NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
Department of Computer Science and Engineering

ARTIFICIAL INTELLIGENCE IN ANTI-VIRUS TECHNOLOGY

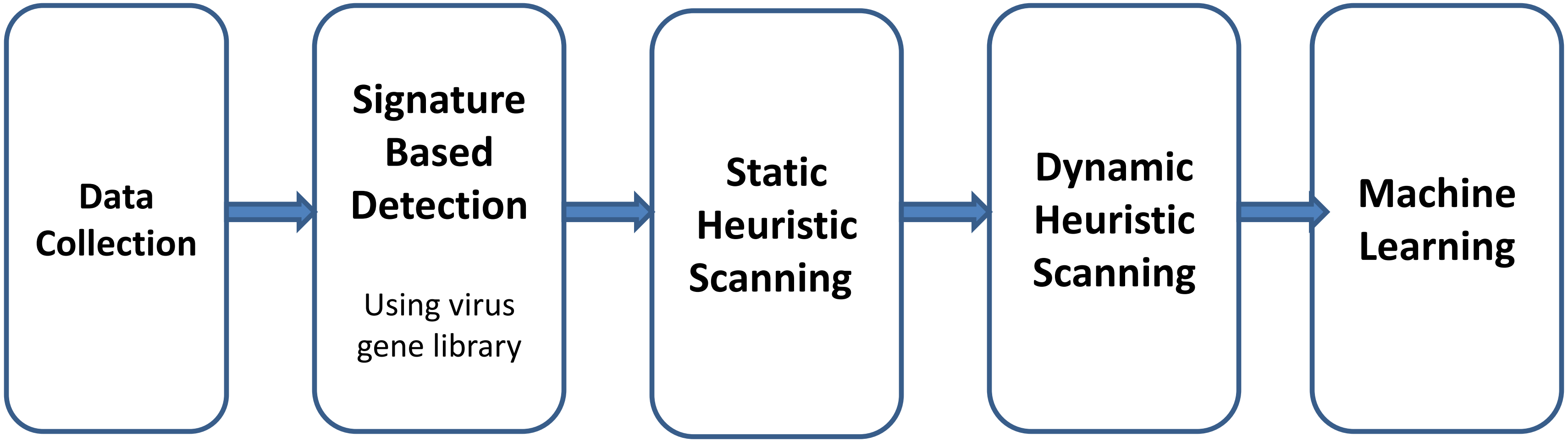
Guide : Mr K Vinay Kumar

By:	**Team 09UG19**	
	Madhuri Shanbhogue	**(09CO47)**
	Adarsh Pradhan JMT	**(09CO04)**
	M C Harshavardhana	**(09CO45)**

PROBLEM DEFINITION AND OBJECTIVE

- Problem Definition: Currently used AI techniques detect large number of false positives

- Objective: Reduce the number of false positives using Learning techniques.



Data Collection

Signature Based Detection

Using virus gene library

Static Heuristic Scanning

Dynamic Heuristic Scanning

Machine Learning

SIGNATURE BASED DETECTION

- Use of virus gene library

- 3,00,000 signatures

- Advantages
  - Fast
  - Available signatures

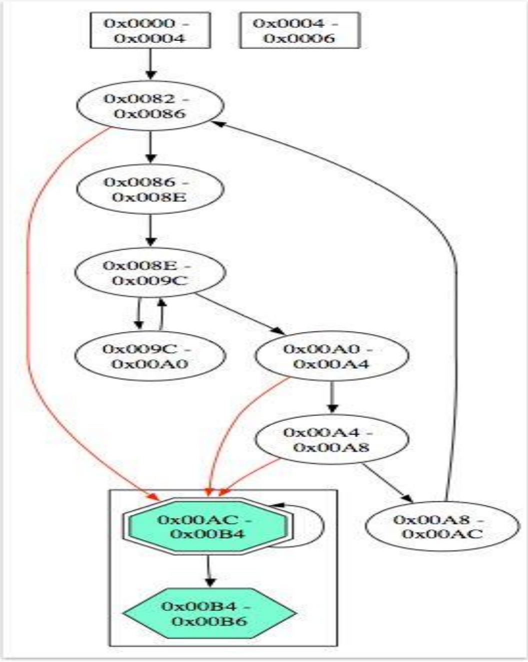
- Disadvantages
  - Cannot detect viruses without signatures

STATIC HEURISTIC SCANNING

- Search for behavior by static code analysis
- Challenges
  - Metamorphic (Obfuscation)
  - Polymorphic (Encryption)
- Obfuscation types
  - Register reassignment
  - Dead Code Insertion
  - Code Transposition
  - Code Substitution
- Obfuscation Handling
  - Regular Expression
  - Control Flow Graph
- Advantages
  - Not computation intensive

Code snippet

```
E800 0000 00(90)*
5B(90)* 8D4B
42(90)* 51(90)*
50(90)* 50(90)*
0F01 4C24 FE(90)*
5B(90)* 83C3 C(90)*
FA(90)*8B2B
```



DYNAMIC HEURISTIC SCANNING

- Decision parameters
  - System calls
  - Unusual destination
  - Analysis of file types and file system
  - Analysis of memory usage – buffer overflow analysis, system registery
  - Access of executables, mutex
  - Access of disk
  - Replication
  - Attempts to hide other files
  - Attempts to terminate programs
  - Attempts to open other executables.

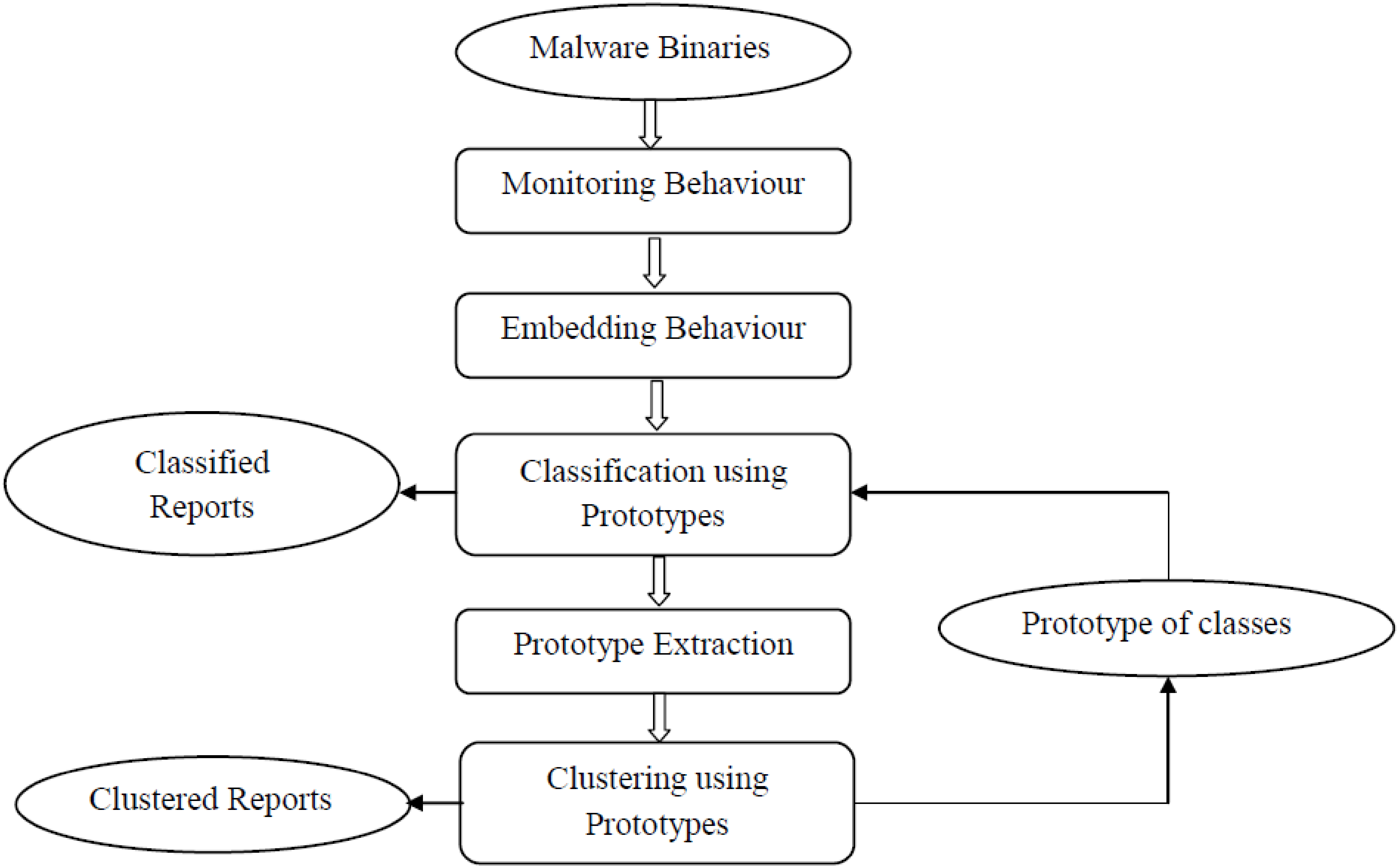
DYNAMIC HEURISTIC SCANNING

- Implementation Details
  - Sandbox Environment
  - Reports of malware binaries

- Advantages
  - Can detect encrypted viruses
  - Sandboxing helps in CPU emulation without affecting the system

- Disadvantages
  - CPU Emulation is expensive

MACHINE LEARNING



MACHINE LEARNING

- Prototype Extraction – Gonzalez algorithm – O (kn)
  - K – number of prototypes, n – number of reports
- Clustering using prototypes
  - Complete linkage - O(k^2 log k + n)

  v/s
  - hierarchical clustering - O(n^2 log n)
  - Speed-up factor of square root n/k
- Classification
  - Clustered malware classes used for training and learning behavior
  - O (kn)
- Incremental analysis
  - Better than batch analysis
  - Uses prototypes stored from previous runs
  - O(nm + k^2 log k)
    - M – number of prototypes from previous run

RESULTS AND ANALYSIS

	Signature Matching	Static Heuristic Scanning	Dynamic Heuristic Scanning	Artificial Learning
Detection of known virus	✓	✓	✓	✓
Detection of unknown viruses	Fails when signature is unavailable	✓	✓	62%
Robustness	Fails when signature is unavailable	✓	✓✓	✓
False positives	No false positives	✓	✓	Reduces false positives
High speed detection	✓✓	✓	Requires CPU emulation	Learning algorithms consume time
Detect metamorphic/ oligomorphic viruses	Fails since virus encrypts itself	✓	✓✓	Efficient only after detection by heuristic scanning
Obfuscation	Fails since virus obfuscates itself	✓	✓✓	After heuristic scanning

RESULTS AND ANALYSIS

Technique	F-measure
Clustering using Prototype	0.950
Classification using Prototype	0.981
Classification using SVM and XML	0.807

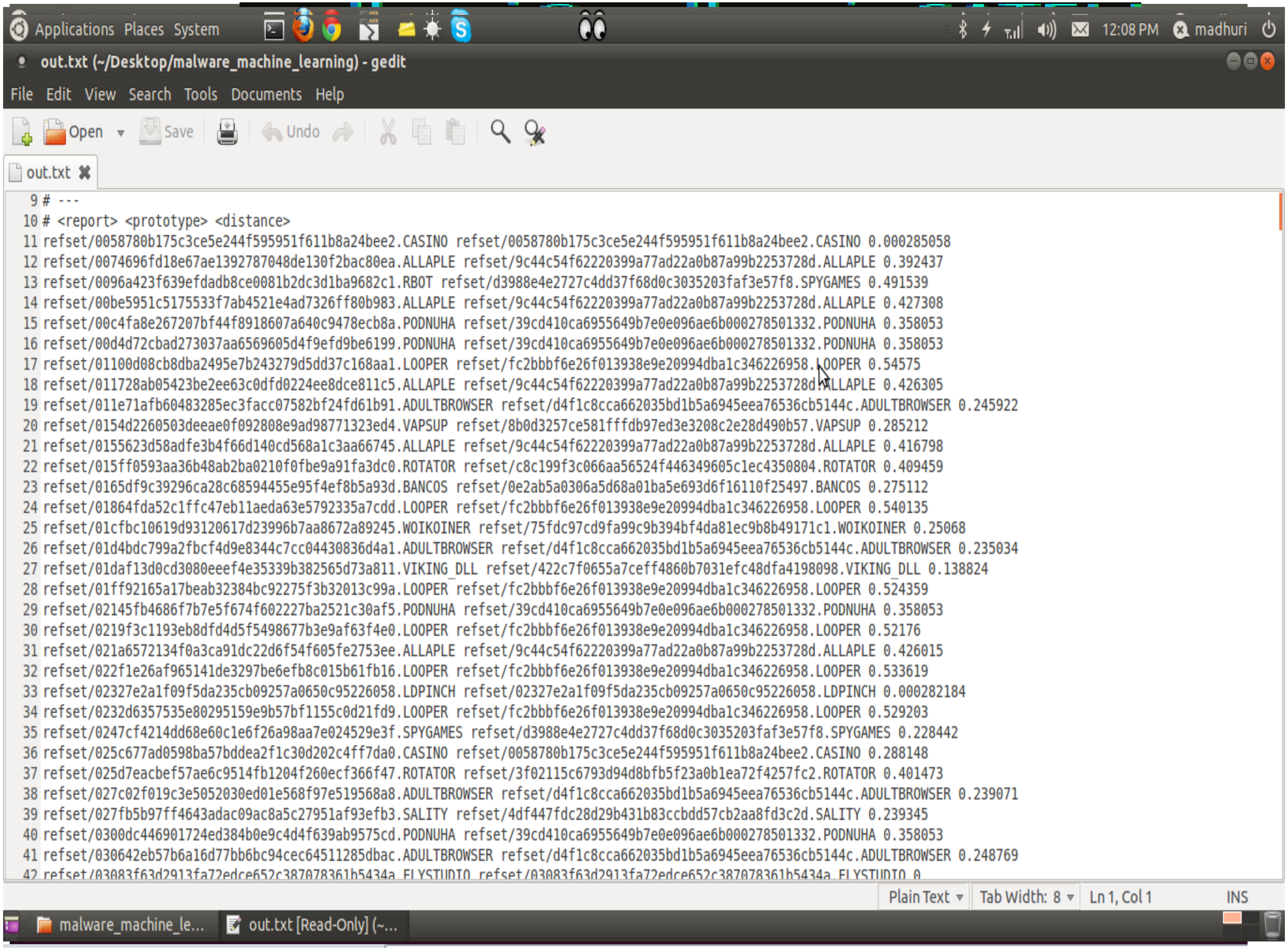
$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- Precision - how well individual clusters agree with malware classes
- Recall - extent to which classes are scattered across clusters
- Inverse relation between Precision and Recall
- F-measure – Combines Precision and Recall
  - 1 => perfect classification
  - 0 => completely incorrect classification

SCREENSHOTS


Applications Places System
12:08 PM madhuri
out.txt (~/Desktop/malware_machine_learning) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
out.txt
Plain Text
Tab Width: 8
Ln 1, Col 1
INS
malware_machine_le...
out.txt [Read-Only] (~...

```
# ---
# <report> <prototype> <distance>
refset/0058780b175c3ce5e244f595951f611b8a24bee2.CASINO refset/0058780b175c3ce5e244f595951f611b8a24bee2.CASINO 0.000285058
refset/0074696fd18e67ae1392787048de130f2bac80ea.ALLAPLE refset/9c44c54f62220399a77ad22a0b87a99b2253728d.ALLAPLE 0.392437
refset/0096a423f639efdadb8ce0081b2dc3d1ba9682c1.RBOT refset/d3988e4e2727c4dd37f68d0c3035203faf3e57f8.SPYGAMES 0.491539
refset/00be5951c5175533f7ab4521e4ad7326ff80b983.ALLAPLE refset/9c44c54f62220399a77ad22a0b87a99b2253728d.ALLAPLE 0.427308
refset/00c4fa8e267207bf44f8918607a640c9478ecb8a.PODNUHA refset/39cd410ca6955649b7e0e096ae6b000278501332.PODNUHA 0.358053
refset/00d4d72cbad273037aa6569605d4f9efd9be6199.PODNUHA refset/39cd410ca6955649b7e0e096ae6b000278501332.PODNUHA 0.358053
refset/01100d08cb8dba2495e7b243279d5dd37c168aa1.LOOPER refset/fc2bbbf6e26f013938e9e20994dba1c346226958.LOOPER 0.54575
refset/011728ab05423be2ee63c0dfd0224ee8dce811c5.ALLAPLE refset/9c44c54f62220399a77ad22a0b87a99b2253728d.ALLAPLE 0.426305
refset/011e71afb60483285ec3facc07582bf24fd61b91.ADULTBROWSER refset/d4f1c8cca662035bd1b5a6945eea76536cb5144c.ADULTBROWSER 0.245922
refset/0154d2260503deeae0f092808e9ad98771323ed4.VAPSUP refset/8b0d3257ce581fffdb97ed3e3208c2e28d490b57.VAPSUP 0.285212
refset/0155623d58adfe3b4f66d140cd568a1c3aa66745.ALLAPLE refset/9c44c54f62220399a77ad22a0b87a99b2253728d.ALLAPLE 0.416798
refset/015ff0593aa36b48ab2ba0210f0fbe9a91fa3dc0.ROTATOR refset/c8c199f3c066aa56524f446349605c1ec4350804.ROTATOR 0.409459
refset/0165df9c39296ca28c68594455e95f4ef8b5a93d.BANCOS refset/0e2ab5a0306a5d68a01ba5e693d6f16110f25497.BANCOS 0.275112
refset/01864fda52c1ffc47eb11aeda63e5792335a7cdd.LOOPER refset/fc2bbbf6e26f013938e9e20994dba1c346226958.LOOPER 0.540135
refset/01cfbc10619d93120617d23996b7aa8672a89245.WOIKOINER refset/75fdc97cd9fa99c9b394bf4da81ec9b8b49171c1.WOIKOINER 0.25068
refset/01d4bdc799a2fbcf4d9e8344c7cc04430836d4a1.ADULTBROWSER refset/d4f1c8cca662035bd1b5a6945eea76536cb5144c.ADULTBROWSER 0.235034
refset/01daf13d0cd3080eeef4e35339b382565d73a811.VIKING_DLL refset/422c7f0655a7ceff4860b7031efc48dfa4198098.VIKING_DLL 0.138824
refset/01ff92165a17beab32384bc92275f3b32013c99a.LOOPER refset/fc2bbbf6e26f013938e9e20994dba1c346226958.LOOPER 0.524359
refset/02145fb4686f7b7e5f674f602227ba2521c30af5.PODNUHA refset/39cd410ca6955649b7e0e096ae6b000278501332.PODNUHA 0.358053
refset/0219f3c1193eb8dfd4d5f5498677b3e9af63f4e0.LOOPER refset/fc2bbbf6e26f013938e9e20994dba1c346226958.LOOPER 0.52176
refset/021a6572134f0a3ca91dc22d6f54f605fe2753ee.ALLAPLE refset/9c44c54f62220399a77ad22a0b87a99b2253728d.ALLAPLE 0.426015
refset/022f1e26af965141de3297be6efb8c015b61fb16.LOOPER refset/fc2bbbf6e26f013938e9e20994dba1c346226958.LOOPER 0.533619
refset/02327e2a1f09f5da235cb09257a0650c95226058.LDPINCH refset/02327e2a1f09f5da235cb09257a0650c95226058.LDPINCH 0.000282184
refset/0232d6357535e80295159e9b57bf1155c0d21fd9.LOOPER refset/fc2bbbf6e26f013938e9e20994dba1c346226958.LOOPER 0.529203
refset/0247cf4214dd68e60c1e6f26a98aa7e024529e3f.SPYGAMES refset/d3988e4e2727c4dd37f68d0c3035203faf3e57f8.SPYGAMES 0.228442
refset/025c677ad0598ba57bddea2f1c30d202c4ff7da0.CASINO refset/0058780b175c3ce5e244f595951f611b8a24bee2.CASINO 0.288148
refset/025d7eacbef57ae6c9514fb1204f260ecf366f47.ROTATOR refset/3f02115c6793d94d8bfb5f23a0b1ea72f4257fc2.ROTATOR 0.401473
refset/027c02f019c3e5052030ed01e568f97e519568a8.ADULTBROWSER refset/d4f1c8cca662035bd1b5a6945eea76536cb5144c.ADULTBROWSER 0.239071
refset/027fb5b97ff4643adac09ac8a5c27951af93efb3.SALITY refset/4df447fdc28d29b431b83ccbdd57cb2aa8fd3c2d.SALITY 0.239345
refset/0300dc446901724ed384b0e9c4d4f639ab9575cd.PODNUHA refset/39cd410ca6955649b7e0e096ae6b000278501332.PODNUHA 0.358053
refset/030642eb57b6a16d77bb6bc94cec64511285dbac.ADULTBROWSER refset/d4f1c8cca662035bd1b5a6945eea76536cb5144c.ADULTBROWSER 0.248769
refset/03083f63d2913fa72edce652c387078361b5434a.FLYSTUDIO refset/03083f63d2913fa72edce652c387078361b5434a.FLYSTUDIO 0
```

References

- Chamorro, E.; Jianchao Han; Beheshti, M.; , "The Design and Implementation of an Antivirus Software Advising System," *Information Technology: New Generations (ITNG), 2012 Ninth International Conference on* , vol., no., pp.612-617, 16-18 April 2012

- Wei Wang; Pengtao Zhang; Ying Tan; Xingui He; , "A Hierarchical Artificial Immune Model for Virus Detection," *Computational Intelligence and Security, 2009. CIS '09. International Conference on* , vol.1, no., pp.1-5, 11-14 Dec. 2009

- Xiao-bin Wang; Guang-yuan Yang, Yi-chao Li, Dan Liu (2008). "Review on the application of artificial intelligence in antivirus detection system," *Proc of 2008 IEEE Conference on Cybernetics and Intelligent Systems*, 506-509.

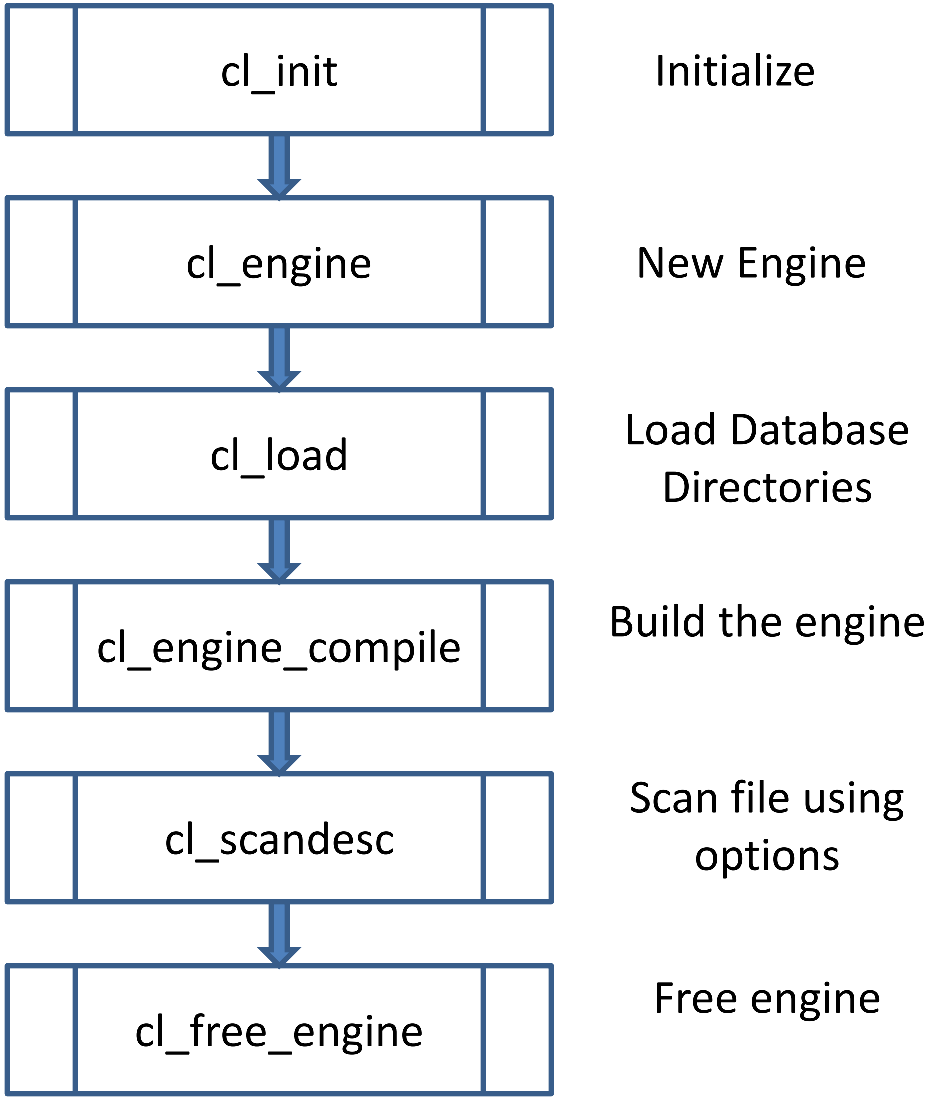
- Charles P. Pfleeger, Shari Lawrence Pfleeger. “Program Security,” in *Security in Computing*, 3rd ed., Prentice Hall, pp.15-67, Dec. 2002.

- *ClamAV User Manual,*2007 - 2011 Sourcefire, Inc. Authors: Tomasz Kojm

- Symantec Corporation, Understanding Heuristics (1997). “Symantec’s Bloodhound Technology,” *Symantec White Paper Series*.

Appendix 1 - Implementation of Signature Based Detection

- Use of ClamAV engine
- Provides an API – libClamAV
- Provides an in-memory database of signatures
- Provides regular updates to two database files
    - main.cvd
    - daily.cvd



cl_init — Initialize

cl_engine — New Engine

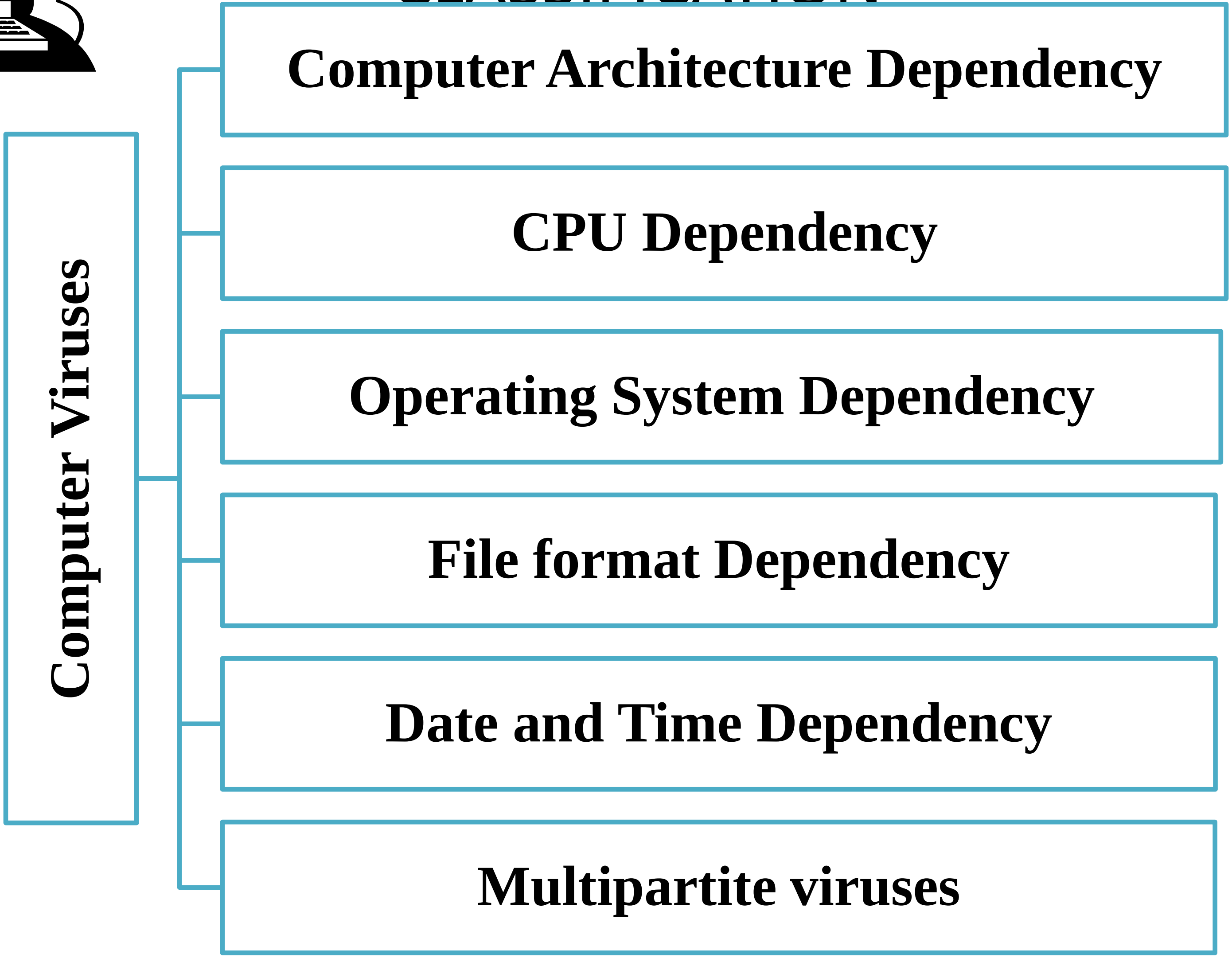
cl_load — Load Database Directories

cl_engine_compile — Build the engine

cl_scandesc — Scan file using options

cl_free_engine — Free engine

Appendix 2 - DEPENDENCY BASED CLASSIFICATION



Computer Viruses

Computer Architecture Dependency

CPU Dependency

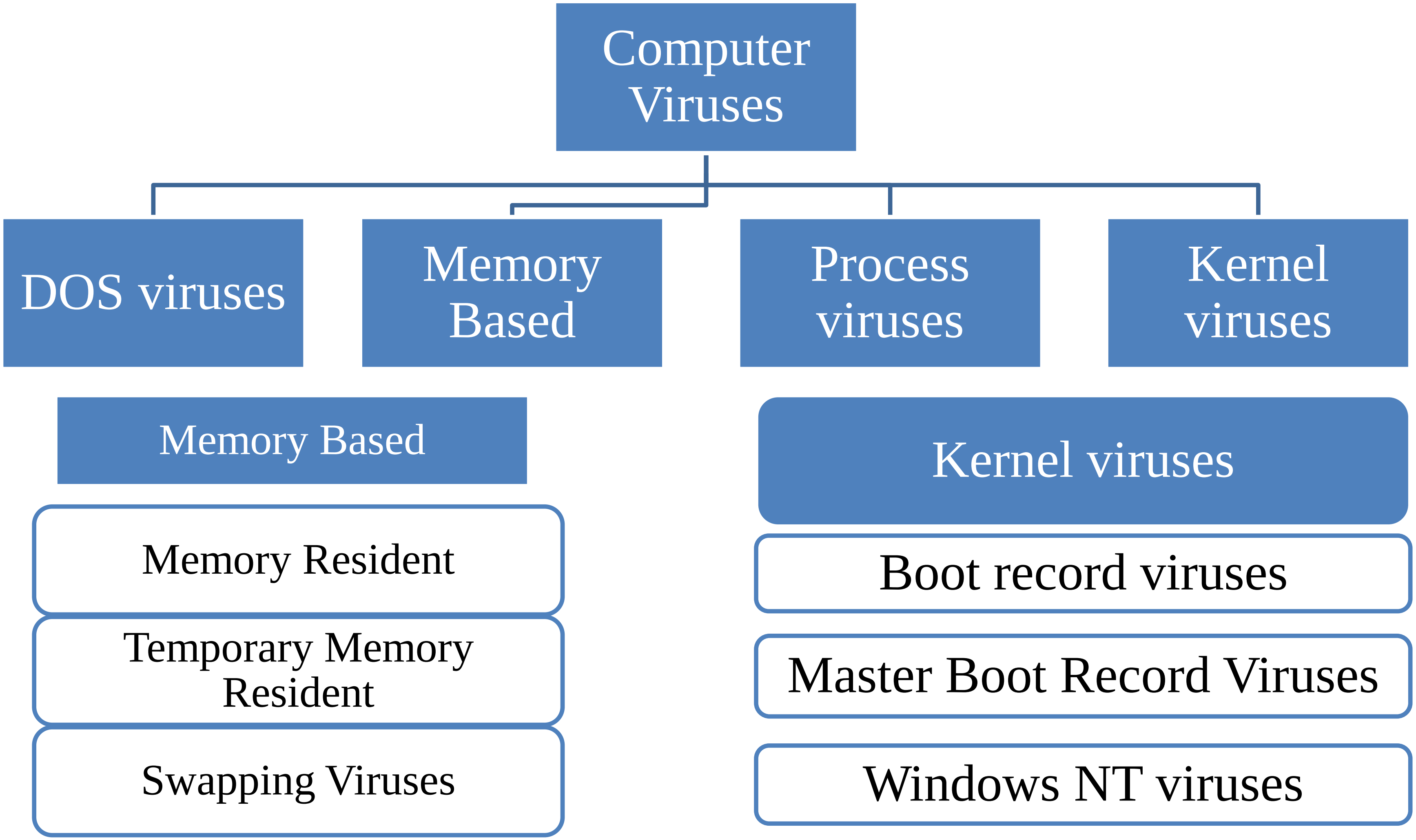
Operating System Dependency

File format Dependency

Date and Time Dependency

Multipartite viruses

Appendix 3 - BEHAVIOR BASED CLASSIFICATION



Computer Viruses

DOS viruses

Memory Based

Process viruses

Kernel viruses

Memory Based

- Memory Resident
- Temporary Memory Resident
- Swapping Viruses

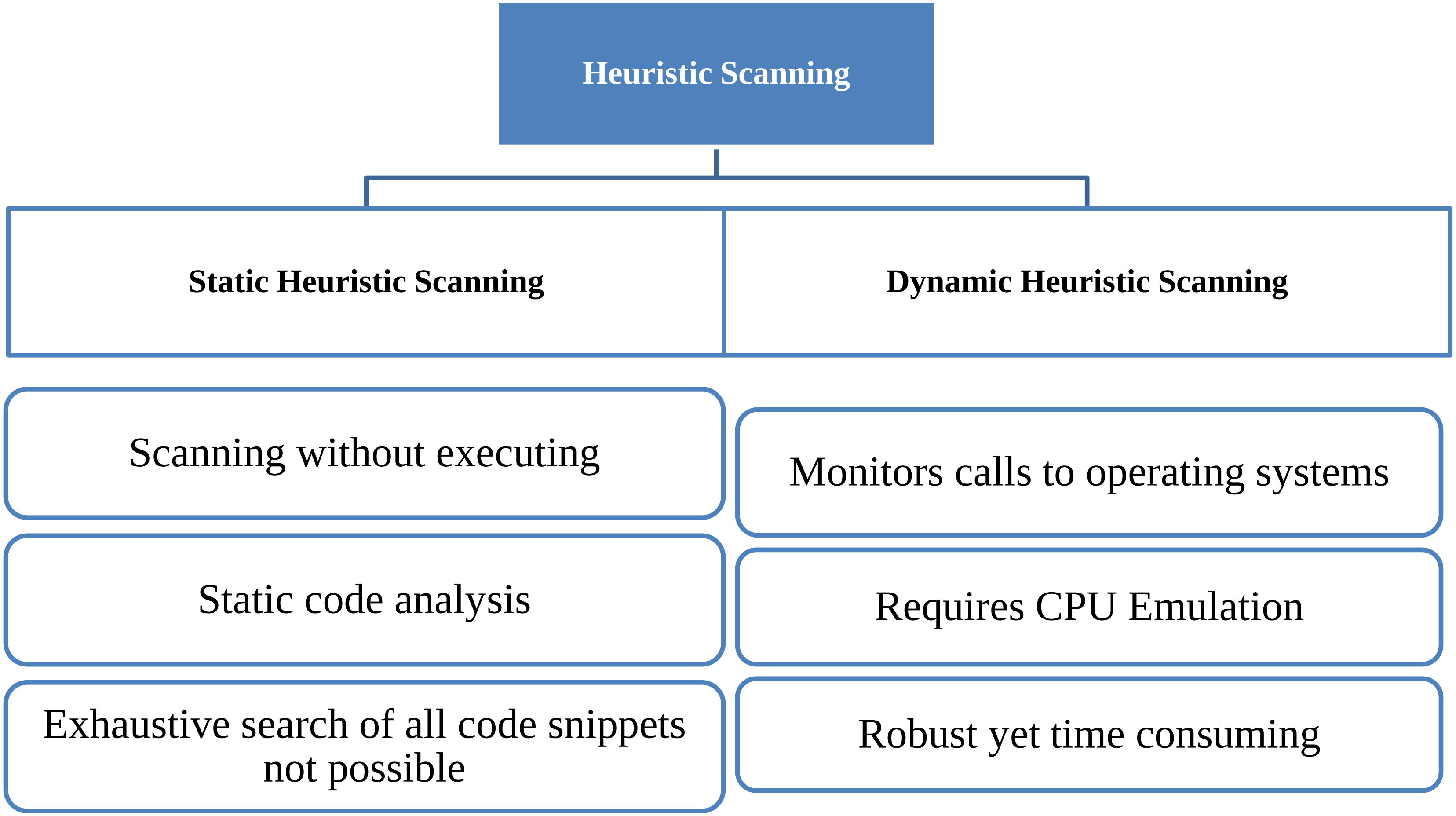
Kernel viruses

- Boot record viruses
- Master Boot Record Viruses
- Windows NT viruses

Appendix 4 - HEURISTIC SCANNING PARAMETERS

- Output using sample data
- Unusual destination
- File types and File System
- Memory Usage – Buffer overflow analysis, system registry
- Access of executables
- Access of disk
- Replication
- Attempts to hide other files
- Source code content matched using wild card characters

Appendix 5 - IMPLEMENTATION OF HEURISTIC SCANNING



Heuristic Scanning

Static Heuristic Scanning	**Dynamic Heuristic Scanning**
Scanning without executing	Monitors calls to operating systems
Static code analysis	Requires CPU Emulation
Exhaustive search of all code snippets not possible	Robust yet time consuming