LA-UR-14-20584
Approved for public release; distribution is unlimited.

Title: Integrating Multiple Data Views for Improved Malware Analysis

Author(s): Anderson, Blake H.

Intended for: Dissertation Defense

Issued: 2014-01-31

Integrating Multiple Data Views for Improved Malware Analysis

Dissertation

Blake Anderson

Introduction

- Malicious programs continue to be a serious threat within the internet landscape
- Advanced Persistent Threats (APT) are becoming more frequent
- APT typically uses 0-day malware (e.g. Stuxnet)
- Given recent trends and how lucrative the cybercrime industry has become, malware is expected to be an ongoing threat

How Can We Stop Malware?

- **Classify 0-day malware**
  - Traditional antivirus software will not detect 0-day malware[3]
  - Develop tools that can accurately classify malware with an acceptable level of false positives
- **Support malware analytics/forensics**
  - Does a new sample of malware belong to a known family?
  - Is it possible to attribute a new sample of malware to a known creator?

Why is this problem hard?

- **Malware has many protection mechanisms in place to prevent analysts from understanding its intent:**
  - Static domain
    - Packers help to obfuscate the code[4]
    - Large portions of the code can be encrypted[4]
  - Dynamic domain
    - Execution-stalling techniques[5]

Thesis

Exploiting multiple views of a program makes obfuscating the intended behavior of a program more difficult allowing for better performance in classification, clustering, and phylogenetic reconstruction.

Contributions

- Use a Markov chain data representation for several well-known data views of malware (security)
- Multiple kernel learning framework to create a highly accurate classifier for malware (security)
- Combine multiple data views for the clustering domain (ML) and apply this to the malware problem (security)
- Multiview method to create a phylogenetic reconstruction (ML) for malware samples (security)

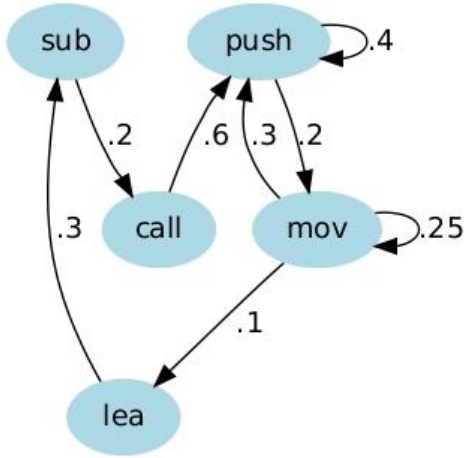
Outline

- **Using the Markov chain data representation**
- Incorporating multiple views of the data for classification
- Incorporating multiple views of the data for clustering
- Phylogenetic Reconstruction

Markov Chain Data Representation

- **Given a sequence-based view of malware (i.e., the dynamic trace), transform this view into a Markov chain**

call	[ebp+0x8]
push	0x70
push	0x010012F8
call	0x01006170
push	0x010061C0
mov	eax, fs:[0x00000000]
push	eax
mov	Fs:[], esp
mov	eax, [esp+0x10]
mov	[esp+0x10], ebp
lea	ebp, [esp+0x10]
sub	esp, eax
…	…



Defining Kernels

- **Use graph kernels to compute the similarity matrix between Markov chains**[9]
- **Gaussian kernel between the edge weights:**

$$K_G(x, x') = e^{-\frac{1}{2\sigma^2}\sum_{ij}\left(x_{ij}-x'_{ij}\right)^2}$$

  - Measures local similarities between the graphs
- **Spectral kernel between the eigenvectors of the graphs:**

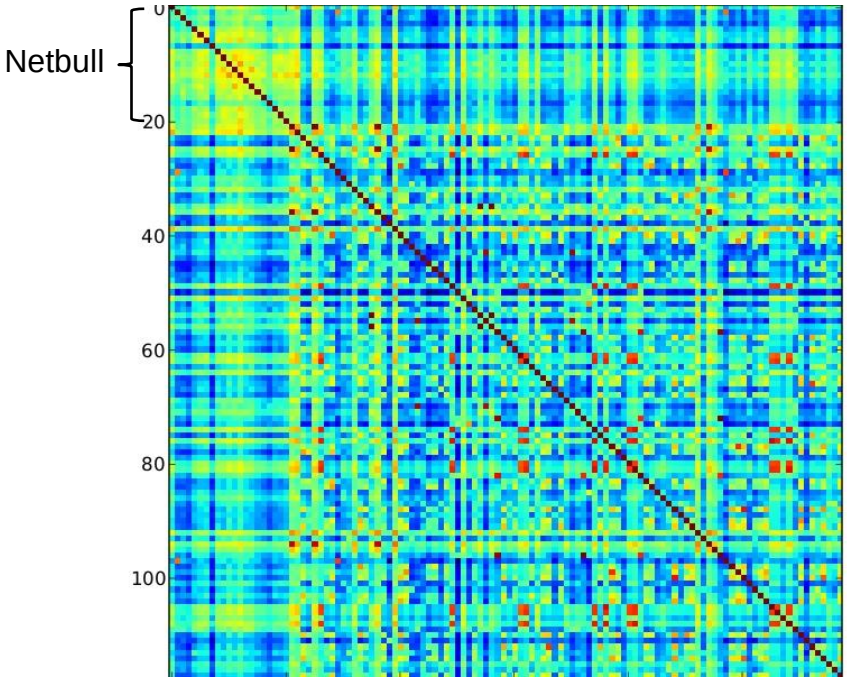
$$K_S(x, x') = e^{-\frac{1}{2\sigma^2}\sum_{k}\left\|\phi_k(x)-\phi_k(x')\right\|^2}$$

  - Measures global similarities between the graphs such as diameter, number of connected components and the stationary distributions

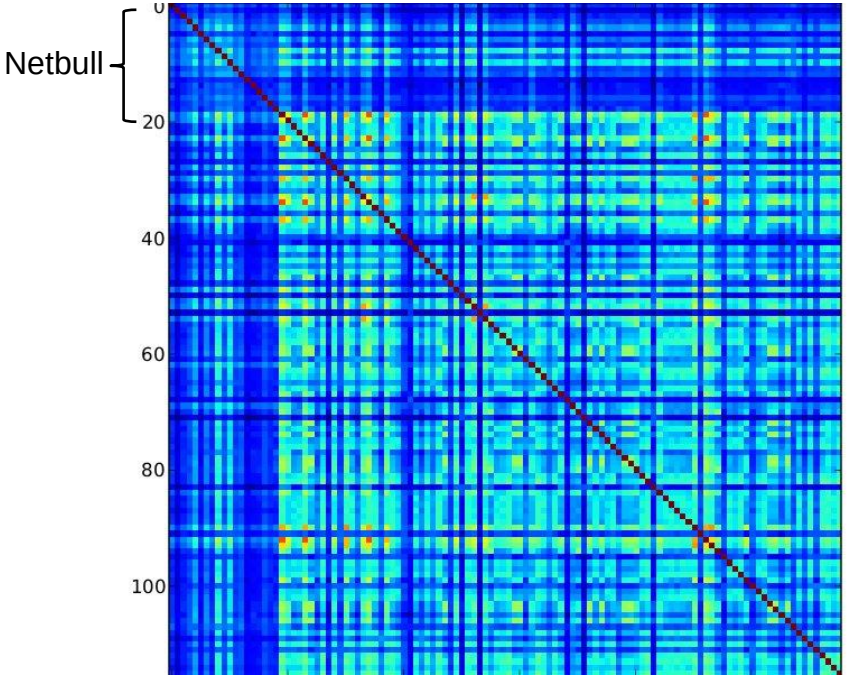
Example Kernels

- **Kernels between 97 randomly selected malware samples and 21 instances of the netbull virus**

Gaussian Kernel



Spectral Kernel



Markov Chain Representation Results

- **2,230 programs: 1,615 malicious programs, 615 benign programs**

Method	Accuracy	FPs
Combined Kernel	96.41%	47
Gaussian Kernel	95.70%	44
Spectral Kernel	90.99%	80
N-gram (3, 2500)	82.15%	300
N-gram (4, 2000)	81.17%	327
N-gram (2, 1000)	80.63%	325
AV0	73.32%	0
AV1	53.86%	1
AV2	49.60%	0

Why Can't We Stop Here?

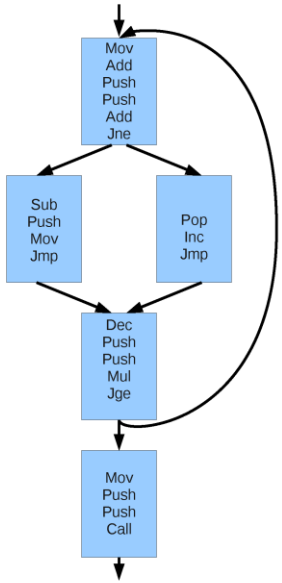
- Dynamic instruction traces are very slow to collect
- Dynamic instruction traces require a lot of resources to collect
- Dynamic instruction traces are not always reliable, as malware has evolved, it has developed execution-stalling techniques[5]
- Hypothesis of this work: incorporating multiple views of malware yields better classification/clustering performance

Outline

- Using the Markov chain data representation
- **Incorporating multiple views of the data for classification**
- Incorporating multiple views of the data for clustering
- Phylogenetic Reconstruction

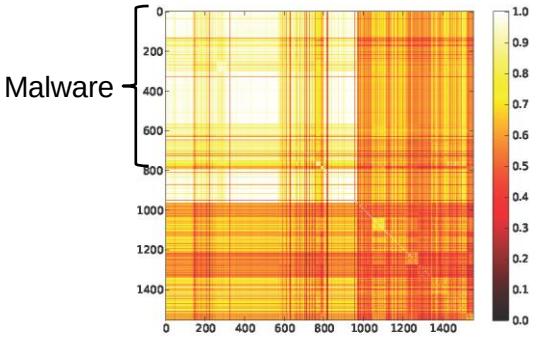
Complementary Data Views

- **Dynamic views I use:**
  - Dynamic instruction calls[7] (MC, Gaussian kernel)
  - System calls[16] (MC, Gaussian kernel)
- **Static data views I use:**
  - Byte information of the executable[18] (MC, Gaussian kernel)
  - Disassembled instructions[17] (MC, Gaussian kernel)
  - Control flow graph[3] (Graphlet kernel)
- **Several previously examined statistics:**
  - Entropy, known packer, size of CFG/binary

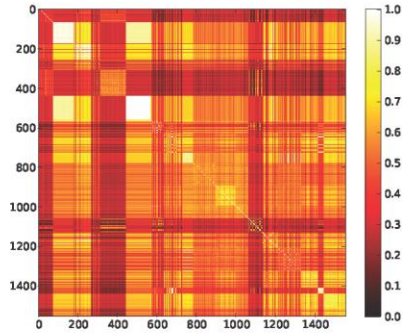


Kernels For Each View

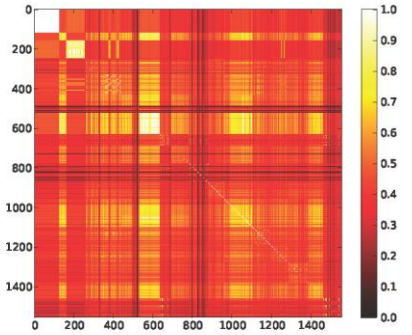
Byte Information



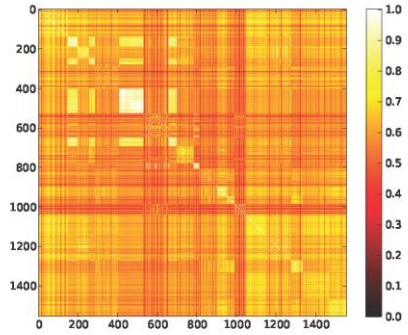
Disassembled



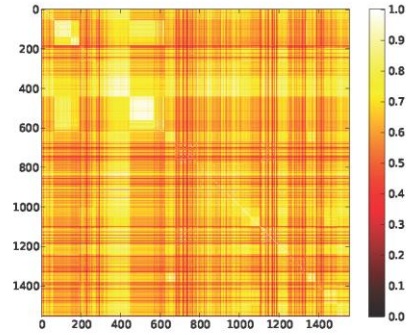
CFG



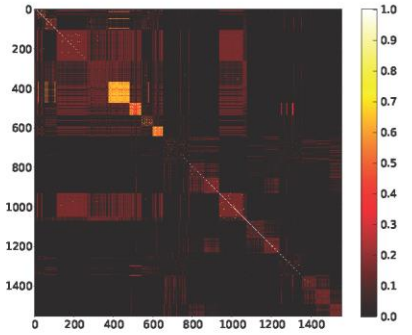
Dynamic Instructions



System Calls



File Information



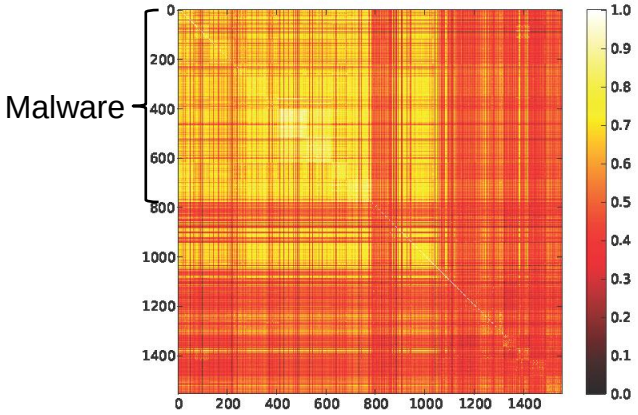
Combining Kernels

- **Goal: find a convex combination of kernels:**

$$K_c = \sum_{i=0}^{M} \beta_i K_i$$

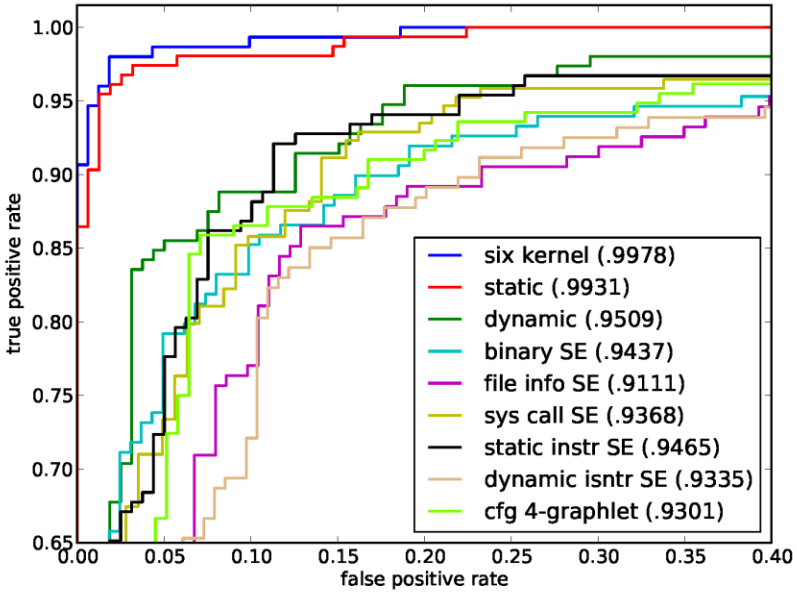
with $\beta_i \geq 0$; $\sum_i \beta_i = 1$ such that we maximize classification accuracy

- **There are standard MKL algorithms to find both the optimal β's and optimal SVM parameters**[10]



MKL Classification Results

- **1556 samples: 780 malicious programs, 776 benign programs**



Outline

- Using the Markov chain data representation
- Incorporating multiple views of the data for classification
- **Incorporating multiple views of the data for clustering**
- Phylogenetic Reconstruction

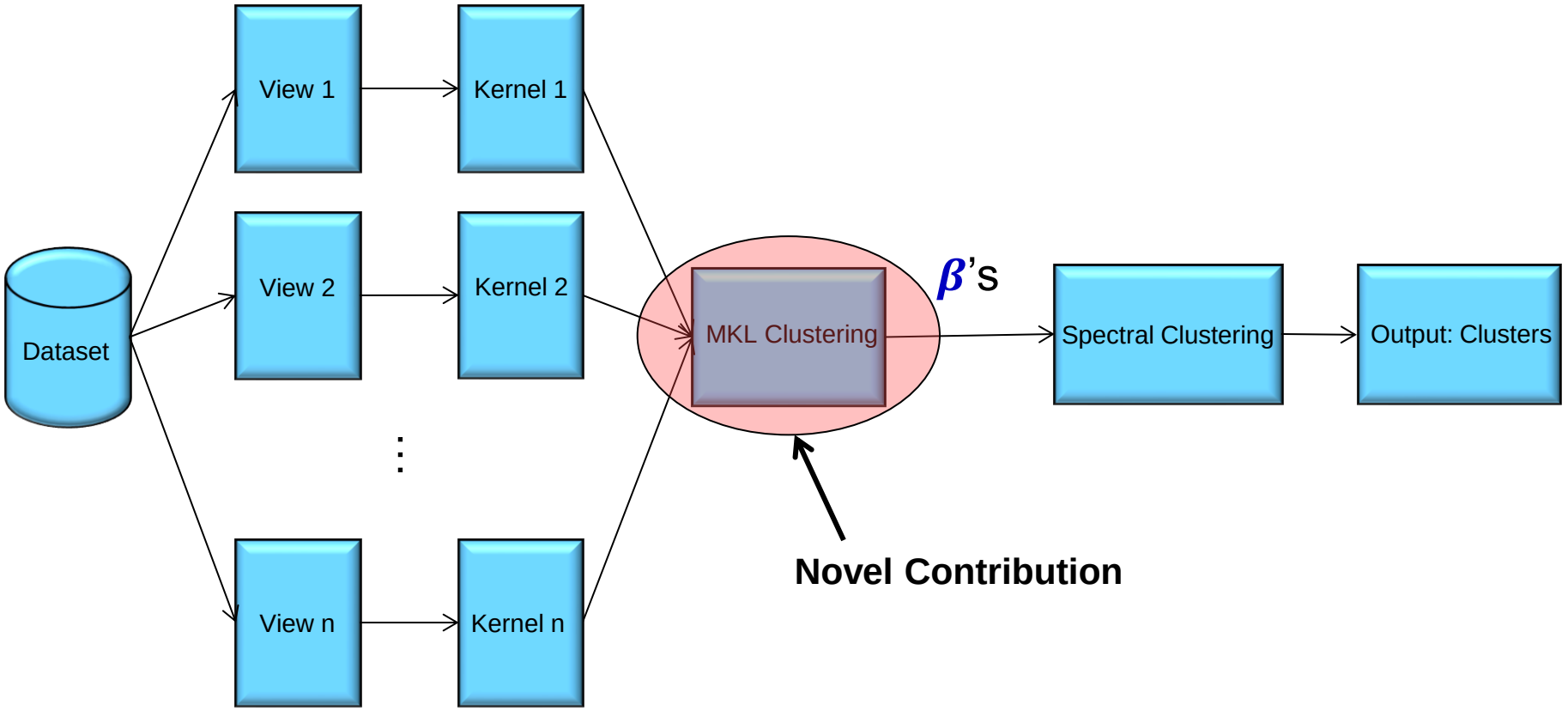
OK, Can We Stop Now?

- **Unfortunately, classifying programs as malicious or benign is only half of the problem**
- **Once a program is known to be malware, the damage the malware has caused needs to be mitigated**
  - Does it belong to a known family of malware?
  - Does it have common functionality with known pieces of malware?
- **Can the malware be attributed to a known organization or creator?**

MKL Clustering

- **Same idea: incorporating multiple views of malware yields better clustering performance**
- **Traditional multi-view clustering techniques have required a priori information as to which views are more informative**[11,12]
  - In many domains, the information is not available
  - In the malware domain, the most informative view will likely change between different datasets/families
- **I have developed a novel extension to the MKL clustering literature which requires no a priori information about the importance of views**

Architecture Diagram



MKL Clustering Algorithm

- **Basic idea: modify the spectral clustering objective function[13] to take multiple views of the data into account:**

$$\min_{U \in R^{n \times k}} tr\left(U^T L(\beta) U\right) \; s.t. \; U^T U = I$$

where we define the unnormalized multi-view Laplacian as:

$$L(\beta) = \sum_{i=1}^{M} \beta_i D_i - \sum_{i=1}^{M} \beta_i K_i$$

- U **now defines a new feature space, taking multiple views into account, in which the instances can be trivially clustered**
- **We still need to find** β

MKL Clustering Algorithm

- The optimal $\boldsymbol{\beta}$ vector can be found with respect to both the spectral clustering objective function and $\boldsymbol{U}$ with the following SDP:

$$\min \|\boldsymbol{A}(\boldsymbol{\beta})\|_* + \frac{1}{2}\boldsymbol{\beta}^T \boldsymbol{C}\boldsymbol{\beta} \quad \boldsymbol{s.t.} \quad \boldsymbol{G}\boldsymbol{\beta} \preccurlyeq \boldsymbol{h}$$

where

$$\boldsymbol{A}(\boldsymbol{\beta}) = \sum_{i=0}^{M} \boldsymbol{\beta}_i \boldsymbol{A}_i$$

and

$$\boldsymbol{A}_i = \boldsymbol{U}^T(\boldsymbol{D}_i - \boldsymbol{K}_i)\boldsymbol{U}$$

- The kernel parameters, $\boldsymbol{\beta}$, and the new features, $\boldsymbol{U}$ can be solved for iteratively

Results

- **606 malware instances from 12 malicious families**

Method	ARI
SDP Normalized	**.8768**
SDP Unnormalized	**.8747**
Centroid	.8702
MKL Classification	.8531
Pair-wise	.8477
Uniform Combination	.8381
Best View	.8174

Outline

- Using the Markov chain data representation
- Incorporating multiple views of the data for classification
- Incorporating multiple views of the data for clustering
- **Phylogenetic Reconstruction**

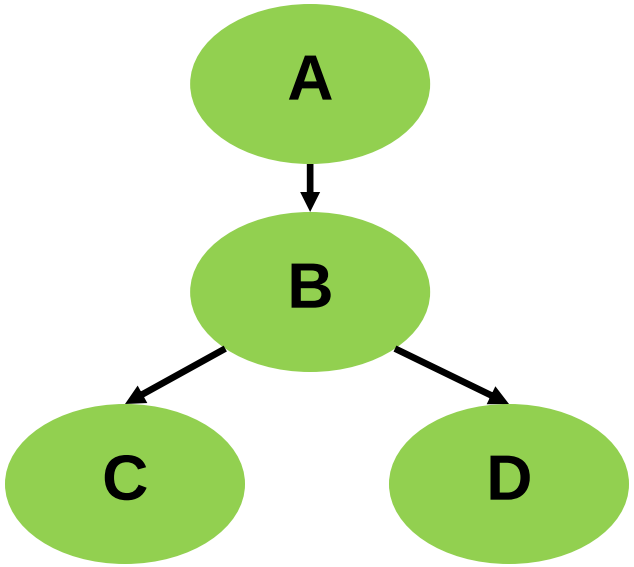
Phylogenetic Reconstruction

- **Malware evolves much like biological organisms**
  - Can have offspring (sexual and asexual)
  - Can exhibit convergent/divergent evolution

- **Malware also has some distinct differences**
  - Tree of Life assumption may not fit
  - Potentially much more dramatic evolution

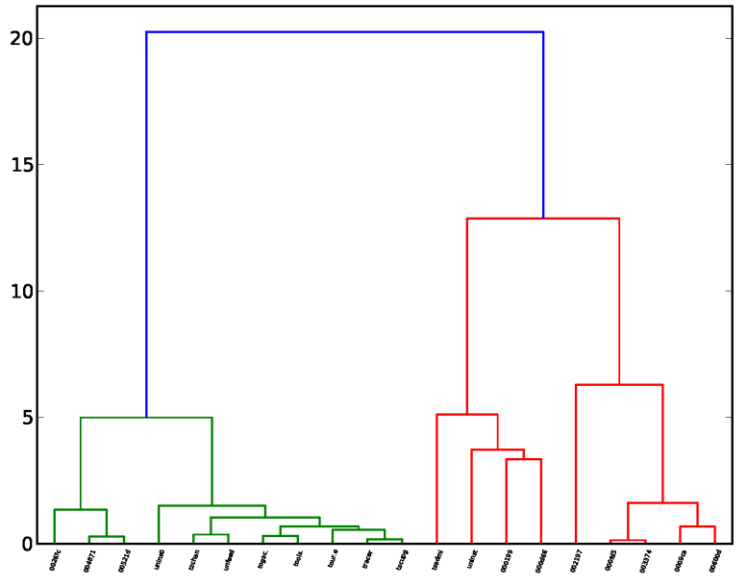
Goal of Phylogenetic Reconstruction

- Given a set of programs, the problem is to find a graph
- The nodes in the graph are the sample programs
- The edges in the graph represent how the program evolves

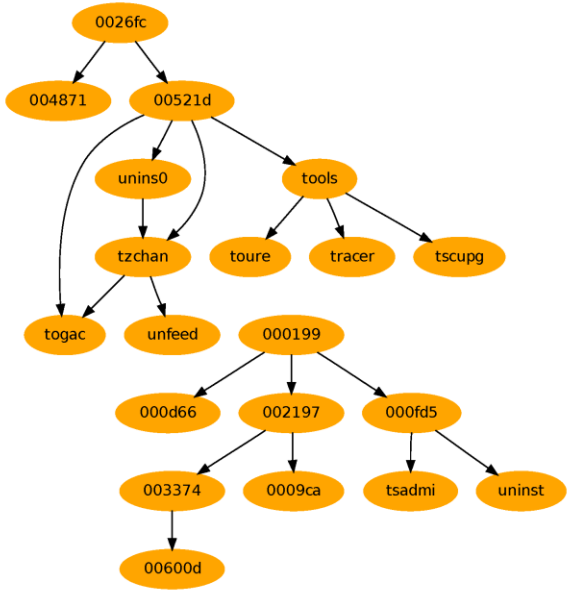


Phylogenetic Reconstruction

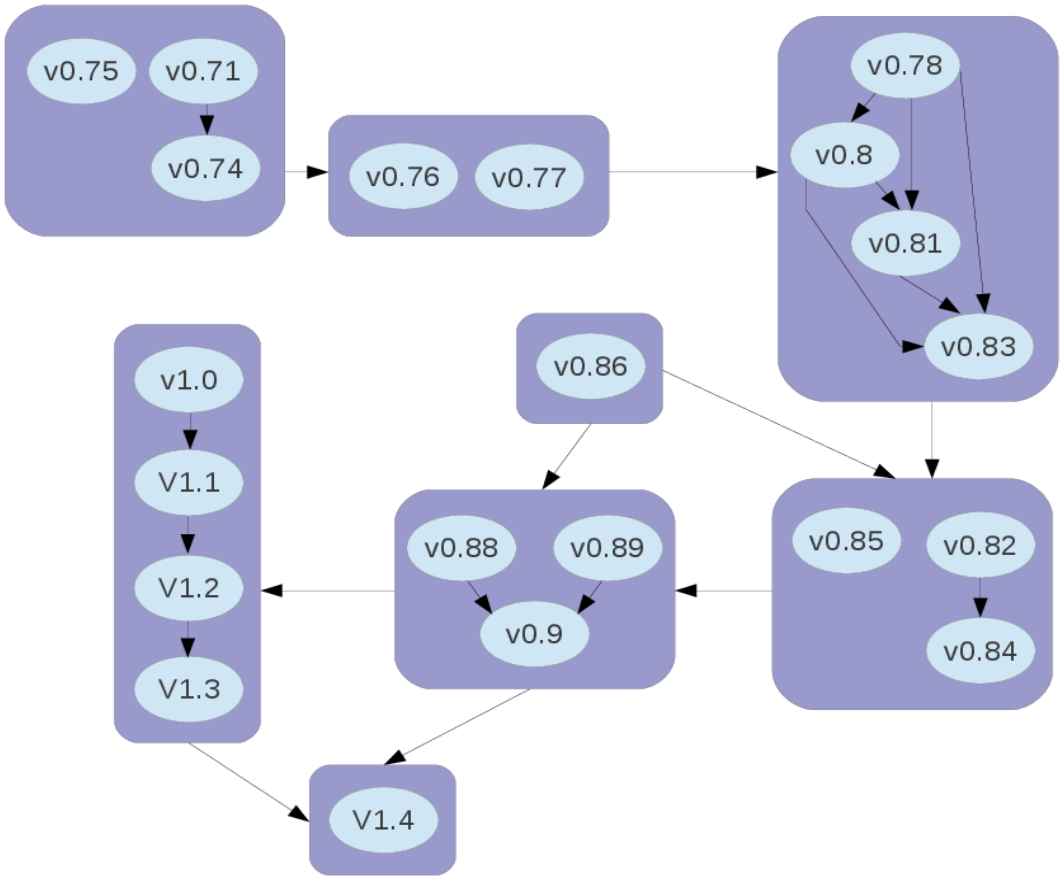
Hierarchical Clustering



Phylogenetic Reconstruction



Phylogenetic Reconstruction Example



Graphical Lasso

- Given a covariance matrix, glasso finds a sparse precision matrix:

$$\max_{\Theta} \; \boldsymbol{log\ det\ \Theta} \; - \boldsymbol{tr(K\Theta)} - \; \boldsymbol{\|\Theta \circ P\|_1}$$

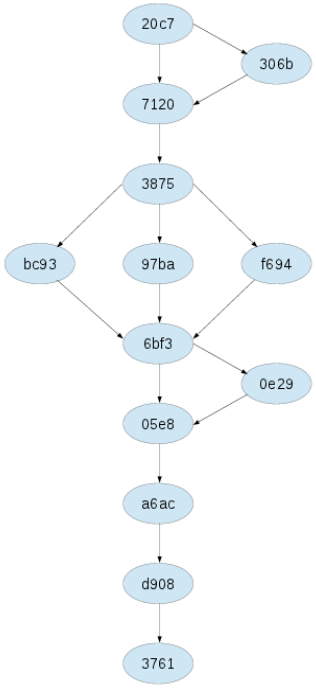
- We need to adjust this to take multiple views into account:

$$\max_{\Theta,\boldsymbol{\beta}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{M}} \boldsymbol{log\ det\ \Theta} \; - \boldsymbol{\beta_i tr(K_i\Theta)} - \; \boldsymbol{\|\Theta \circ P\|_1} \; - \boldsymbol{\lambda\|\beta\|_2}$$

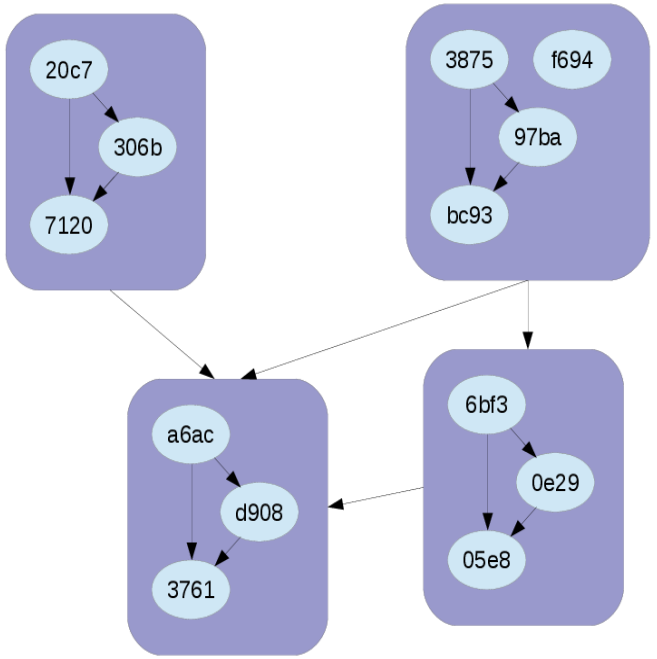
- I have developed a novel algorithm using alternating projections to solve this problem

Results: Mineserver

Ground Truth



Phylogenetic Reconstruction



Experiment Setup

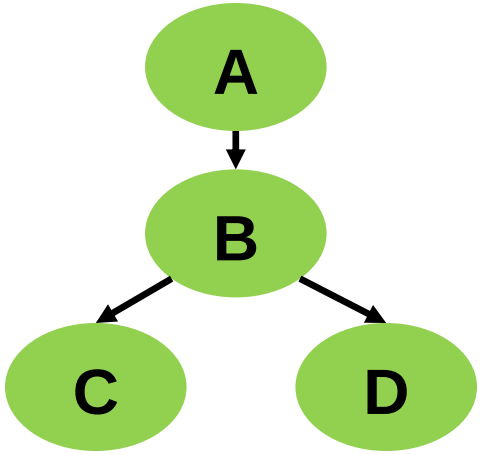
- **5 different families**
  - Mineserver (from github repository), 13 instances
  - NetworkMiner (from svn repository), 21 instances
  - Bagle, 25 instances
  - Koobface, 19 instances
  - Mytob, 20 instances
- **Views of each program:**
  - Byte information
  - Disassembled instructions
  - Control flow graph
  - Dynamic instructions
  - Summary feature vector

Competing Methods

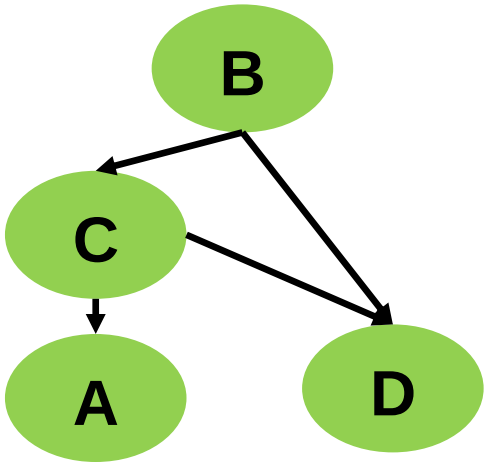
- **Graphical Lasso**
  - Single best view
- **Minimum Spanning Tree**
  - Naïve baseline
- **Gupta Algorithm[19]**
  - Developed specifically for malware
  - Based on pruning a completely connected graph
  - If the weight of any pair of edges < δ_2, prune weaker edge
  - If the weight of all incoming edges < δ_1, prune all incoming edges

Precision/Recall

Ground Truth



Reconstructed Graph



- **Precision: 2/4**
- **Recall: 2/3**

Results

Recall					
	NetworkMiner	MineServer	Bagle	Mytob	Koobface
MKLGlasso	.85	.8125	.3333	.5263	.5
Glasso	.55	.1935	.1176	.1935	.3171
Gupta	.40	.3438	.125	.0526	.3333
Min Spanning	.70	0.0	.0417	.1053	.0556

Precision					
	NetworkMiner	MineServer	Bagle	Mytob	Koobface
MKLGlasso	.4857	.7222	.20	.1563	.5812
Glasso	.2895	.4118	.0704	.0864	.2391
Gupta	.3810	.8462	.12	.05	.3158
Min Spanning	.35	0.0	.0208	.0526	.0278

Conclusion

- **To take steps toward stopping malware, we need to:**
  - Accurately classify new 0-day malware
  - Cluster malware to help reverse engineers more quickly understand its function
  - Learn to attribute malware to known creators/organizations
- **I have presented several novel methods which use the multiple views of programs to achieve these three goals**
- **We are currently implementing pieces of the MKL classification framework within LANL's CodeVision antivirus technology**

Publications

Markov chain data representation

- Blake Anderson, Daniel Quist, Curtis Storlie, Joshua Neil, and Terran Lane. Graph-Based Malware Detection using Dynamic Analysis. *Journal of Computer Virology,* pages 1-12, 2011.
- Curtis Storlie, Blake Anderson, Scott Vander Wiel, Daniel Quist, Curtis Hash, and Nathan Brown. Stochastic Identification and Clustering of Malware with Dynamic Traces. Annals of Applied Statistics. Accepted.

Multiple view classification

- Blake Anderson, Curtis Storlie, and Terran Lane. Improving Malware Classification: Bridging the Static/Dynamic Gap. *Proceedings of the 5th ACM workshop on Security and Artificial Intelligence*, pages 3-14, 2012.

Multiple view clustering

- Blake Anderson, Curtis Storlie, and Terran Lane. Multiple Kernel Learning Clustering with an Application to Malware. *Proceedings of the 12th IEEE International Conference on Data Mining*, pages 804-809, 2012.

Applications

- Blake Anderson, Daniel Quist, and Terran Lane. Detecting Code Injection Attacks in Internet Explorer. *Proceedings of the IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW)*, pages 90-95, 2011.

Patents

- Blake Anderson, Curtis Storlie, and Terran Lane. Integrating Multiple Data Sources for Malware Classification. S13/909,985, 2013.

Thank You!

References

[1] Symantec 2011 Annual Report, http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_2011_21239364.en-us.pdf

[2] Consumer Reports, July 2011, http://www.consumerreports.org/cro/magazine-archive/2011/june/june-2011-toc.htm

[3] Mihai Christodorescu and Somesh Jha. Static Analysis of Executables to Detect Malicious Patterns. In *Proceedings of the 12th USENIX Security Symposium*, pages 169-186, 2003.

[4] Andreas Moser, Christopher Kruegel, and Engin Kirda. Limits of Static Analysis for Malware Detection. *Computer Security Applications Conference, Annual,* pages 421-430, 2007.

[5] Clemens Kolbitsch, Engin Kirda, and Christopher Kruegel. The Power of Procrastination: Detection and Mitigation of Execution-Stalling Malicious Code. In *Proceedings of the 18th ACM conference on Computer and Communications Security, pages 285-295, 2011.*

[6] Ulrich Bayer, Andreas Moser, Christopher Kruegel, and Engin Kirda. Dynamic Analysis of Malicious Code. *Journal of Computer Virology,* pages 67-77, 2006.

[7] Blake Anderson, Daniel Quist, Curtis Storlie, Joshua Neil, and Terran Lane. Graph-Based Malware Detection using Dynamic Analysis. *Journal of Computer Virology,* pages 1-12, 2011.

[8] Jianyong Dai, Ratan Guha, and Joohan Lee. Efficient Virus Detection Using Dynamic Instruction Sequences. *Journal of Computers,* Volume 4, Issue 5, 2009.

[9] H. Kashima, K. Tsuda, and A. Inokuchi. *Kernel for Graphs.* MIT Press, 2004.

[10] Soren Sonnenburg, Gunnar Raetsch, and Christin Schaefer. A General and Efficient Multiple Kernel Learning Algorithm. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems,* 2005.

[11] Abhishek Kumar, Piyush Rai, and Hal Daume III. Co-regularized Multi-view Spectral Clustering. In *Advances in Neural Information Processing Systems,* Volume 24, pages 1413-1421, 2011.

References

[12] Dengyong Zhou and Christopher J. C. Burges. Spectral Clustering and Transductive Learning with Multiple Views. In *Proceedings of the 24th International Conference on Machine Learning.* Pages 1159-1166, 2007.

[13] Ulrike Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing.* Pages 395-416, 2007.

[14] Md. Karim, Andrew Walenstein, Arun Lakhotia, and Laxmi Parida. Malware Phylogeny Generation Using Permutation of Code. *Journal of Computer Virology.* Pages 13-23, 2005.

[15] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse Inverse Covariance Estimation with the Graphical Lasso. *Biostatistics.* Pages 432-441, 2008.

[16] Steven A. Hofmeyr, Stephanie Forrest, and Anil Somayaji. Instruction Detection Using Sequences of System Calls. *Journal of Computer Security*. Pages 151-180, 1998.

[17] Daniel Bilar. Opcodes as Predictors for Malware. *International Journal of Electronic Security and Digital Forensics.* Pages 156-168, 2007.

[18] Jeremy Kolter and Marcus Maloof. Learning to Detect and Classify Malicious Executables in the Wild. *The Journal of Machine Learning Research.* Pages 2721-2744, 2006.

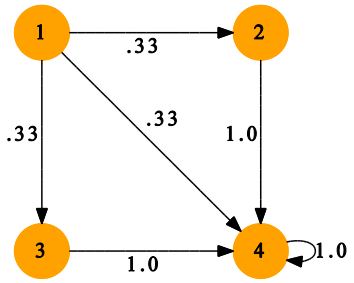
[19] Archit Gupta, Pavan Kuppili, Aditya Akella, and Paul Barford. An Empirical Study of Malware Evolution. Communication Systems and Networks and Workshops, 2009.

Using the Dynamic Instruction Trace (Backup Slide)

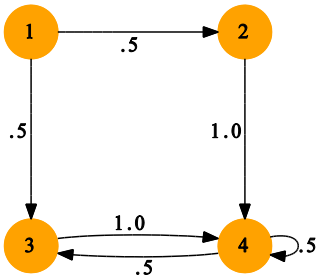
- **Collect the instructions the program executes in a safe (virtual) environment**
- **Compared to the disassembled (static) instructions, it is a more reliable source for the intended behavior of a program**[4]
- **It is not always possible to get the disassembled instructions from a program**[4]
- **Dynamic instructions have been shown to yield excellent classification accuracies**[6,7]
- **Typically, the feature vectors for dynamic instructions have used *n*-grams**[6,8]

Kernel Example (Edge Weights) (Backup Slide)

Markov Chain 1



Markov Chain 2



Adjacency Matrix 1

0.0	.33	.33	.33
0.0	0.0	0.0	1.0
0.0	0.0	0.0	1.0
0.0	0.0	0.0	1.0

Adjacency Matrix 2

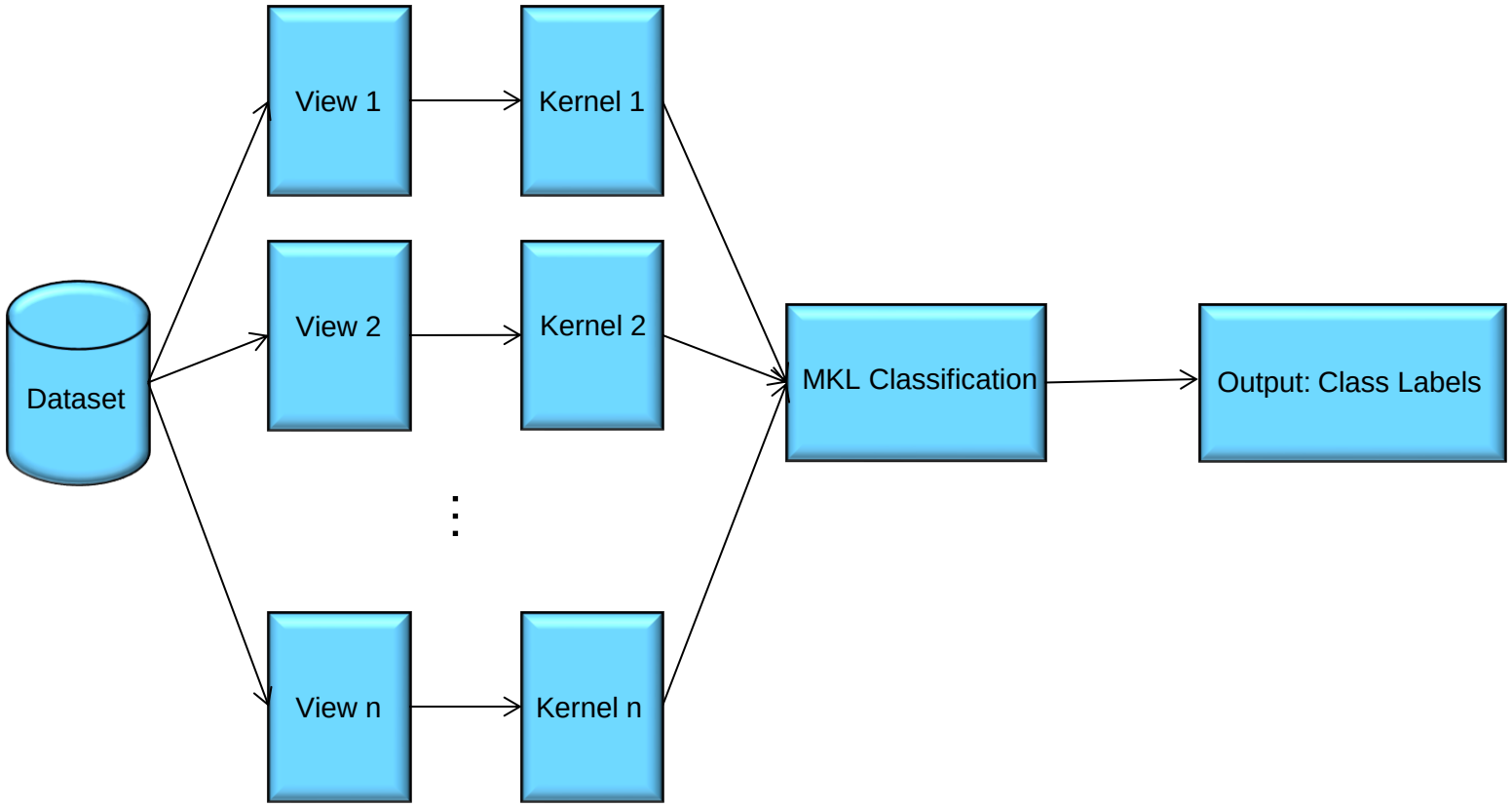
0.0	0.5	0.5	0.0
0.0	0.0	0.0	1.0
0.0	0.0	0.0	1.0
0.0	0.0	0.5	0.5

$$K(x, x') = e^{-\frac{1}{2\sigma^2}\left((.33-.5)+(.33-.5)+(.33-.0)+(1.0-1.0)+(1.0-1.0)+(.0-.5)+(1.0-.5)\right)^2}$$

Experimental Setup (Backup Slide)

- **Hypothesis: Markov chains is a more informative representation compared to *n*-grams**
- **I had 2,230 programs: 1,615 malicious programs, 615 benign programs**
- **I collected dynamic instruction traces from each program**
  - Xen hypervisor/Ether collected traces
  - Ether attempts to hide itself from malware
  - I ran each program for 5 minutes
- **I compared against traditional *n*-gram representation**
  - *n* varied from 2 to 6
  - *L* varied from 500 to 3,000 in increments of 500
- **I used support vector machines for the classification**

Architecture Diagram (Backup Slide)

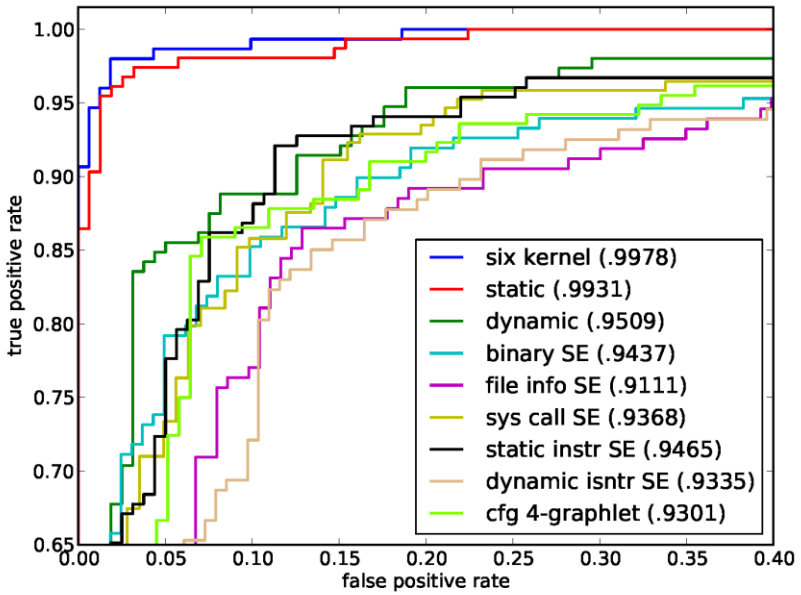


Experimental Setup (Backup Slide)

- **Hypothesis: Combining multiple views of the programs will increase the accuracy of our classifier**
- **I had 1556 samples: 780 malicious, 776 benign**
- **I collected the dynamic data under KVM with the Intel Pin program**
  - 5 minute traces were extracted
  - Pin is able to simultaneously collect instructions and system calls
  - But, unlike Ether, Pin does not attempt to hide itself
- **I used IDA Pro to collect the disassembled data and CFGs**

MKL Classification Results (Backup Slide)

Method	Accuracy	FPs
All Six Views	98.07%	16
Static Views	95.95%	37
Dynamic Views	88.75%	88
Binary	88.11%	93
Disassembled	89.46%	75
CFG	87.72%	87
Dynamic Instructions	87.34%	92
System Calls	87.08%	88
File Information	84.83%	126
AV0	78.46%	4
AV1	75.26%	7
AV2	71.79%	0



MKL Timing Results (Backup Slide)