KASPERSKY lab

# Iterative System Call Patterns Blow the Malware Cover


Mansour Ahmadi, Ashkan Sami, Hossein Rahimi, Babak Yadegari,
MansourAhmadi@gmail.com, asami@ieee.org
Shiraz University, Iran


"IT Security for the Next Generation"
Asia Pacific & MEA Cup 2011
University of Technology MARA, Malaysia
March 4- 6, 2011

# Outline

- Introduction
- Related Works
  - Static & Dynamic Analysis
- Motivation
- Our Method
- Experiments
- Conclusion
- References

# Introduction

- Anti-Viruses use signature-based method for detecting malware
- Polymorphic and metamorphic malware mutate their structures so anti-viruses regularly need to be updated
- We focused on MS windows portable executable (PE) malware
- Because of obfuscation, we should analyze PEs interactions with OS
- Our approach is based on data mining approach to facilitate and expedite the malware diagnosis

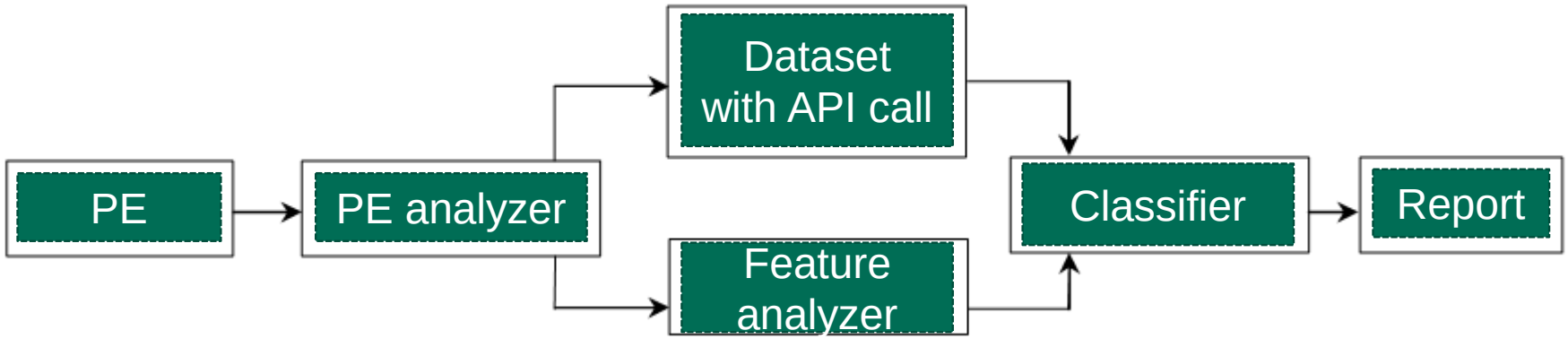
# Related Works

- Malware Detection Based On Mining API Calls

  Ashkan Sami, Hossein Rahimi , Babak Yadegari, Naser Peiravian, Sattar Hashemi, Ali Hamze, (Shiraz University)

  the *ACM Symposium on Applied Computing,* 2010.

- Static, Mining API Calls from Header, Accuracy = 98%
- Obfuscation, Injecting Fake API Calls



# Related Works (…)

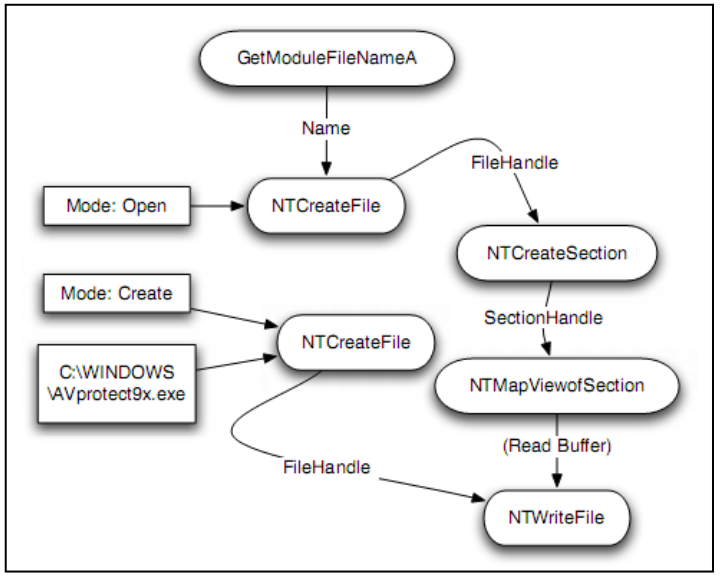
- Effective and Efficient Malware Detection at the End Host

  Clemens Kolbitsch, Paolo Milani Comparetti, Christopher Kruegel, Engin Kirda, Xiaoyong Zhou (Secure System Labs, California)

  Usenix. August 2009.

- Dynamic, Create Graph, Accuracy = 63%, high complexity

```
1  GetModuleFileNameA([out] lpFilename -> "C:\
       netsky.exe")
2  ...
3  NtCreateFile(Attr->ObjectName:"C:\netsky.exe",
       mode: open, [out] FileHandle -> A)
4  ...
5  NtCreateFile(Attr->ObjectName:"C:\WINDOWS\
       AVprotect9x.exe", mode: create, [out]
       FileHandle -> B)
6  ...
7  NtCreateSection(FileHandle: A, [out]
       SectionHandle -> C)
8  NtMapViewOfSection(SectionHandle: C,
       BaseAddress: 0x3b0000)
9  ...
10 NtWriteFile(FileHandle: B, Buffer: "MZ\90\00...
       ", Length: 16896)
11 ...
```



# Related Works (…)

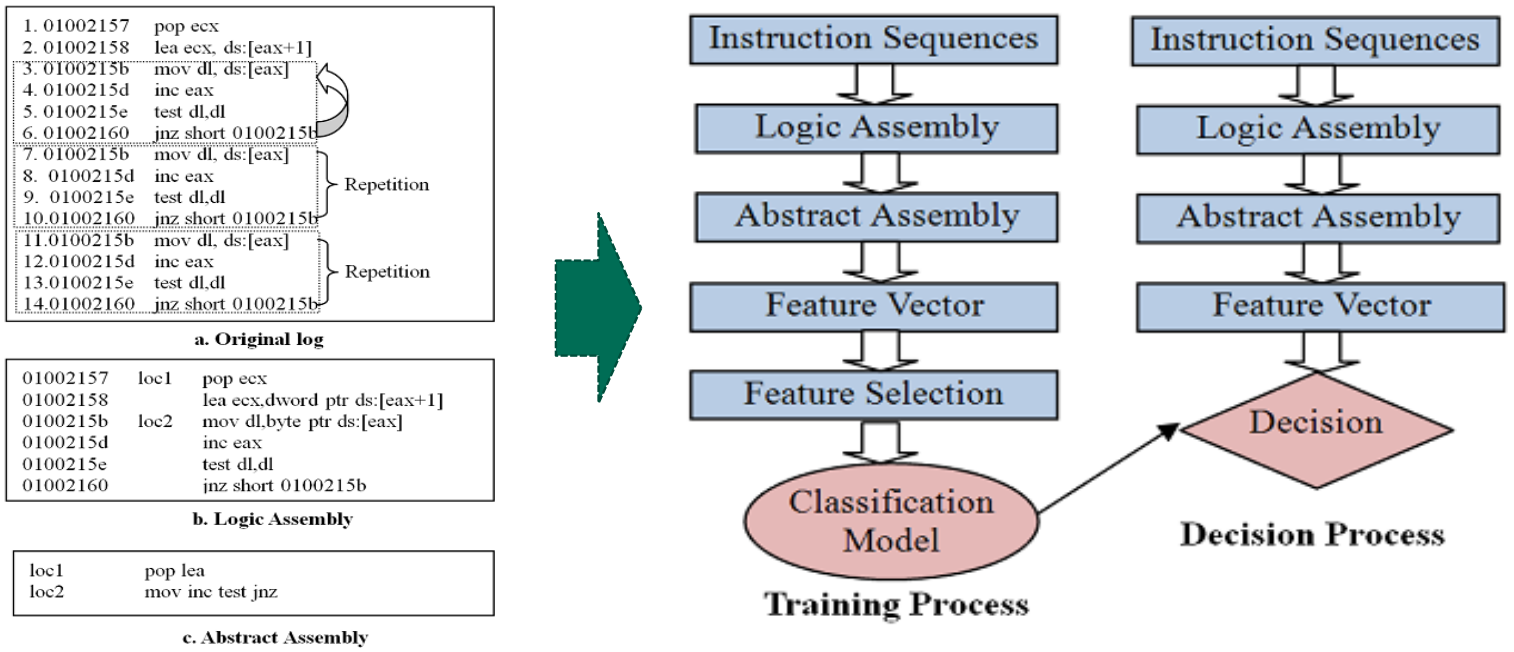


- Efficient Virus Detection Using Dynamic Instruction Sequences

  Jianyong Dai, Ratan Guha, Joohan Lee (University of Central Florida)

  Journal Of Computers. May 2009.

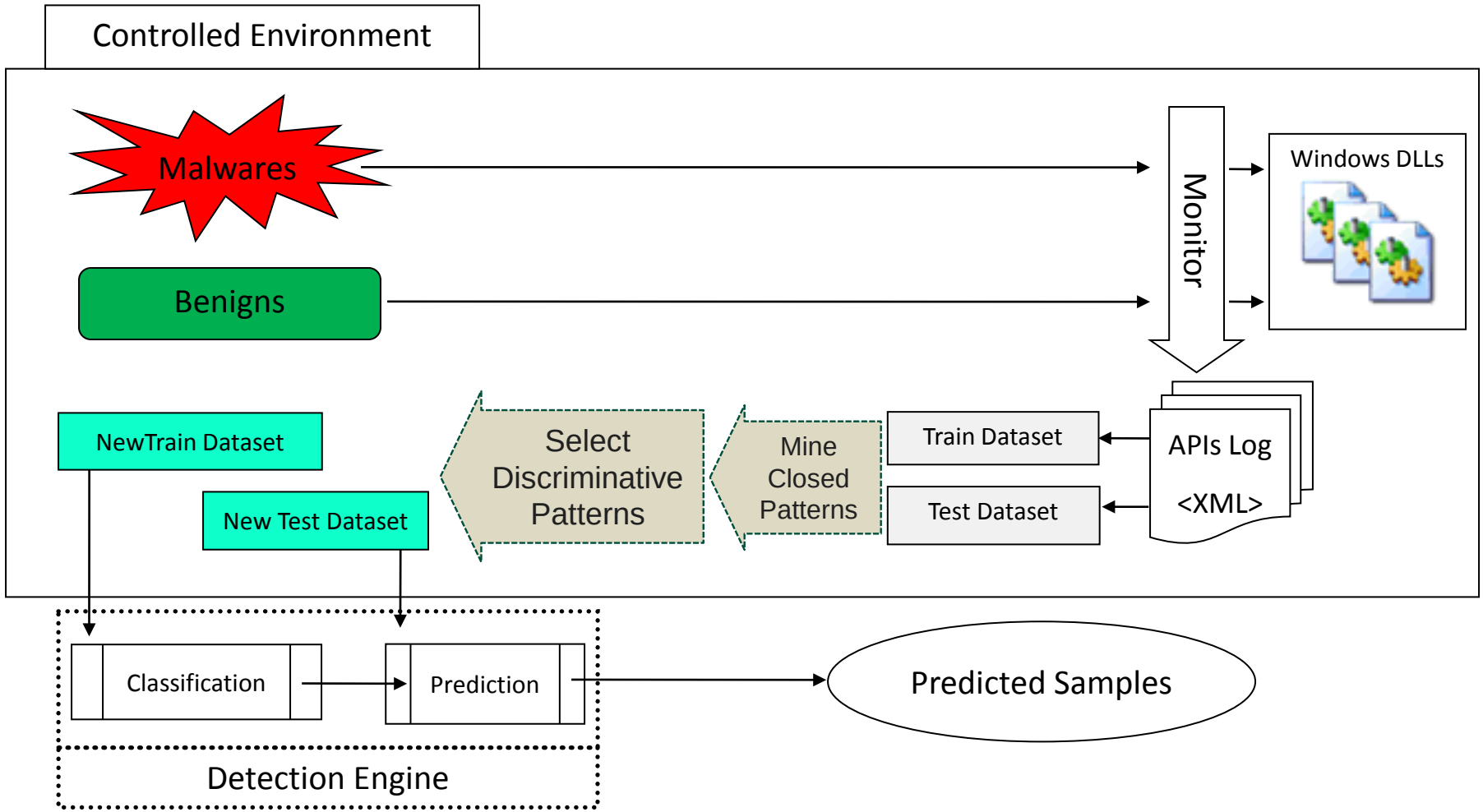
- Dynamic, Accuracy =91%
- Metamorphic Malwares replacing assembly instructions



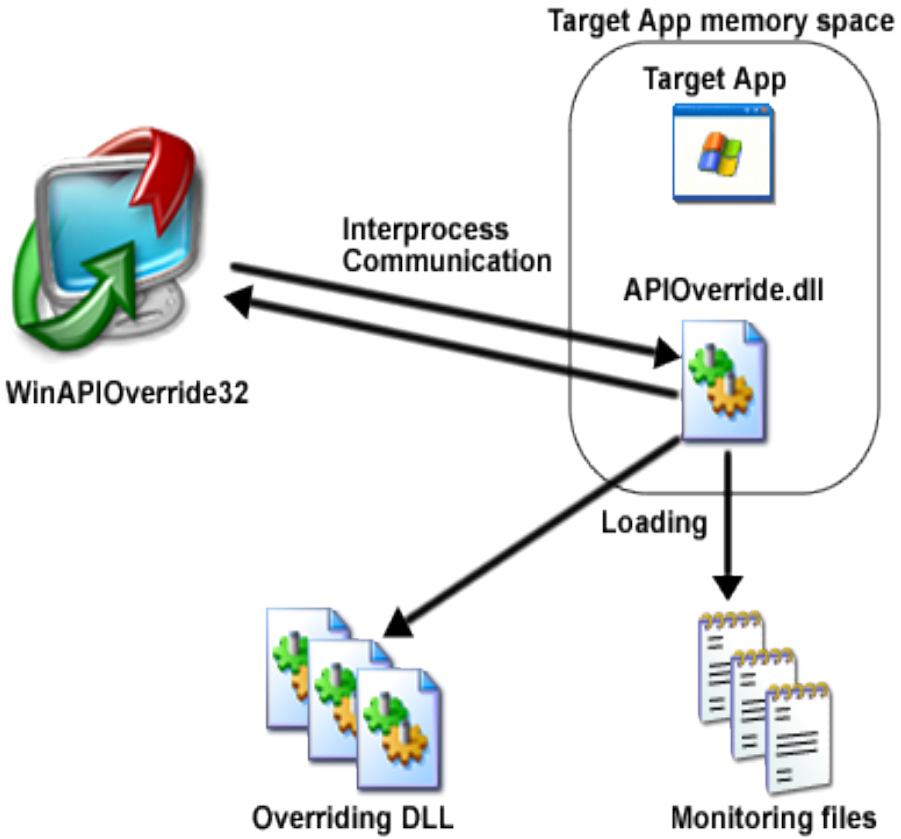
# Motivation

- Detecting polymorphic and metamorphic malware in case of obfuscation.
- Prior works focused on graph isomorphism which is computationally very expensive.
- We wanted to generate graphs to detect recurring patterns in malware runs.
- Graph data mining is also very computationally expensive
- So we mined iterative patterns in sequences
- Replacing API Calls to evade detection is not easily possible.

# Our Method


Controlled Environment
Malwares
Benigns
Monitor
Windows DLLs
APIs Log
<XML>
Train Dataset
Test Dataset
Mine Closed Patterns
Select Discriminative Patterns
NewTrain Dataset
New Test Dataset
Classification
Prediction
Predicted Samples
Detection Engine

# Monitoring




Backdoor.Win32.Agent.cy

RegOpenCurrentUser,OpenProcessToken,AllocateAndInitializeSid,CheckTokenMembership,FreeSid,RegOpenKeyExW,RegQueryValueExW,RegCloseKey,RegCloseKey,RegOpenKeyExW,RegOpenKeyExA,mmRegQueryValueExA,RegCloseKey,RegOpenKeyExW,RegOpenKeyExW,InitializeSecurityDescriptor,InitializeAcl,AddAccessAllowedAce,AddAccessAllowedAce,SetSecurityDescriptorDacl,MD4Init,MD4Update,MD4Update,MD4Update,MD4Final,OpenSCManagerA,OpenServiceA,CreateServiceA,StartServiceA,CloseServiceHandle,CloseServiceHandle

# Pattern Matching

- Consider a pattern P(<A,B>)
- Database consisting of two PEs

| Identifier | Sequence of API calls |
|---|---|
| Benign1 | <D, B, A, F, B, A, F, B, C, E> |
| Malware1 | <D, B, A, D, B, B, B, A, B> |

- Inst(P) denote as the set of instances of P :

  {(1, 3, 5), (1, 6, 8), (2, 3, 5), (2, 8, 9)}

- Multiple occurrences of an iterative pattern are considered to reflect **repetition of a behavior** (e.g. all the worms copy themselves)

# Closed Frequent Pattern

- *An iterative pattern P is **frequent** if its instances occur above a certain threshold of min_sup in APIDB, i.e. ,*

$$\left|Inst(P,APIDB)\right| \geq \min\_sup$$

- *A frequent iterative pattern P is **closed** if there exists no super sequence Q s.t. :*
  - *P and Q have the same support & Inst(P) ≈ Inst(Q)*
- To evaluate the discriminative power of a feature, statistical measure of **Fisher score** is adopted.

# Create Dataset

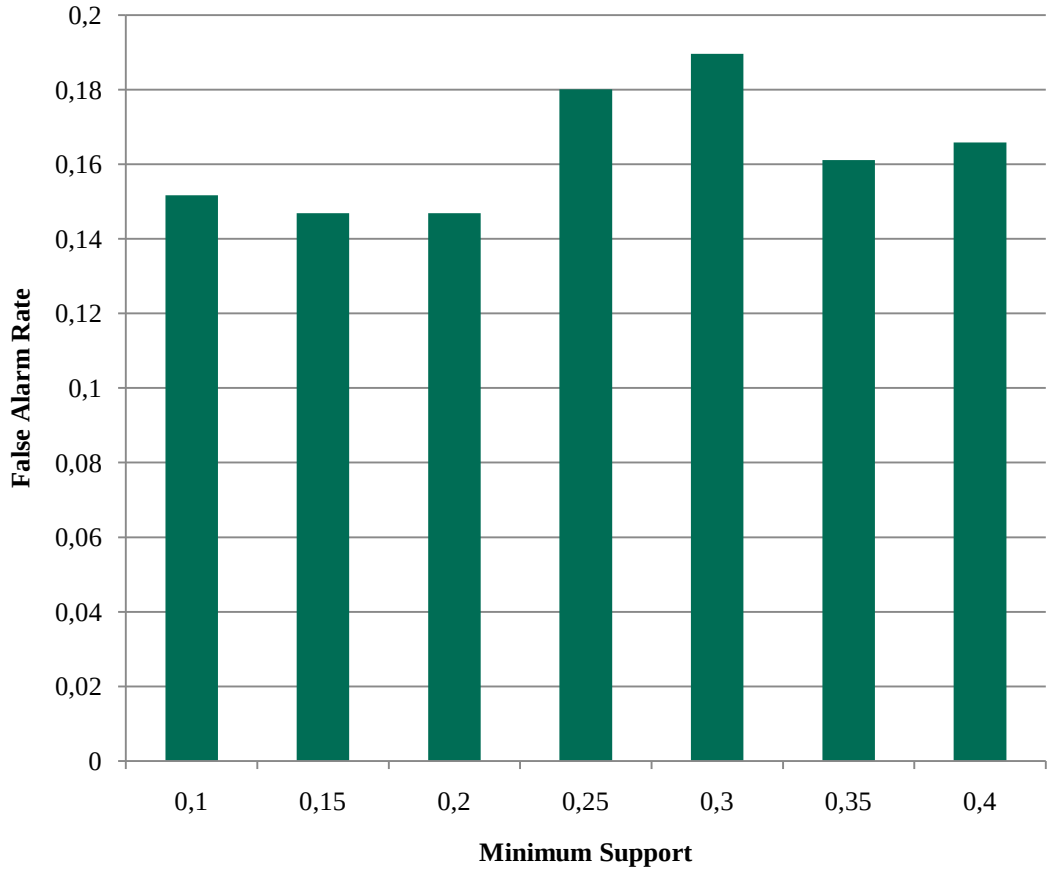
| Name Of PE | Single API1 | Single API2 | … | Closed Frequent API Pattern 1 | Closed Frequent API Pattern 2 | … |
|---|---|---|---|---|---|---|
| Benign 1 | Total of API1 | Total of API2 | … | Support | Support | … |
| Benign 2 | Total of API1 | Total of API2 | … | Support | Support | … |
| … | … | … | … | … | … | … |
| Malware 1 | Total of API1 | Total of API2 | … | Support | Support | … |
| Malware 2 | Total of API1 | Total of API2 | … | Support | Support | … |
| … | … | … | … | … | … | … |

# Experiments

- Random Forest with 10 Fold Cross Validation
- Results on 269 malware and 211 benign

| S | TP | FP | TN | FN | DR | ACC |
|---|---|---|---|---|---|---|
| 0.1 | 243 | 26 | 179 | 32 | 90.33 | 87.91 |
| 0.15 | 247 | 22 | 180 | 31 | 91.8 | 88.95 |
| 0.2 | 247 | 22 | 180 | 31 | **91.8** | 88.95 |
| 0.25 | 245 | 24 | 173 | 38 | 91.1 | 87.08 |
| 0.3 | 247 | 22 | 171 | 40 | 91.8 | 87.02 |
| 0.35 | 242 | 27 | 177 | 34 | 90 | 87.29 |
| 0.4 | 243 | 26 | 176 | 35 | 90.3 | 87.29 |

# False alarm rate



# Conclusion & future works

- Due to rapid growth and increasing malware, need for a mechanized system for detecting malware are bound.
- Anti-virus signature-based methods are reliable but not enough.
- Adding detection methods based on artificial intelligence can improve performance of anti-virus.
- We presented a method that find the best patterns for distinguishing malware and benign.
- We want to monitor more PEs with more DLLs and use complex structures for improving the results.

# References


[1] Ashkan Sami, Hossein Rahimi , Babak Yadegari, Naser Peiravian, Sattar Hashemi, Ali Hamze, "Malware Detection Based On Mining API Calls,". the *ACM Symposium on Applied Computing-Data Mining Track*, Sierre, Switzerland, 2010.

[2] P. M. C. Clemens Kolbitsch, Christopher Kruegel, Engin Kirda, Xiaoyong Zhou and XiaoFeng Wang "Effective and Efficient Malware Detection at the End Host," presented at the 18th Usenix Security Symposium, 2009

[3] R. G. a. J. L. J. Dai, "Efficient Virus Detection Using Dynamic Instruction Sequences," *Journal of Computers,* 2009

[4] Christoph Csallner, Yannis Smaragdakis, Tao Xie. DSD-Crasher: A Hybrid Analysis Tool for Bug Finding. *ACM Transactions on Software Engineering and Methodology*, Vol. 17, Issue 2, pp. 345-371, July 2008.

[5] David Lo, Siau-Cheng Khoo and Chao Liu. Efficient Mining of Iterative Patterns for Software Specification Discovery. *13th SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*. California. Aug 12-15, 2007

[6] M. Christodorescu, Jha, S., Seshia, S., Song,D.,Bryant,R., "Semantics-aware malware detection,". the *IEEE Symposiumon Security and Privacy*, 2005.

[7] POTIER, J. 2010. *WinAPIOverride32* [Online]. Available: http://jacquelin.potier.free.fr/winapioverride32/.

[8] M. C. a. S. Jha, "Static analysis of executables to detect malicious patterns," *USENIX Security Symposium*, 2003.

[9] T. Yetiser, "Polymorphic Viruses – Implementation, detection, and protection," 1993.

[10] V. Heavens. *computer virus collection*. Available: http://vx.netlux.org/vl.php

# Thank You

## Iterative System Call Patterns Blow the Malware Cover

Mansour Ahmadi, Ashkan Sami, Hossein Rahimi, Babak Yadegari,
MansourAhmadi@gmail.com, asami@ieee.org
Shiraz University, Iran




"IT Security for the Next Generation"
Asia Pacific & MEA Cup 2011
University of Technology MARA, Malaysia
March 4- 6, 2011