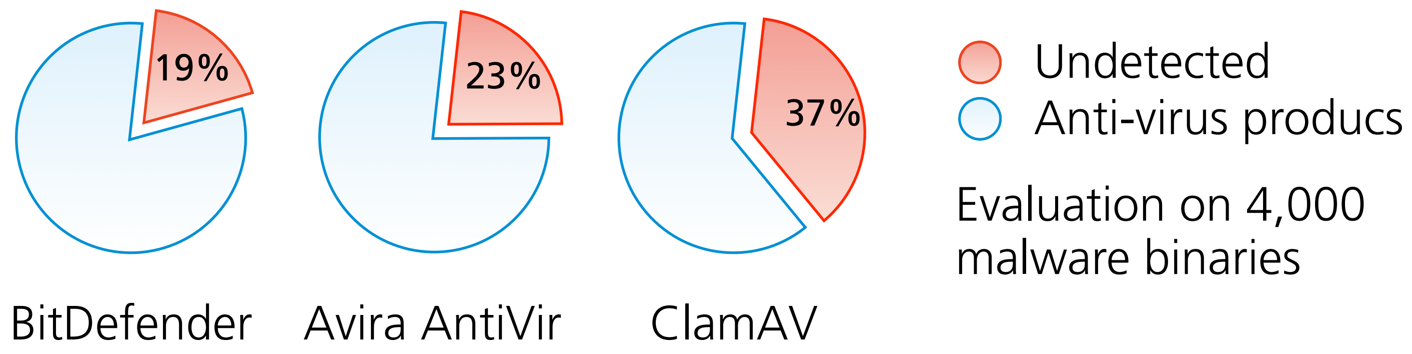
# *Learning and Classification of Malware Behavior*

Konrad Rieck[1], Thorsten Holz[2], Carsten Willems[2],
Patrick Düssel[1], and Pavel Laskov[1]

DIMVA 2008, Paris, France

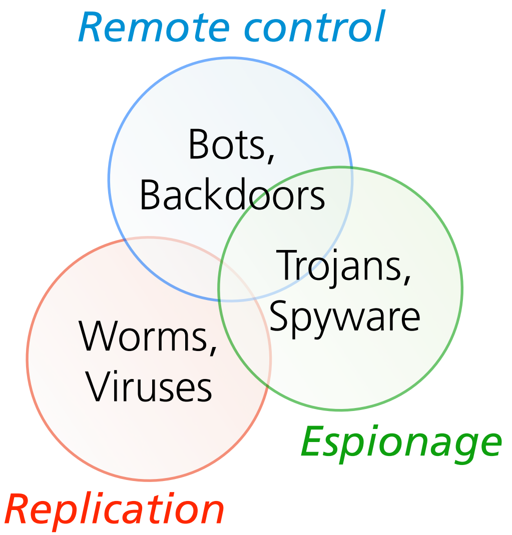
[1] Fraunhofer Institute FIRST, Germany
[2] University of Mannheim, Germany

- **Malicious software: A vivid threat**
  - Plethora of worms, trojans, bots, backdoors
  - Exponential growth of malware in the wild
  - Emergence of criminal “industries”
- **Conventional static defenses insufficient**
  - High degree of polymorphy and obfuscation



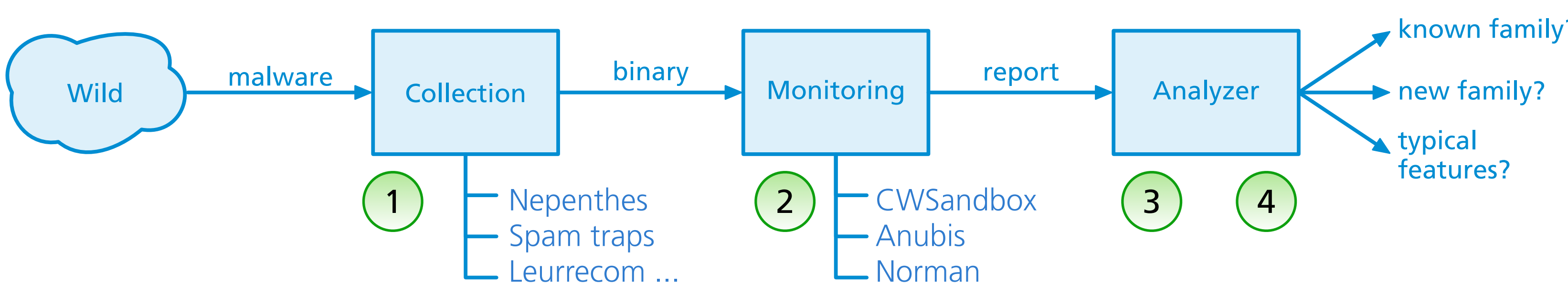

Evaluation on 4,000 malware binaries

- **Malware behavior**
  - Malware differs in purpose and functionality
  - Typical and discriminative behavioral patterns



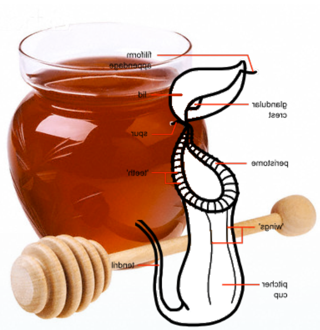

- **Behavior-based analysis**
  - Monitoring and detection of malicious behavior
  - AV products: manually generated behavior rules
  - Alternative, fully automated approaches?

- Clustering of malware behavior *(e.g. Bailey et al., RAID 2007)*
  - Difficult to control cluster models (many vs. few)
  - Clustering often non-predictive, e.g. linkage clustering
- Idea: *Generalize from behavior and prior knowledge*
  - Incorporate (noisy) labels, e.g. by anti-virus tool
  - Learn classification of malware families using labels



- Automatic collection of current malware families
  - Broad range of malware using diverse methods, e.g. honeypots, spam traps, honeyclients

*Vulnerability emulation*



Nepenthes


Spam traps

*Client-side emulation*

Honeyclients

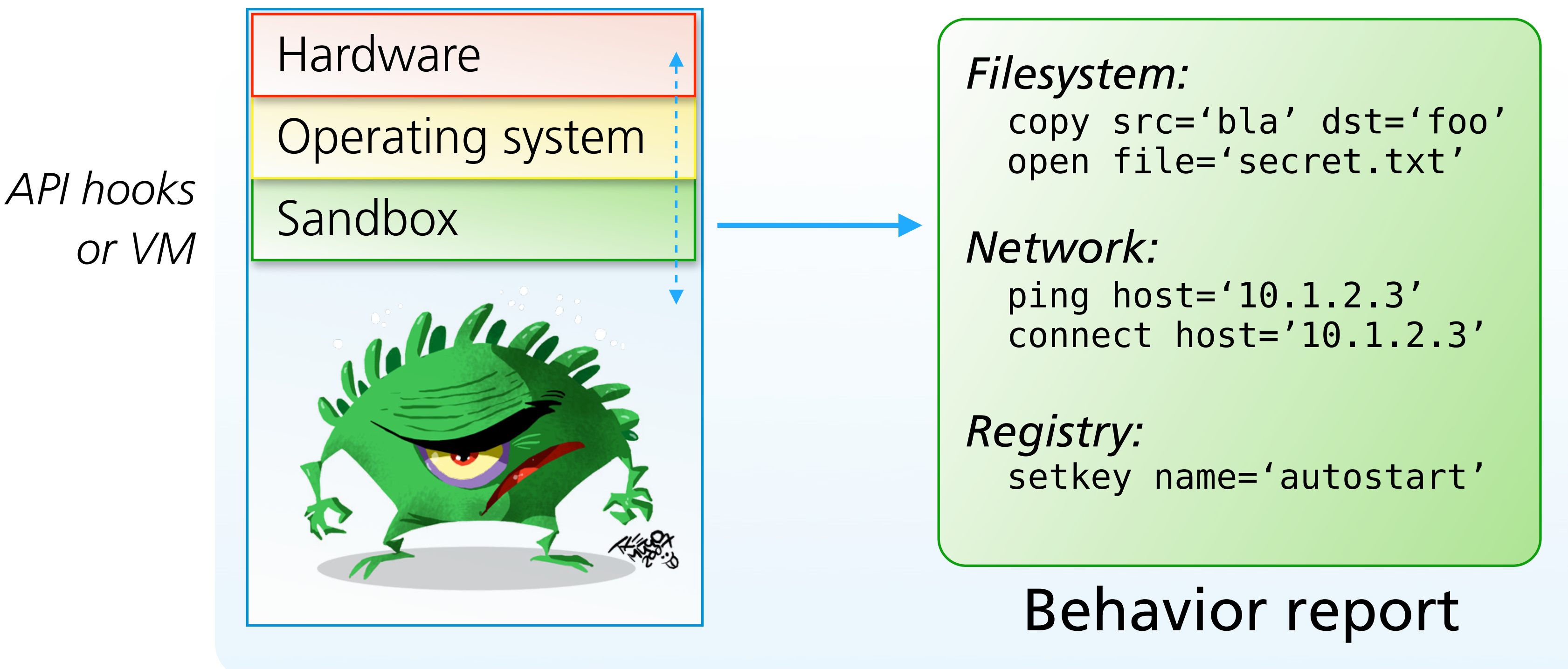
Self-replicating malware

Trojans and backdoors

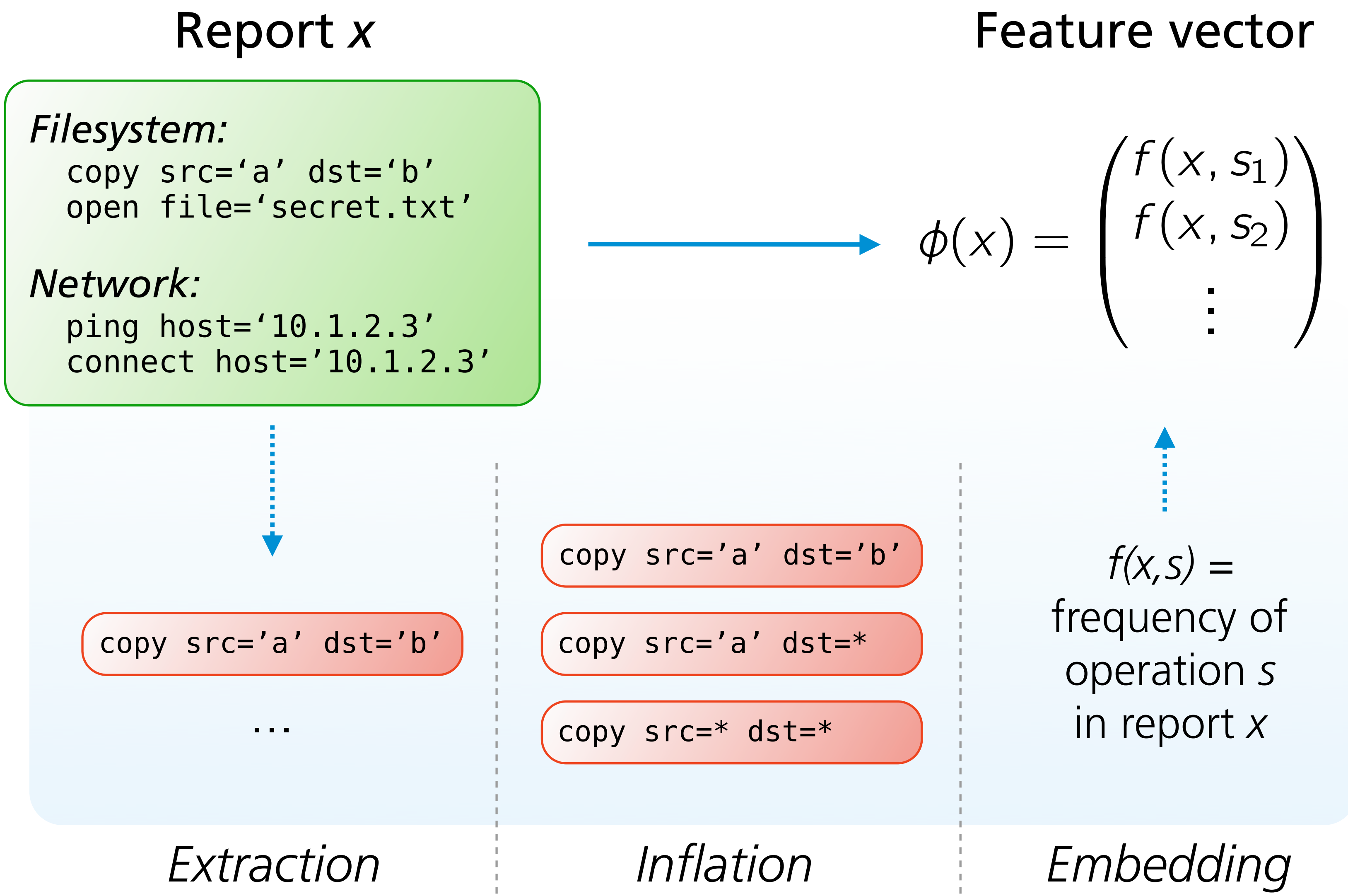
Drive-by malware

*(e.g. Bächer et al., RAID 2006)*

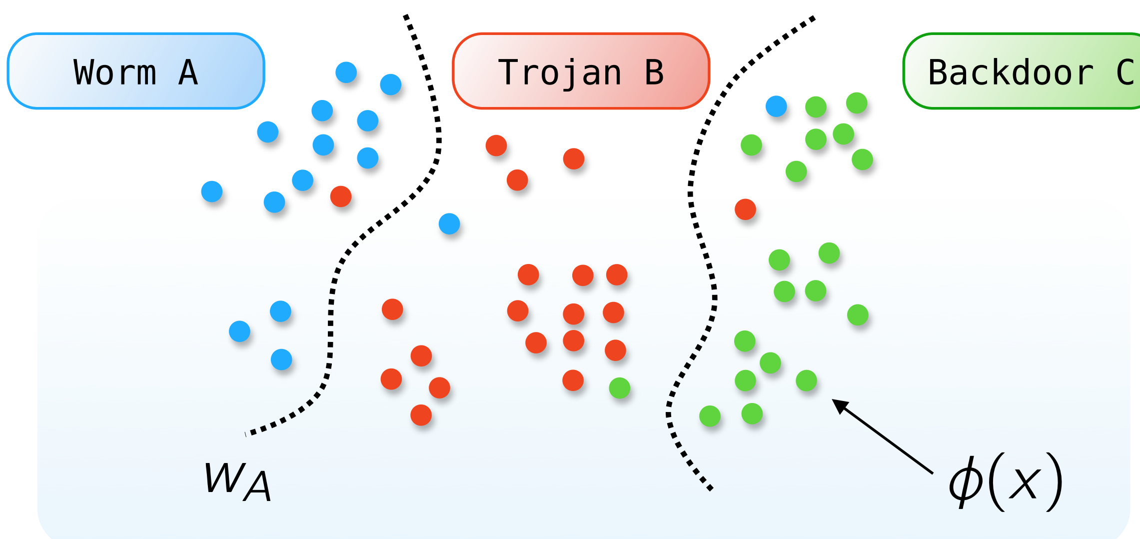
- **Sandbox for malware**
  - Protected execution environment (e.g. CWSandbox)
  - Monitors and reports observed behavior




(Willems et al., IEEE S&P Mag. 2007)


Report x
Feature vector
Filesystem:
copy src='a' dst='b'
open file='secret.txt'
Network:
ping host='10.1.2.3'
connect host='10.1.2.3'
$\phi(x) = \begin{pmatrix} f(x, s_1) \\ f(x, s_2) \\ \vdots \end{pmatrix}$
copy src='a' dst='b'
...
copy src='a' dst='b'
copy src='a' dst=*
copy src=* dst=*
f(x,s) = frequency of operation s in report x
Extraction
Inflation
Embedding

- Discrimination of malware families in feature space
  - Assign family label to embedded reports, e.g. AV label



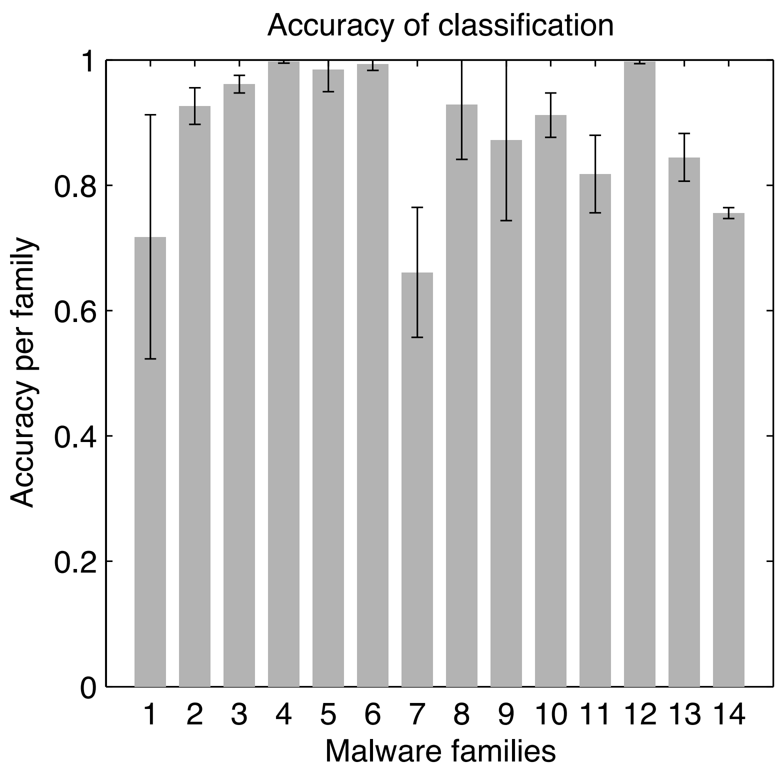

  - Learn maximum-margin hyperplane *w* for each family
  - Incorporation of non-linearity using kernel functions

*(see Burges, KDDM 1998)*

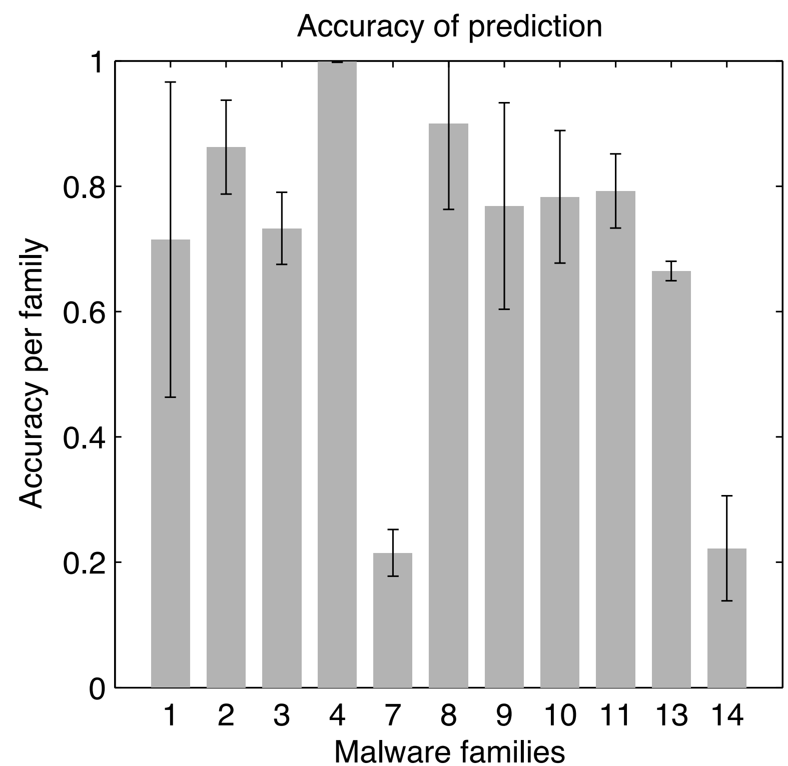
- Malware collection labeled using AV tool (AntiVir)
  - #*k*: 10,072 malware binaries from 14 families
  - #*u*: 3,139 unknown variants (detected 4 weeks later)

| Malware family | #k | #u | Malware family | #k | #u |
|---|---|---|---|---|---|
| 1: Backdoor.VanBot | 91 | 169 | 8: Worm.Korgo | 244 | 4 |
| 2: Trojan.Bancos | 279 | 208 | 9: Worm.Parite | 1215 | 19 |
| 3: Trojan.Banker | 834 | 185 | 10: Worm.PoeBot | 140 | 188 |
| 4: Worm.Allaple | 1500 | 614 | 11: Worm.Rbot | 1399 | 904 |
| 5: Worm.Doomber | 426 | 0 | 12: Worm.Sality | 661 | 0 |
| 6: Worm.Gobot | 777 | 0 | 13: Worm.SdBot | 777 | 597 |
| 7: Worm.IRCBot | 229 | 107 | 14: Worm.Virut | 1500 | 144 |

- Learning on known, prediction on unknown variants




Known variants, avg. 88%




Unknown variants, avg. 69%

- **High detection accuracy** (Note: random guessing = 7%)

- Explanation of learned malware behavior classifier
  - Most discriminative dimensions in hyperplane vectors

*Worm.Sality*

```
0.0142: create_file ... srcpath="C:\windows\system32\" src=*
0.0073: create_file ... srcpath="C:\windows\system32\" src="vcmgcd32.dl_"
0.0068: delete_file ... srcpath="C:\windows\system32\" src=*
0.0051: create_mutex name="kuku_joker_v3.09"
0.0035: enum_processes apifunction="Process32First”
```

*Worm.Doomber*

```
0.0084: create_mutex name="GhostBOT0.58c"
0.0073: create_mutex name="GhostBOT0.58b"
0.0052: create_mutex name="GhostBOT0.58a"
0.0014: enum_processes apifunction="Process32First"
0.0011: query_value key="HKEY_LOCAL_MACHINE\...\run" value="GUARD"
```

# Conclusions

- **Behavior-based malware analysis**
  - Extension of current AV tools *(see Oberheide et al., USENIX 2008)*
  - Hinders simple obfuscation and polymorphy
- **Supervised learning on malware behavior**
  - Detection accuracy: *69% unknown malware variants*
  - No black box: *Explanation via hyperplane vectors*
  - Further extension: *Rejection of unknown behavior*
- **Perspectives**
  - Semi-supervised learning: Best of both worlds.

**Thanks.** ***Questions?***

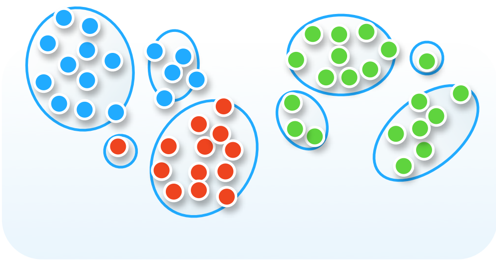
## Evasion?

- Evasion attacks
  - Detection of honeypot or sandbox environment
  - Obfuscated and polymorphic behavior
  - Mimic behavior of benign programs or other malware
- Consequences & defenses
  - Run multiple honeypots and sandboxes in parallel
  - Obfuscation and polymorphy: Discriminative features?
  - Fruitless to mimic benign program = No real activity

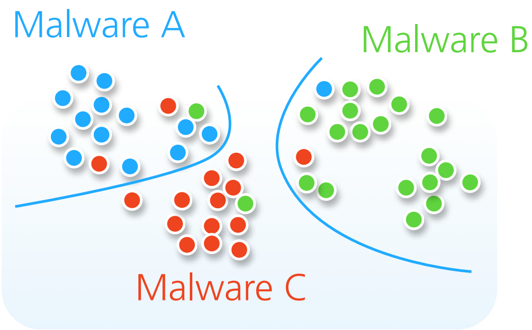
# References


- Bächer, Kötter, Holz, Dornseif, Freiling. *The Nepenthes platform: An efficient approach to collect malware*. RAID 2006.
- Bailey, Oberheide, Andersen, Mao, Jahanian, Nazario. *Automated classification and analysis of Internet malware.* RAID 2007.
- Burges. *A Tutorial on Support Vector Machines for Pattern Recognition.* Knowledge Discovery and Data Mining 2(2), 1998.
- Oberheide, Cooke, Jahanian. *N-Version Antivirus in the Network Cloud.* USENIX 2008.
- Rieck, Laskov. *Linear-Time Computation of Similarity Measure for Sequential Data*. Journal of Machine Learning Research 9(1), 2008.
- Willems, Holz, Freiling. *Towards automated dynamic binary analysis*, IEEE Magazine Security & Privacy 5(2), 2007.

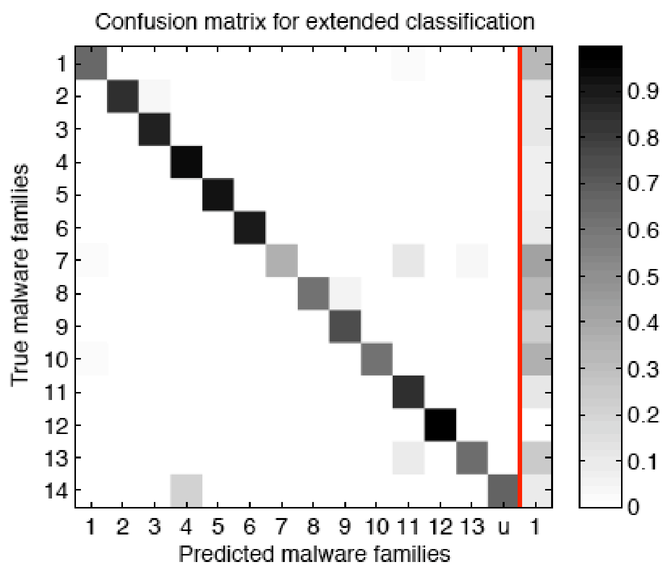
- Clustering (unsupervised)
  - Determine malware families from structure only
  - Difficult to control model complexity without labels



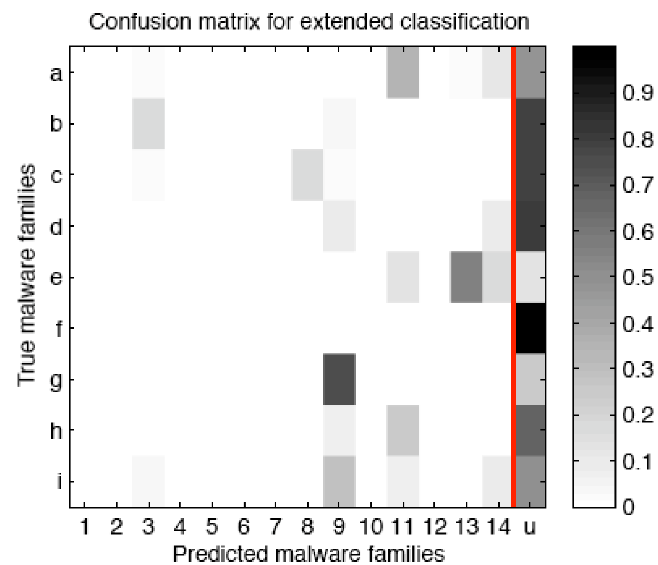
- Classification (supervised)
  - Determine malware families using structure *and* labels
  - Generalization beyond noisy labels



- Rejection of unknown behavior
  - Probabilistic fit on output of classifier (reject if <0.5)



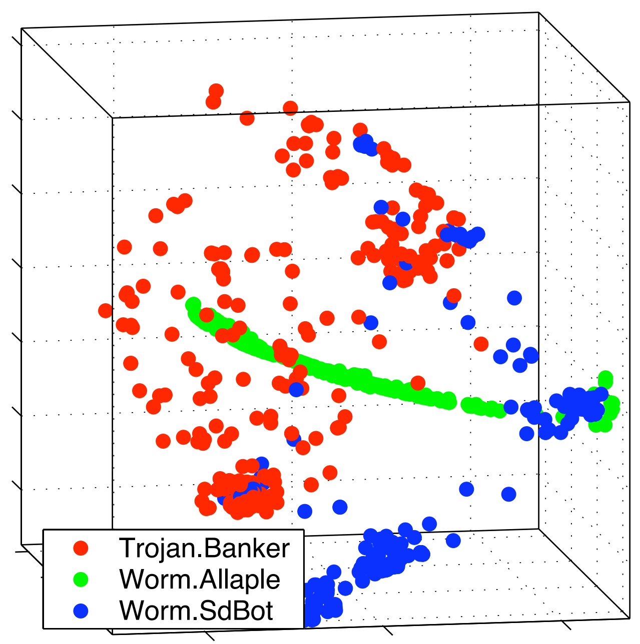

Known families




Unknown families

- Reliable rejection of unknown behavior, yet accuracy decreases from 88% to 73%

- **Embedding to high-dimensional vector space**
  - Each operation spans several dimensions
  - > 1,000,000 and more dimensions




  - Visualization using projections (e.g. with PCA)