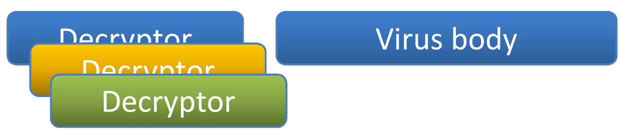
Classification (and Detection) of Metamorphic Malware Using Value Set Analysis

Felix Leder – leder@cs.uni-bonn.de
Bastian Steinbock – steinboc@cs.uni-bonn.de
Peter Martini – martini@cs.uni-bonn.de

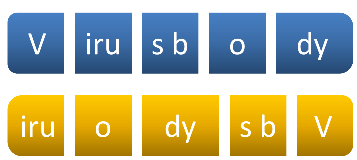
universitätbonn CS 0100

Encrypted malware: Encrypted virus body is decrypted at run-time (example: UPX)

Polymorphic malware: Morphing/varying decryptor stub



Metamorphic malware: Morphes the whole virus body.



Informatik IV – leder@cs.uni-bonn.de

universitätbonn CS

Metamorphic Malware is **hardly detectable with regular string signatures**.

Virus scanners use customized detection engines for each family.

Problem:

In 2009 Symantec detected more than 2.8 million new malware specimen. (170% growth)

Impossible to analyze every sample or by hand. Pre-classification needed.

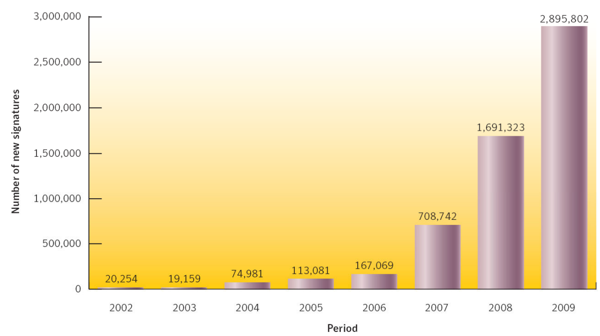


Figure 10. New malicious code signatures
Source: Symantec.

Bad detection example – Lexotan32:

- File infecting virus from 2002
- Virus total detection rate in 2009: 12.9%
- None of 40 scanners detected all of the samples

Informatik IV – leder@cs.uni-bonn.de

- Code changes completely
- Common subsequences have sizes of max. 5 bytes
- Every infection looks completely different

```
1: jmp 4
2: reg_2 = reg_1+2
3: jmp …
4: reg_1 = 5
5: jmp 2
```

```
reg_1 = 5
reg_2 = reg_1+2
```

Result always 7

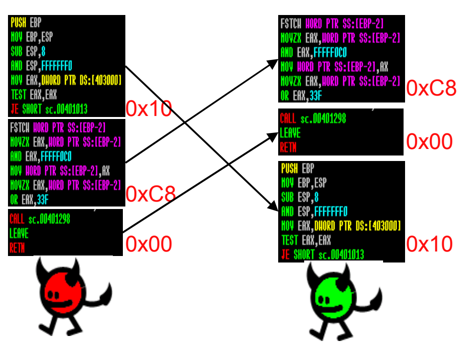
```
reg_3 = 5
reg_2 = reg_3+2
```

```
reg_1 = 5
nop
reg_2 = reg_1+2
```

```
reg_1 = 5
reg_2 = -(-reg_1-2)
```

Informatik IV – leder@cs.uni-bonn.de

- While structure changes, behavior has to stay the same
- Existing approaches:
  - Code normalization: Standard representations
    - metamorphism can be very complex
    - only shown for W32/Evol, self-made examples
  - Execution traces / blackboxing: Possible but easy to defeat (Waiting, changing execution order, environment detection)



Informatik IV – leder@cs.uni-bonn.de

- **Static analysis** investigates whole sample without execution
- Behavior is reflected by **values** / memory contents
- Each program contains characteristic values it cannot change:

```
For I = 0 to 1000 do:
    …
```

```
socket(AF_INET = 2,
       SOCK_STREAM = 6,
       PF_INET = 2)
sockaddr_in.sin_port = 80
connect(…)
```

```
If var_1 > 9:
   var_1 = 10
```


Informatik IV – leder@cs.uni-bonn.de

Methodology

VALUE SET ANALYSIS

Informatik IV – leder@cs.uni-bonn.de

Value Set Analysis - VSA:

“What values are possible for a specific variable/memory-location at a specific location inside the program?”

- **Static data flow tracking** and approximate memory contents
- Scalability: Over-approximate when too complex

```
Start (0x100):
      mov eax, 1
      mov ebx, start
      add eax, ebx
```

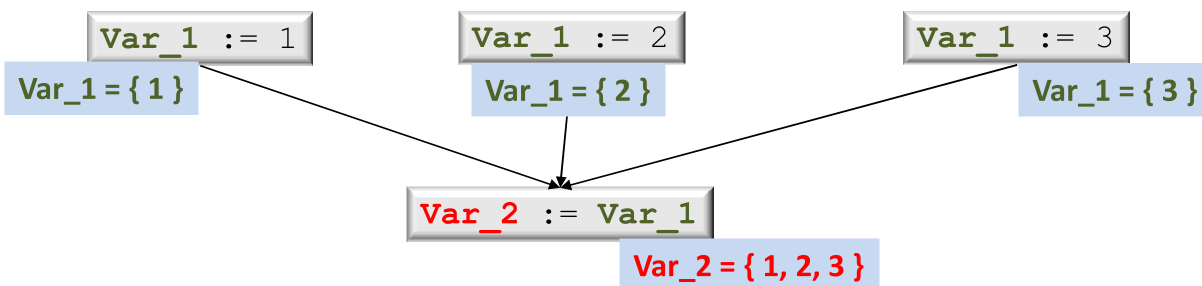
Value Sets:

eax = {1}; ebx = {}

eax = {1}; ebx = {0x100}

eax = {0x101}; ebx = {0x100}

Informatik IV – leder@cs.uni-bonn.de



Informatik IV – leder@cs.uni-bonn.de

```
Var_1 := 1
```

Var_1 = { 1 }

```
Var_1 := 2
```

Var_1 = { 2 }

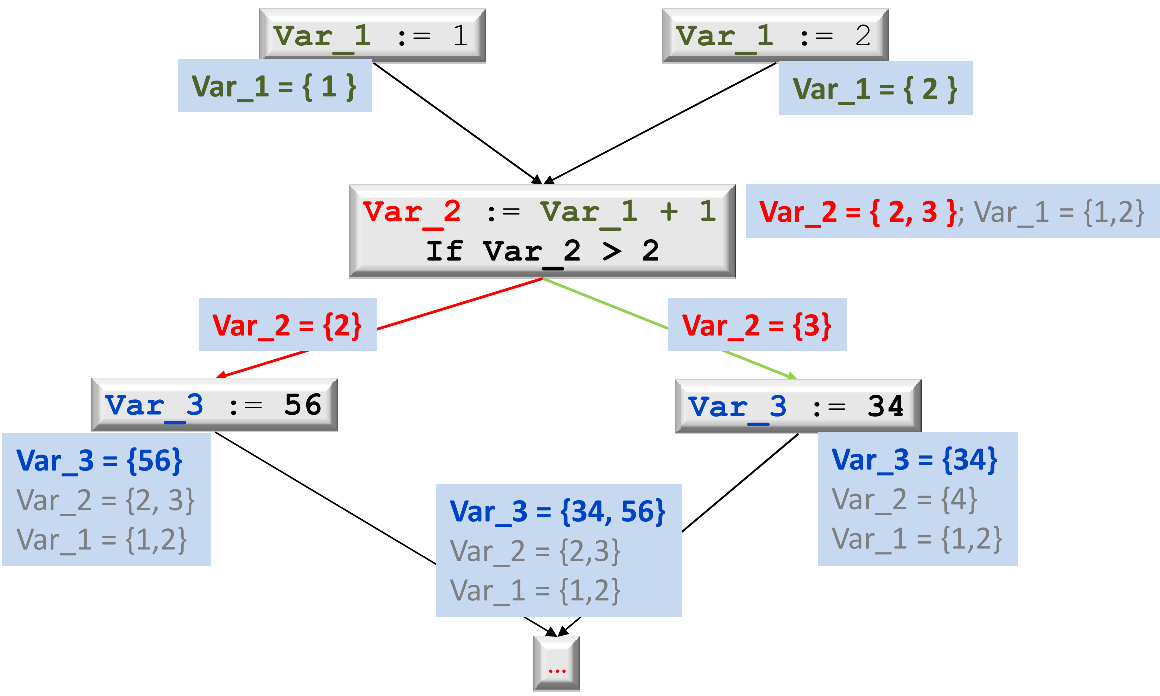
```
Var_1 := 3
```

Var_1 = { 3 }

```
Var_2 := Var_1 + 1
Var_3 := Var_1 * 2
```

Var_2={2,3,4}; Var_1={1,2,3}
Var_3={2,4,6}; Var_2={2,3,4}; Var_1={1,2,3}

Informatik IV – leder@cs.uni-bonn.de



Informatik IV – leder@cs.uni-bonn.de

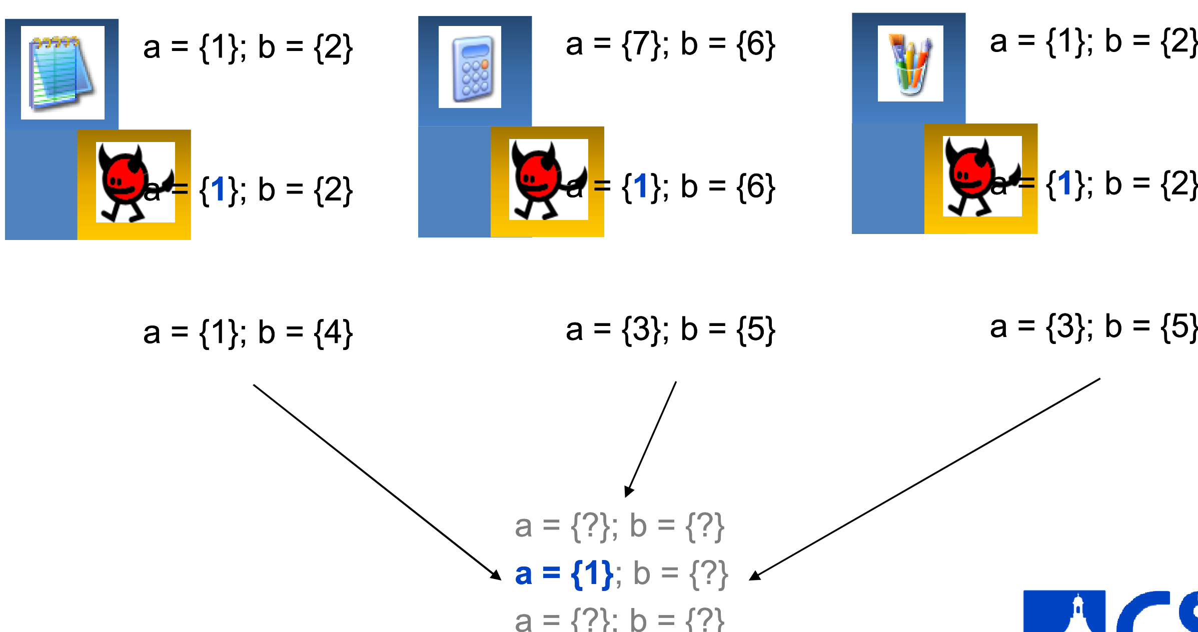
universitätbonn CS

Determine Characteristic Value Sets

REFINEMENT

Informatik IV – leder@cs.uni-bonn.de

- Metamorphic malware is often file infecting
- Challenge: Distinguish host/malware Value Sets



METHODOLOGY

Informatik IV – leder@cs.uni-bonn.de

eax = {1, 5, 10} | ebx = {0, 1, 2}

Value Set

Data Objects

File

```
00401436 sub     esp, 8
00401439 sub     esp, 8
0040143C mov     eax, [ebp+var_4]
0040143F call    eax
00401441 add     esp, 8
00401444 push    eax
```


Informatik IV – leder@cs.uni-bonn.de

universitätbonn CS

- **Matching:** How to quantify the **similarity** of two …

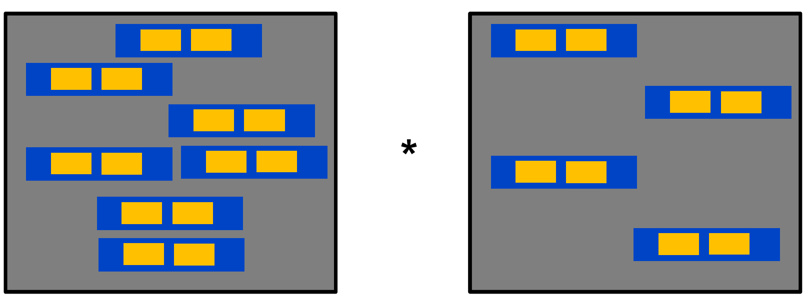
Data Objects

eax = {1, 2, 3} * eax = {1, 2, 99} = ?? %

Value Sets

[eax = {…} | ebx = {…}] * [[sp-4] = {} | esi = {…}] = ?? %

Files



= ?? %

universitätbonn CS 0100

Informatik IV – leder@cs.uni-bonn.de

Different matching strategies possible:

- **Average matching:** Score = % of equal elements
- **Threshold matching:** Score = $\begin{cases} 1 & \text{if \% of equal elements} > \Delta \\ 0 & \text{otherwise} \end{cases}$

eax = {1, 2, 3} * **eax = {1, 2, 99}** = **66** %

Average score : 66%

Threshold score with (Δ =0.6) : 100%

Threshold score with (Δ =0.7) : 0 %

Have to be used for all layers: Data Obj., Value Sets, Files

universitätbonn
Informatik IV – leder@cs.uni-bonn.de

Penalties for (unsimilar) Data Objects - Data Object Adjustments

- |Refined set | < |Infected set|

eax = {**1**} * eax = {**1,3,5,10,99,1010,445,110,22,1337**} ➔ score - x%

- Location of the value (stack, heap, global memory)

[global_12345678] = {13} * **[esp-4]** = {13} ➔ score - y%

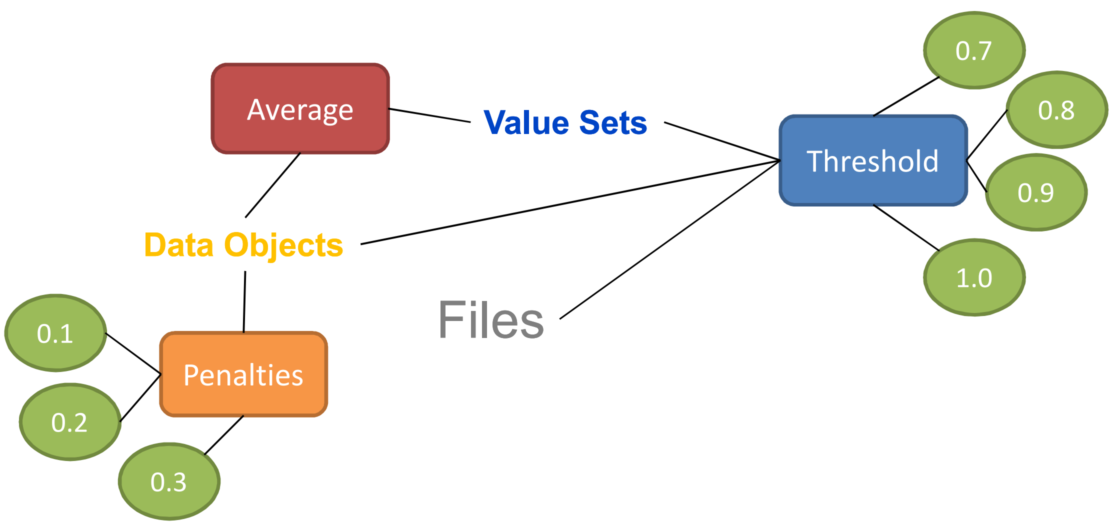
- Similar stack offsets indicate more similarity

[**ebp - 4**] = {13} * [**ebp - 128**] = {13} ➔ score - z%

universitätbonn CS
Informatik IV – leder@cs.uni-bonn.de

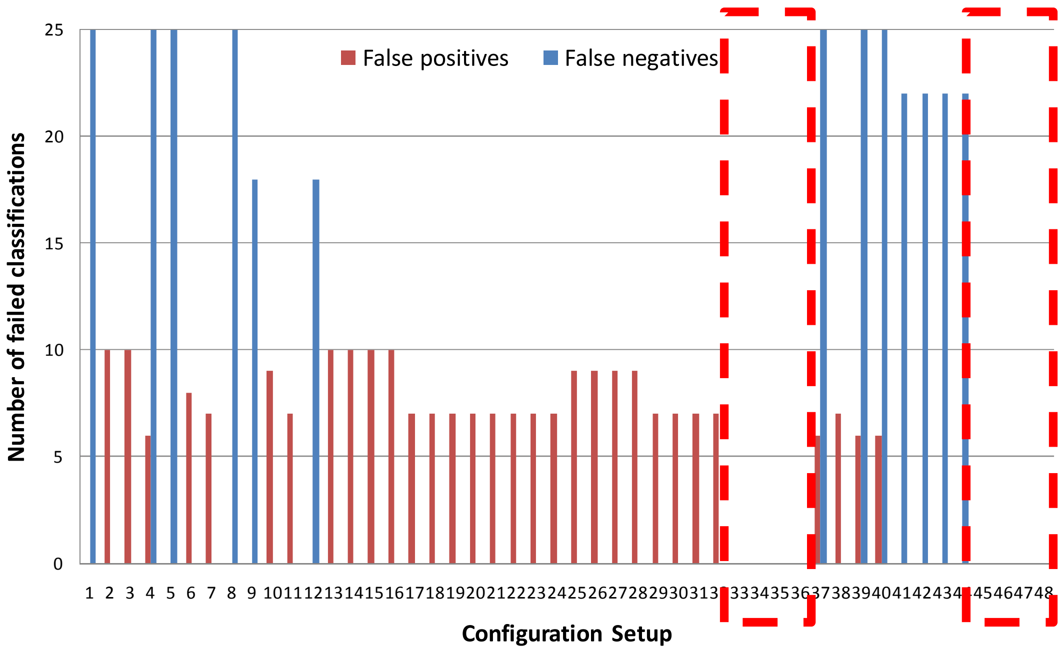
PARAMETER DERIVATION

- Lexotan32 (sophisticated metamorphic engine)
- 25 variants, 25 benign programs
- Sensitivity Analysis → best combination



Informatik IV – leder@cs.uni-bonn.de

24 setups with perfect separation (out of 192)

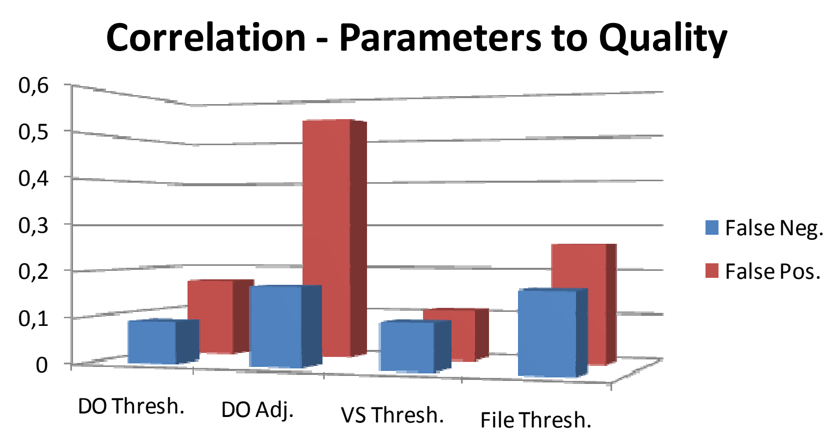


Question: Luck, small test set, or parameter influence?

universitätbonn CS 0100
Informatik IV – leder@cs.uni-bonn.de

Which parameters have strongest impact?

(or need to be set specifically)



Informatik IV – leder@cs.uni-bonn.de

universitätbonn CS

- DO Adjustments (Penalty) of 30%
- Threshold matching
- High threshold for Files and Data Objects
- Threshold for Value Sets almost irrelevant

Are those parameters family dependent?

universitätbonn CS 0100
Informatik IV – leder@cs.uni-bonn.de

Evaluation for Parameter Generality

MALWARE DETECTION USING VSA

Informatik IV – leder@cs.uni-bonn.de

- 7 different metamorphic malware: W32… /
  - Lexotan32
  - Evol

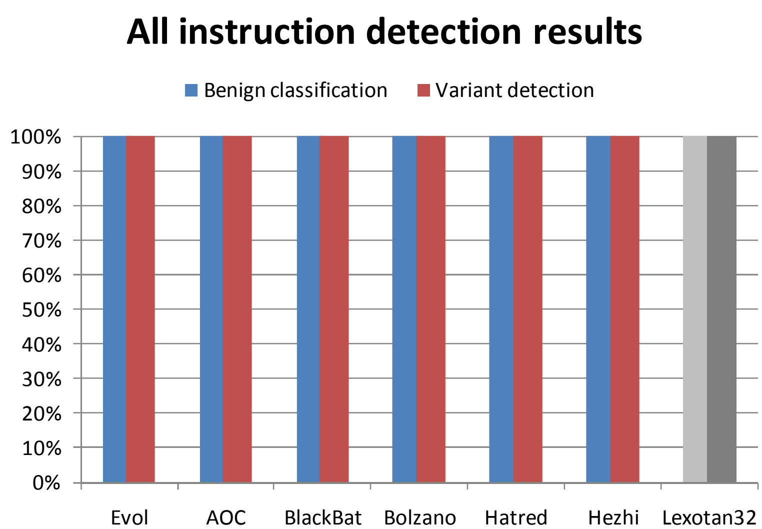
  - AOC
  - Blackbat
  - Bolzano
  - Hatred
  - Hezhi

- **Each** test set: 55 files:
  - 5 infected for refinement
  - 25 infected
  - 25 clean

universitätbonn CS 0100

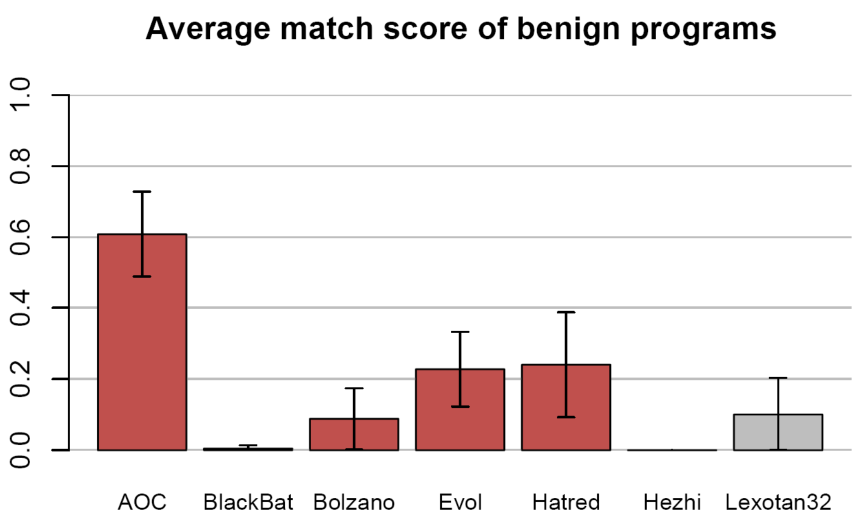
Informatik IV – leder@cs.uni-bonn.de

100 % Separation = 0 FALSE NEGATIVES , 0 FALSE POSITIVES



universitätbonn

- False Positive distance
- How characteristic are the Value Sets?
- False positive estimation for real-world



Parameter set is suitable for other families, too!

Informatik IV – leder@cs.uni-bonn.de

Decreasing complexity → Only specific Points Of Interest in prog.

Malware	All instruction POIs	Jump POIs	Call POIs	Function POIs
W32/AOC	☑	1 f.p. / 0 f.n.	no value sets	no value sets
W32/BlackBat	☑	☑	☑	☑
W32/Bolzano	☑	☑	no value sets	no value sets
W32/Evol	☑	☑	☑	1 f.p. / 0 fn
W32/Hatred	☑	☑	no value sets	no value sets
W32/Hezhi	☑	no value sets	no value sets	no value sets

☑- 100% detection, 0 false positives

Summary:
- 2 False positives in 120 samples
- No false negatives
- Refinement to strict for special POI types

universitätbonn CS

Informatik IV – leder@cs.uni-bonn.de

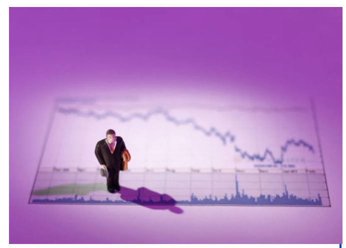
Larger sample sets

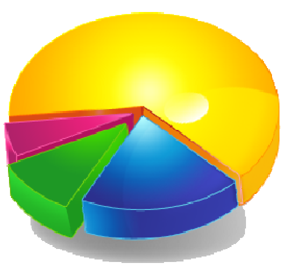
CLASSIFICATION OF METAMORPHIC FAMILIES

Informatik IV – leder@cs.uni-bonn.de

Setup

- 4197 samples from MWCollect database
- 7 metamorphic families
- Same parameter set as before
- (All Instructions)

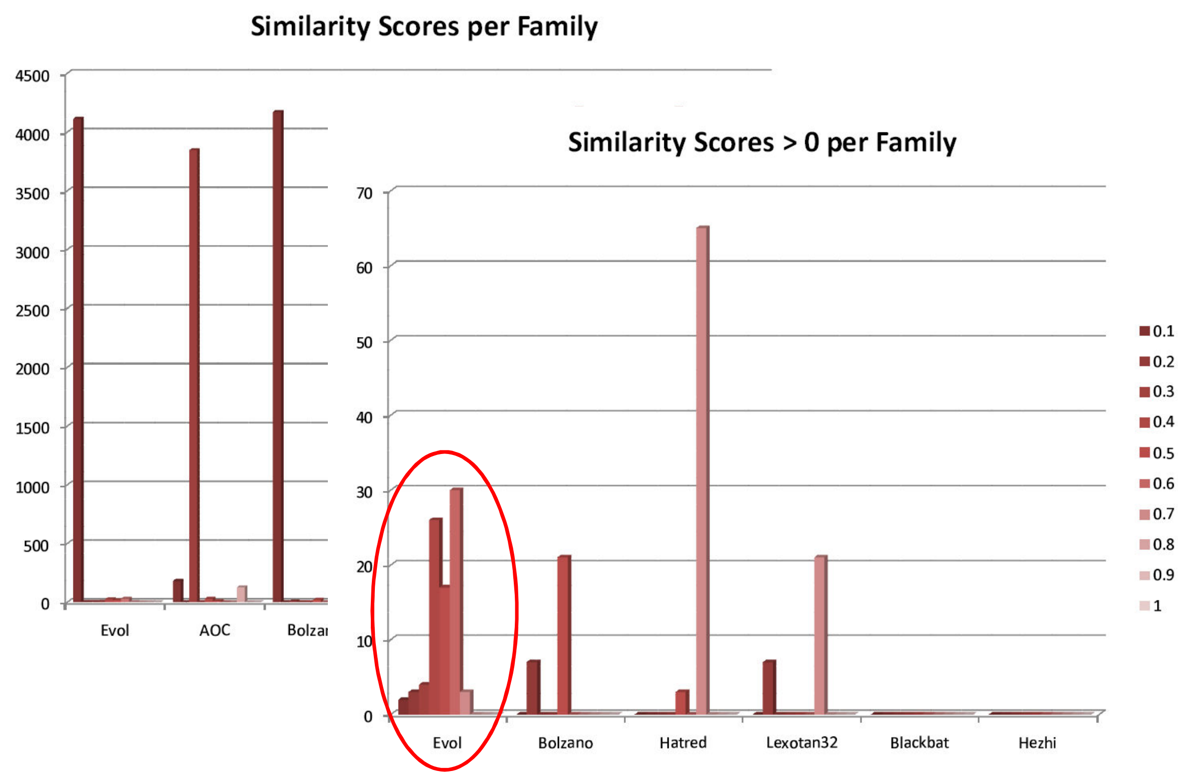




Classification goal – perfect separation

- Variant classification as family members
- All other samples as non-members

Informatik IV – leder@cs.uni-bonn.de

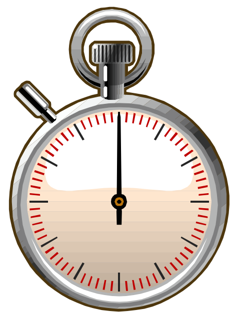


Informatik IV – leder@cs.uni-bonn.de

universitätbonn CS

Competition

RUNTIME PERFORMANCE



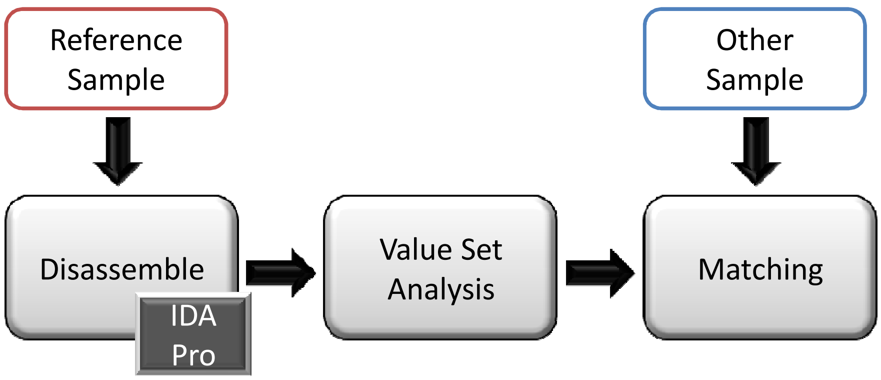
Informatik IV – leder@cs.uni-bonn.de

Best classification is unusable if too slow for use case

Existing approaches and use-cases

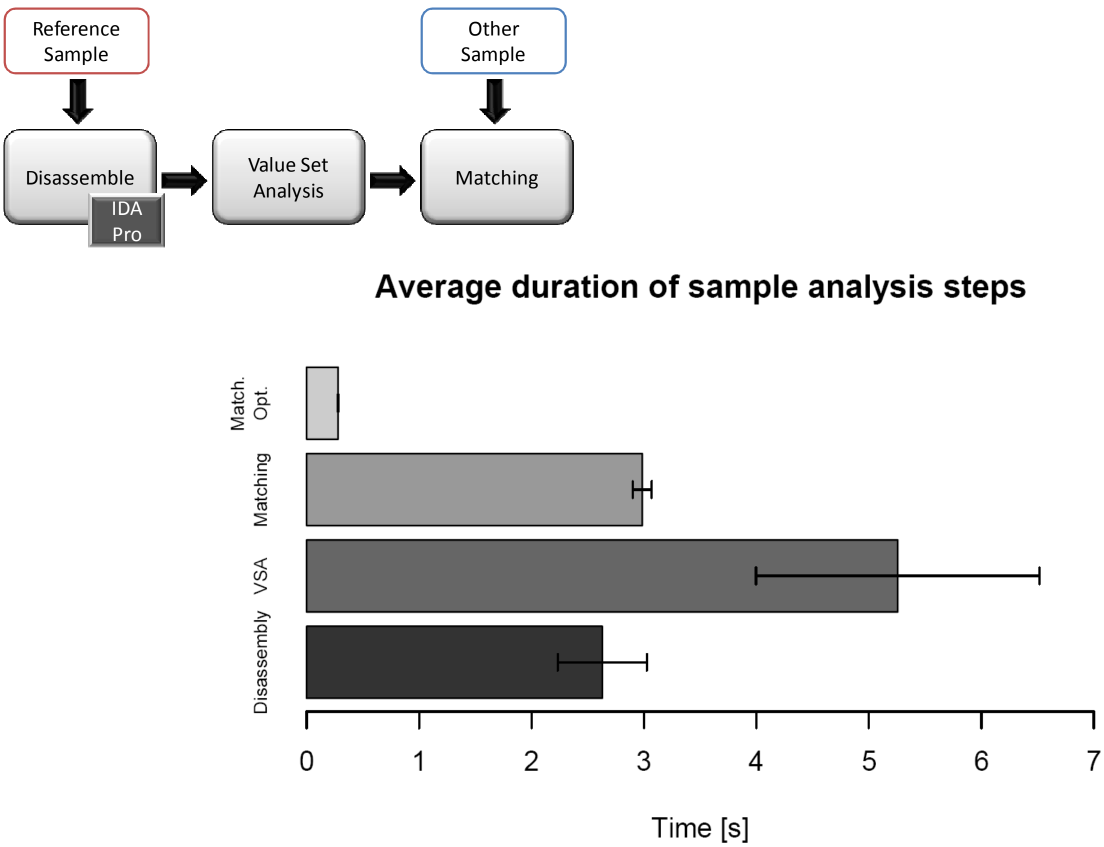
- Classification (mostly blackbox):
  - Run time: 2 to 10 minutes + classification time
- Virus-Detection (on-demand)
  - Application slow-down: 100% - 200% overhead for most AV
  - Data throughput: 3.6 – 18 MB/s
- Mail gateways
  - Greylisting introduces delays of 5 - 15 minutes

Informatik IV – leder@cs.uni-bonn.de



Run-Time measurements give upper bound…

- IDA Pro – unnecessary analysis steps
- Value Set Analysis – Python (IDAPython Integration)
- Matching – Python
- C up to 280 times faster than Python [Armin Rigo - Psycho]



Informatik IV – leder@cs.uni-bonn.de

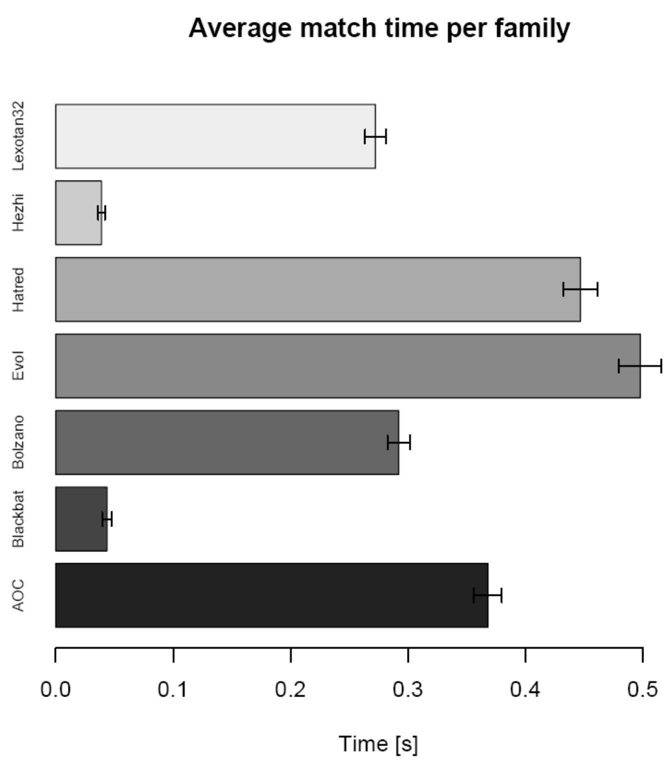
- **Average total analysis time / sample: 7.9 s**
- **Average match time +0.28 s**

- **Data throughput: 20 KB/s**

Current implementation…

+ Faster than Blackbox
+ Ok for mail gateways
- Too slow for on-demand scanning
  (may be an additional means to AV)

Informatik IV – leder@cs.uni-bonn.de


Average match time per family
Lexotan32
Hezhi
Hatred
Evol
Bolzano
Blackbat
AOC
0.0
0.1
0.2
0.3
0.4
0.5
Time [s]

- Structure of metamorphic malware changes but (general) behavior stays similar
- Static Analysis can estimate behavior based on data flow relations and values → **Value Set Analysis**

- **Strict parameters allow for good differentiation**
- Detection possible
- Classification/**differentiation** from other families **perfect**

- Run-time ok for classification, mail gateways, …
- Too slow for on-demand scanning

Informatik IV – leder@cs.uni-bonn.de

DIMVA 2010
July 8-9 in Bonn.

SIG SIDAR Conference on
Detection of **I**ntrusions and
Malware & **V**ulnerability **A**ssessment

http://www.dimva.org

Malware Boot Camp
- Summer- and winter-school
- February and September
- 5 weeks hardcore fun

leder@cs.uni-bonn.de

BACKUP

Informatik IV – leder@cs.uni-bonn.de

a = {1, 5, 10}	b = {0, 1, 2}

Value Set

Data Objects

Number of values in real malware

W32/Lexotan32

Original (primary) file:	188 VS - 933 Data Objects
1st refinement:	108 VS - 238 Data Objects
2nd refinement:	108 VS - 225 Data Objects.
3rd – 9th refinement:	no changes

Informatik IV – leder@cs.uni-bonn.de

universitätbonn CS

Points of Interest (POI)

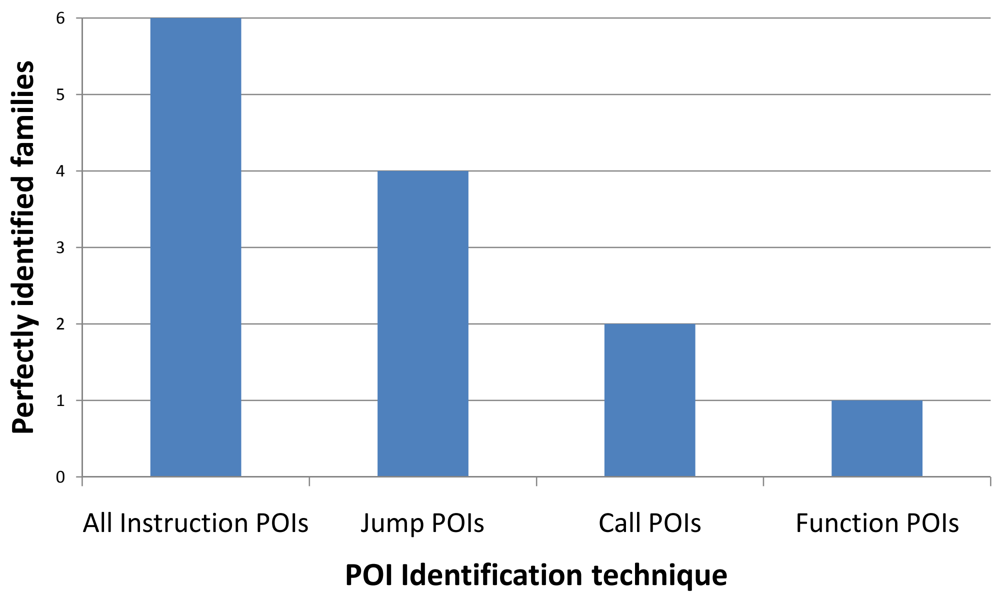
- Points in Executable that are likely to contain characteristic Value Sets

- **All instruction POIs**: Every instruction may be interesting
- **Jump POIs:** Decision dependent values
- **Call POIs:** Function parameters and caller state
- **Function POIs:** State at the beginning of an (internal) function

Informatik IV – leder@cs.uni-bonn.de

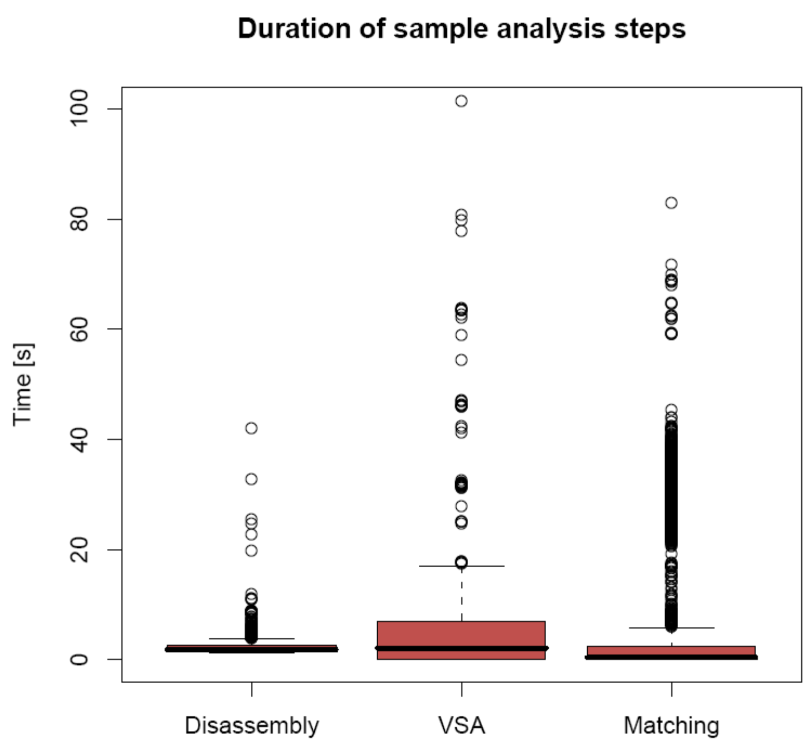
Aiming for **PERFECTION**…

(100% detection, 0 false positives)



Informatik IV – leder@cs.uni-bonn.de

- Outliers
  (up to 2000 s)
- Real-world:
  Impl. time limit



Informatik IV – leder@cs.uni-bonn.de

universitätbonn CS