents
 - Positive false alarm counts for both signatures

References

1. C. Estan, S. Savage, and G. Varghese, Automatically inferring patterns of resource consumption in network traffic, *SIGCOMM*, Aug. 2003.
2. J. Wang, D. J. Miller, and G. Kesidis, Efficient Mining of the Multidimensional Traffic Cluster Hierarchy for Digesting, Visualization, and Anomaly Identification, To appear in *IEEE JSAC*, 2006.
3. S. Singh, C. Estan, G. Varghese, and S. Savage, Automated worm fingerprinting, In *Proceedings of the 6th ACM/USNIX Symposium on Operating System Design and Implementation*, Dec. 2004.
4. H. A. Kim and B. Karp, Autograph: Toward Automated, Distributed Worm Signature Detection, In *Proceedings of the 13th USENIX Security Symposium*, Aug. 2004.
5. J. Newsome, B. Karp, and D. Song, Polygraph: Automatically Generating Signatures for Polymorphic Worms, In *Proc. of the IEEE Symp. on Security and Privacy*, May 2005.

Thanks!