Computational Forensics

Forensic Investigations in Cyberspace: what about big data?

Katrin Franke
Norwegian Information Security Laboratory (NISlab),
Department of Computer Science and Media Technology,
Gjøvik University College

http://www.nislab.no

Forensics Lab TESTIMON

Computational Forensics

Crime in the Modern World

- Massive amount of data:
  - 247 billion email per day
  - 234 million websites
  - 5 billion mobile-phone users
- ICT Infrastructures:
  - Complex, rapidly growing
  - Dynamically changing
  - Hostile, adversary environment
- Cybercrime:
  - One million victims daily
  - Expected losses 297 billion Euro
  - Crowd sourcing -> Crime sourcing
  - Flash mobs -> Flash robs

- Proactive, Ultra-large scale Forensic Investigations, Computational Forensics:
  - Situation-aware methods
  - Quantified, measurable indicators
  - Adaptive, self-organizing models
  - Distributed, cooperative, autonomous
- Rule-of-Law:
  - Culture, social behaviours
  - Legal & privacy aspects
  - Cross-jurisdiction cooperation
  - European / International cyberlaw
  - Law as framework for ICT
  - Law as contents of ICT, Automation, programming of legal rules

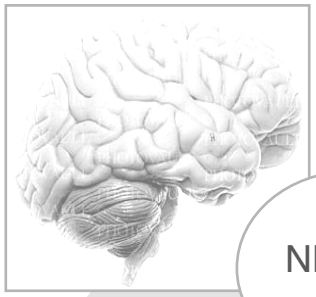
HØGSKOLEN I GJØVIK

Forensics Lab STESTIMON

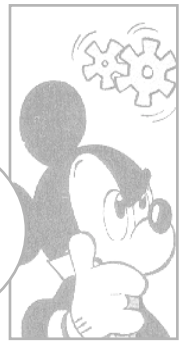
Computational Forensics:

Adding Efficiency and Intelligence to BIG DATA Investigation

Requirement of Adapted Computer Models & Operators



Brain



Reasoning

NN

FL

EC

Natural Evolution

Imprecision, Uncertainty, Partial Truth

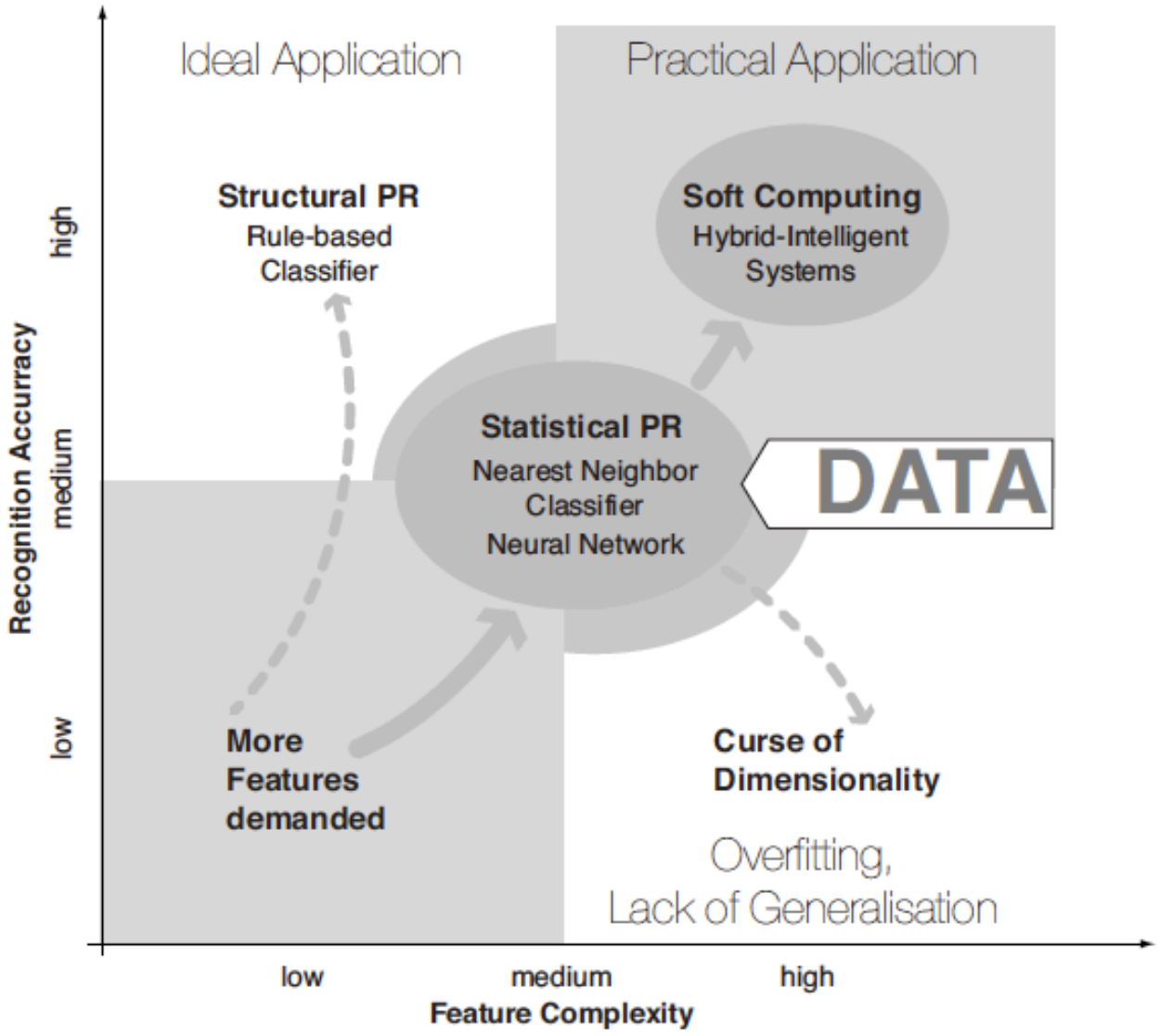
Computational Intelligence

NN: Neuronal Networks
FL: Fuzzy Logic
EC: Evolutionary Computation

Methods of Computational / Machine Intelligence

- **Signal / Image Processing** : one-dimensional signals and two-dimensional images are transformed for the purpose of better human or machine processing,
- **Computer Vision** : images are automatically recognized to identify objects,
- **Computer Graphics / Data Visualization** : two-dimensional images or three-dimensional scenes are synthesized from multi-dimensional data for better human understanding,
- **Statistical Pattern Recognition** : abstract measurements are classified as belonging to one or more classes, e.g., whether a sample belongs to a known class and with what probability,
- **Data Mining** : large volumes of data are processed to discover nuggets of information, e.g., presence of associations, number of clusters, outliers in a cluster,
- **Robotics** : human movements are replicated by a machine, and
- **Machine Learning** : a mathematical model is learnt from examples.

Data-driven Approaches



Big Data Analysis

Inter-relation of feature complexity and expected recognition accuracy.
(Franke 2005)

Forensics Lab STESTIMON

Reverse Engineering Malware

Lars Arne Sand, Katrin Franke,
Jarle Kittilsen, Peter Ekstrand Berg, Hai Thanh Nguyen
Norwegian Information Security Laboratory (NISlab)
Gjøvik University College
www.nislab.no

Reverse Engineering Malware

- **Static analysis**
- **System artifacts**
- **Dynamic analysis**
- Debugging
- Analyzing malicious content
  - **PDFs**
  - **JavaScripts**
  - **Office documents**
  - Shellcode
  - **Network traffic**

Static Analysis

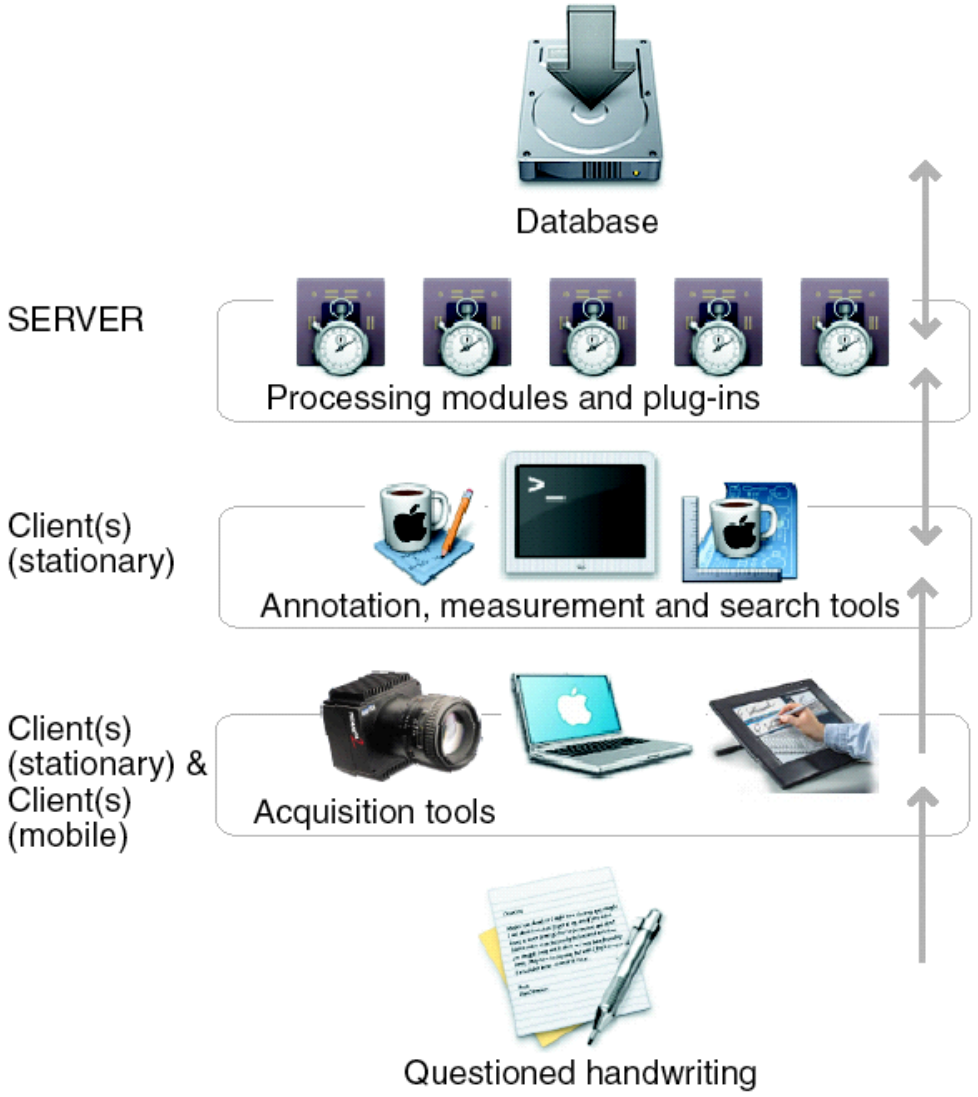
- Static analysis
  - Does not execute malware
  - Analyze:
    - System artifacts
    - Debugging
    - Source code *(not included)*
    - Disassembled code *(not included)*

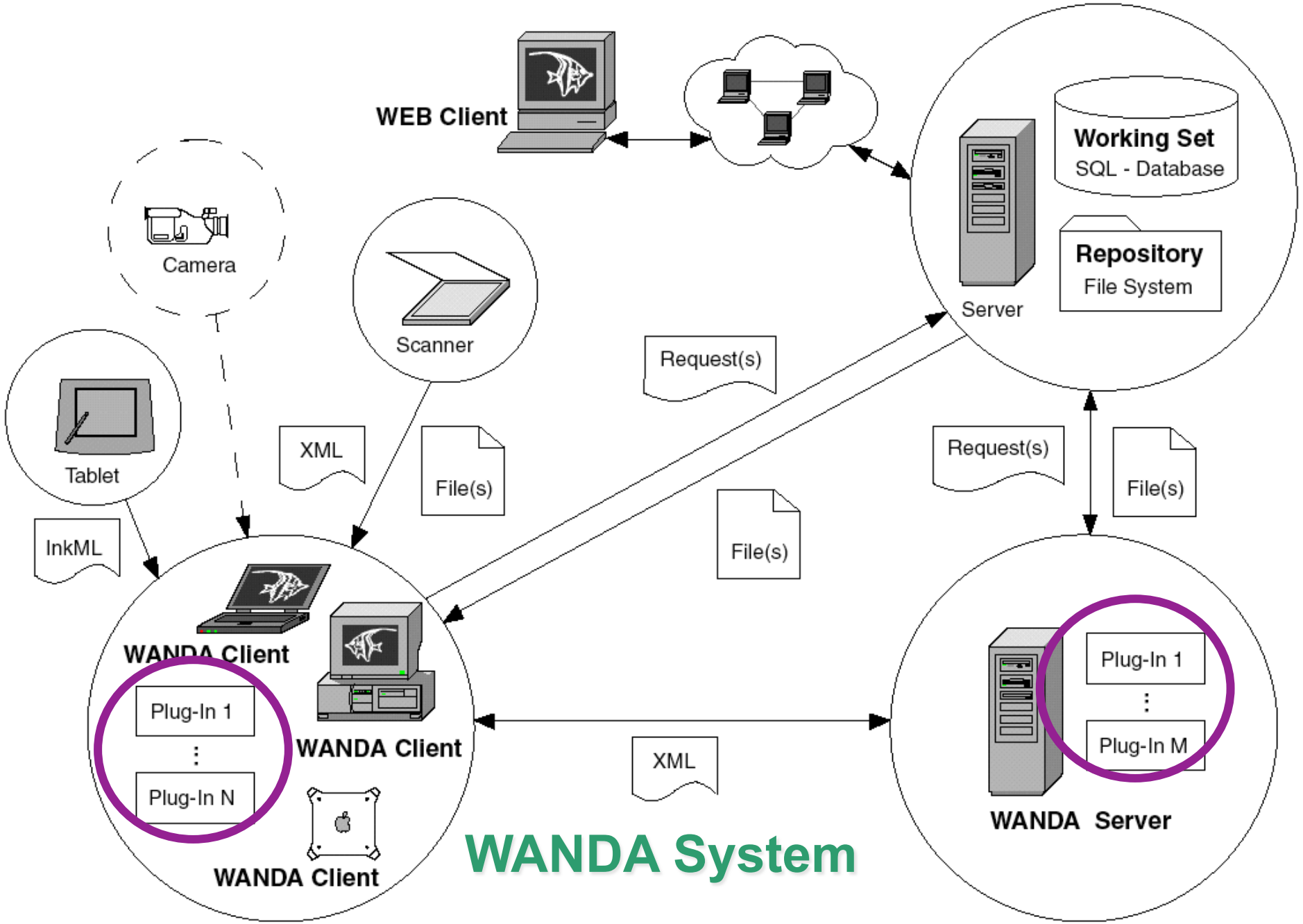
Dynamic Analysis

- Definition
  - *Dynamic analysis is the process of executing malware in a monitored environment to observe its behaviors*
- Deals with finding and understanding the changes made to the system
- Pro:
  - Provide quick information about created and changed files, registry keys, processes, handles, contacted websites, etc.
- Con:
  - Excessive and overwhelming results
  - Need to know the normal behavior of a system

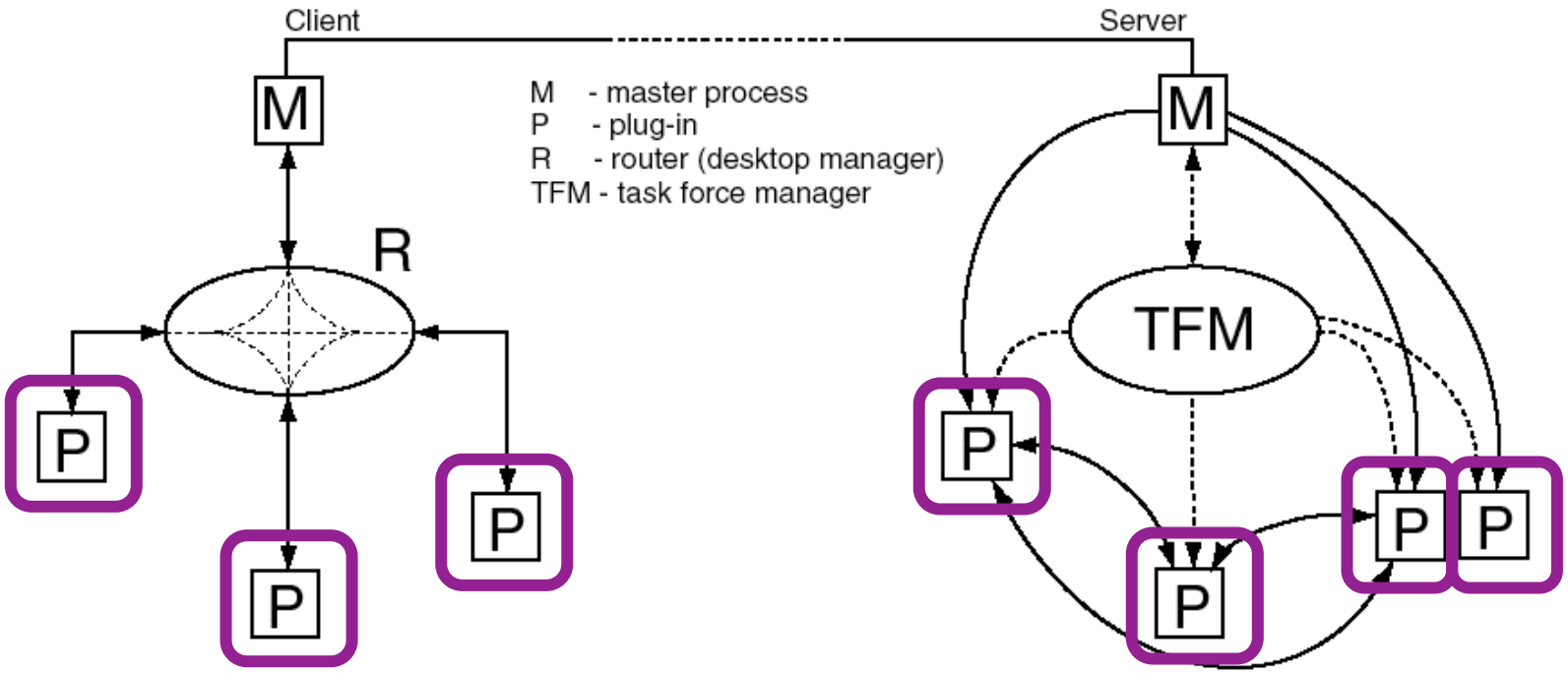
Framework concept

- User interacts via Java client
- Client is the front-end for accessing & processing information
- Information is distributed over and hosted by trusted servers
- Via their clients, users request services provided by the servers




WEB Client
Working Set
SQL - Database
Repository
File System
Server
Camera
Scanner
Tablet
InkML
XML
File(s)
Request(s)
File(s)
Request(s)
File(s)
WANDA Client
Plug-In 1
Plug-In N
WANDA Client
WANDA Client
XML
Plug-In 1
Plug-In M
WANDA Server
WANDA System

Plug-In Concept



a) Client plug-in organization - Master-slave point to point via Router

b) Server plug-in organization - Partial / Full connectivity

Reverse Engineering Malware

- **Static analysis**
- **System artifacts**
- **Dynamic analysis**
- Debugging
- Analyzing malicious content
  - **PDFs**
  - **JavaScripts**
  - **Office documents**
  - Shellcode
  - **Network traffic**

Behavioral Malware Detection

(static, dynamic, combined)

Lars Arne Sand, Katrin Franke

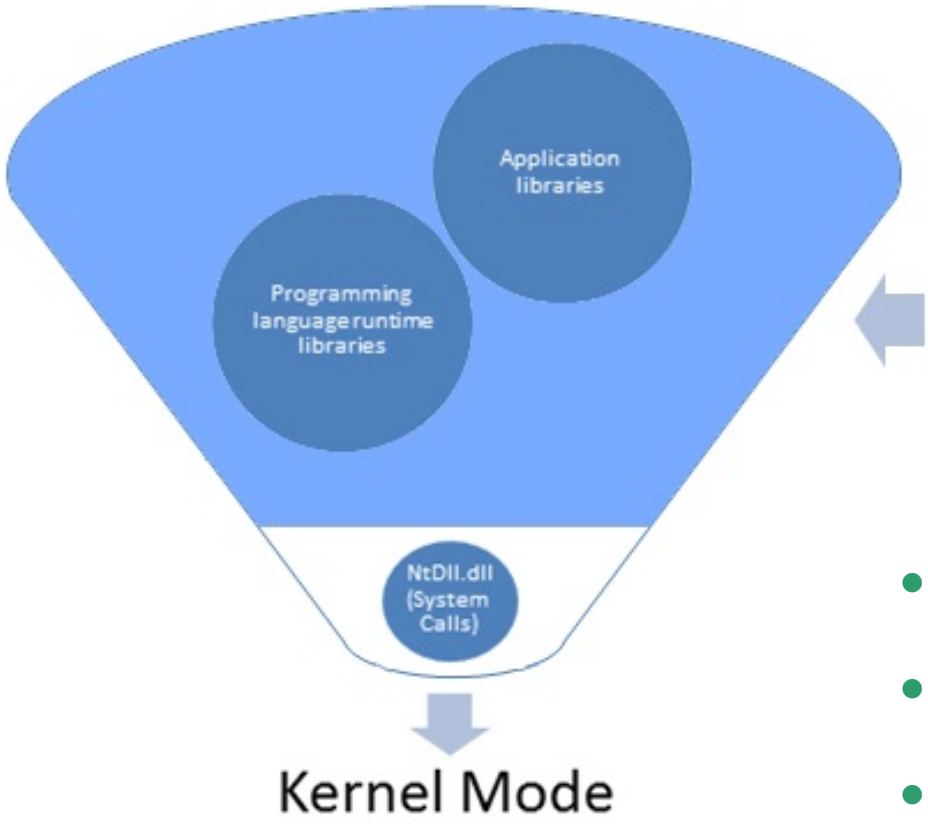
Norwegian Information Security Laboratory (NISlab)

Gjøvik University College

www.nislab.no

Forensics Lab STESTIMON

Layers of Detection

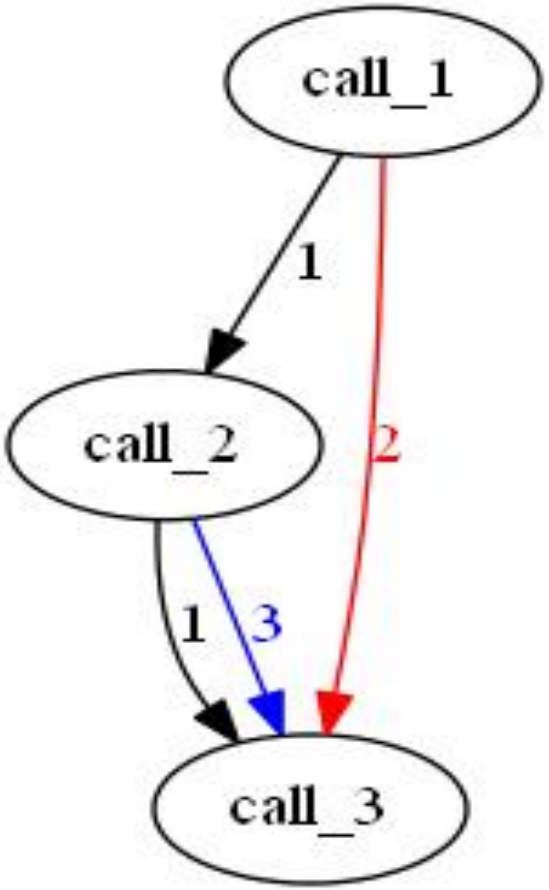


- User mode (library calls)
- Kernel mode (system calls)
- Hybrid (function calls)

Computational Forensics

Information-based Dependency Matching

- Ordering dependency (**1**)
  - sequence
- Value dependency (**2**)
  - parameters
- Def-use dependency (**3**)
  - Parameter and return value
- Sample:
  - call_1(parameter1,**ffff0000**)=0
  - call_2(par)=**0x4fff0418**
  - call_3(**0x4fff0418**,**0xffff0000**)=0



Forensics Lab STESTIMON

Example #1

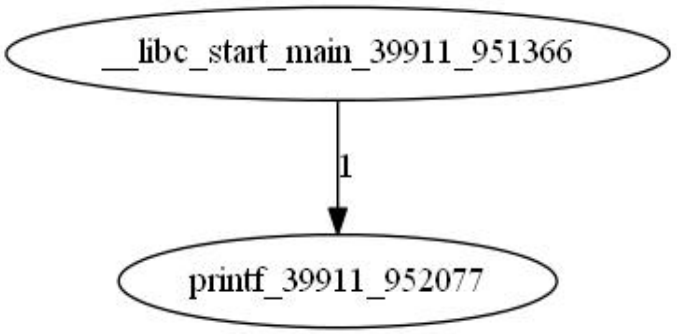
- Library calls (Hello World.c)
  - Code

```c
#include <stdio.h>

int main() {
    printf("Hello world!!!");

    return 0;
}
```

  - Trace

```
11:05:11.951366 __libc_start_main(0x80483c4, 1, 0xbf96afa4,
0x8048400, 0x80483f0 <unfinished ...>
11:05:11.952077 printf("Hello world!!!")          = 14
11:05:11.953227 +++ exited (status 0) +++
```

  - Graph



Example #2

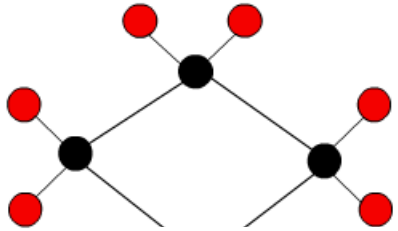
- System calls (Hello world.c)
  - Trace
    - Much more extensive due to memory mapping
    - Example trace
  - Graph
    - Example Graph

Example #3

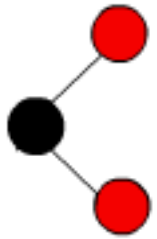
- Actual malware example
  - Malware system call Graph Examples
    - Virus.Linux.Snoopy.a
    - Rootkit.Linux.Matrics.a
    - Exploit.Linux.Small.k

Experimental Design & Data Set #1

- Graph-based Matching
  - http://ailab.wsu.edu/subdue/unsupervised.swf
  - Subdue finds substructures by compressing graphs
  - Supervised Learning is performed by finding substructures that occur frequently in one class but seldom in another



- Dataset
  - Malware
    - Extracted from: vx.netlux.org/index.html (currently down)
    - 190 samples: **7150 vertices, 7790 edges**
  - Benign Software
    - Ubuntu binaries
    - 75 samples: **9025 vertices, 9395 edges**

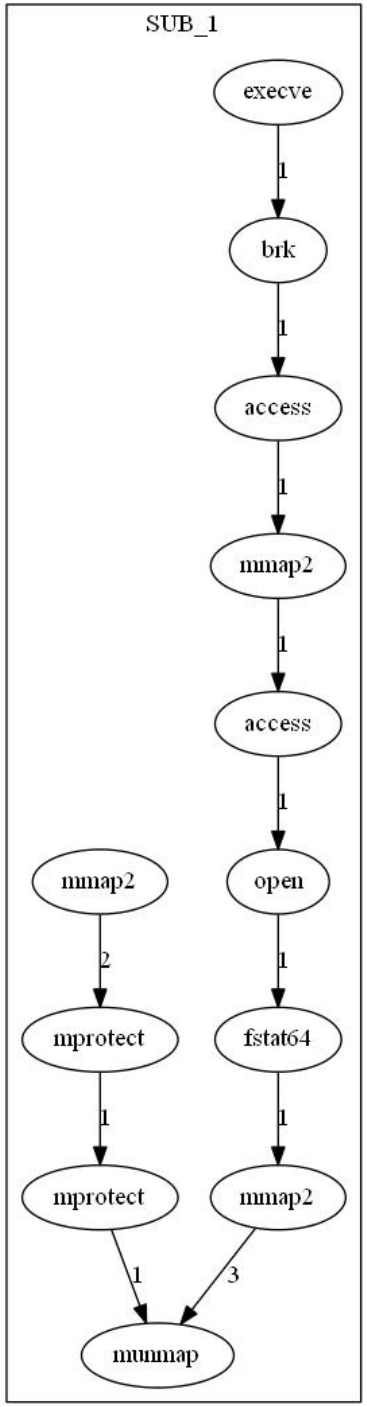


Preliminary Results #1: Graph-based Matching

- Detection rate of 98,9%
- Confusion matrix

System calls		
	Classified as	
Correct class	Malware	Software
Malware	190	0
Software	3	72
	1	0,96

- 190/190 Malware correctly classified
- 72/75 Software correctly classified

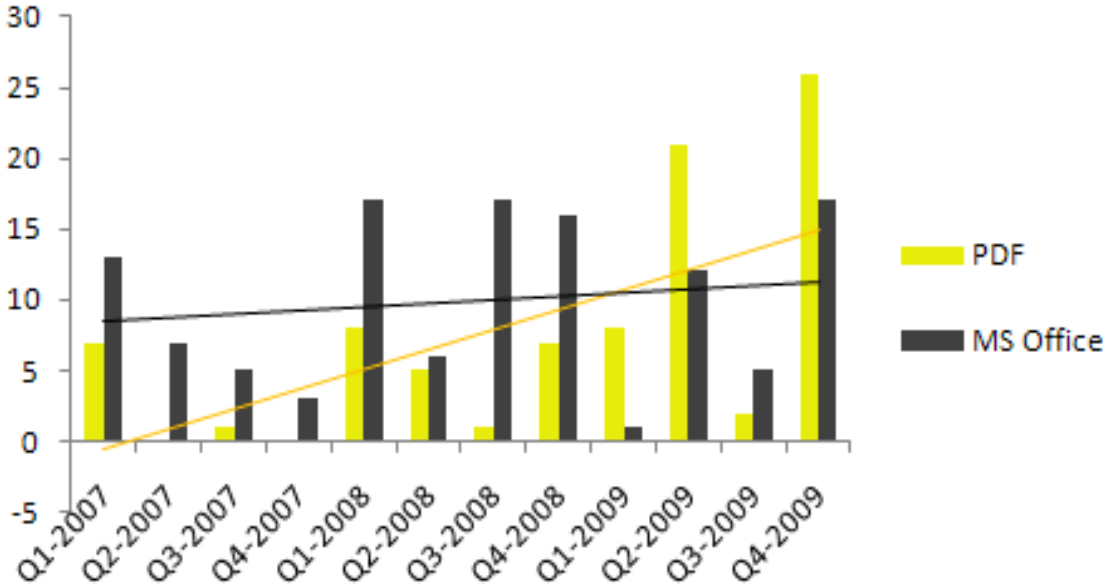


Detecting Malicious PDF

Jarle Kittelsen, Katrin Franke, Hai Thanh Nguyen
Norwegian Information Security Laboratory (NISlab)
Gjøvik University College
www.nislab.no

Analyzing Malicious Content #1

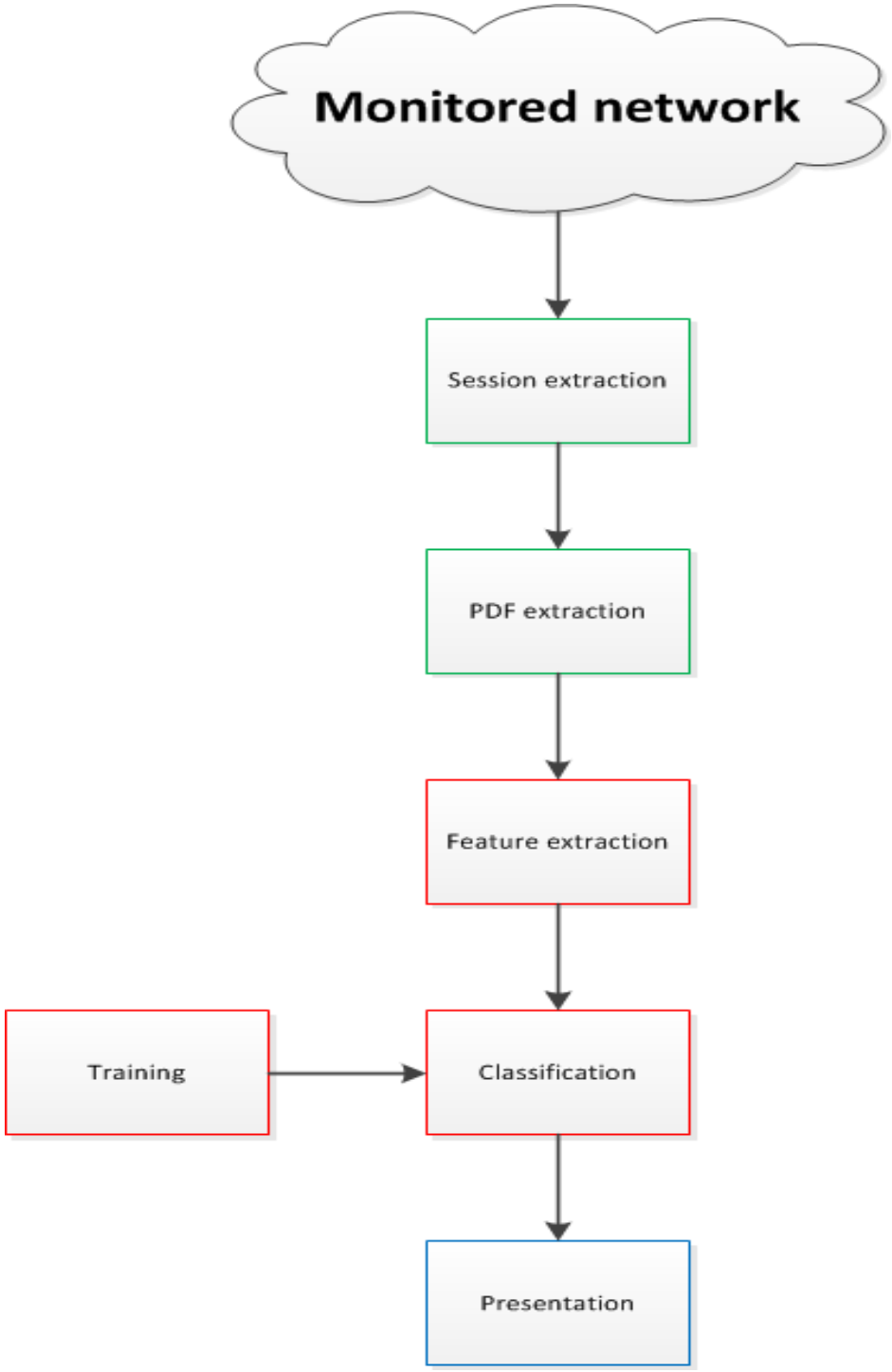
- Frequent analysis:
  - **PDF**
  - **JavaScript**
  - **Office Documents**
  - Flash *(not included)*
  - Shellcode
  - Network Traffic



Research Questions

- Which **features** are significant for detecting malicious PDF documents?
- Which **classifier design** and configuration yields optimal performance in malicious PDF detection?
- How can a real-world IDS, capable of detecting malicious PDFs in network traffic, be implemented?

Method Overview



Data Collection

- PDFs collected within the malware research community and through webcrawling, e.g.,
  - Websense
  - Abuse.ch
  - Sourcefire
- Malicious samples have been submitted globally and detected in various ways, some of the samples are under NDA.
- Data set in total:
  - **7,454 unique benign** PDF samples.
  - **16,296 unique malicious** PDF samples.

Expert-Knowledge Features (KPI)

- Keys from the PDF format (ISO 32000) relevant to malicious PDFs, e.g.,
  - /JavaScript
  - /OpenAction
  - /AcroForm
- Key selection based upon the independed research by (i) Didier Stevens, (ii) Paul Baccas.
- **18 features (keys)** are selected to initialize.
- Additional **feature-set for Javascript**.

Experiments (Exp 1…4)

1. Feature & Classifier Selection
2. Classifier Optimalization and Testing
3. Real-world testing
4. Embedded javascripts

Exp 1: Feature & Classifier Selection

Original feature vector (18):

AA, RichMedia, xref, Encrypt, JBIG2Decode, Launch, JavaScript, OpenAction, Colors,JS, obj_mis, startxref, AsciiHexDecode, ObjStm, AcroForm, stream_mis, Page, trailer

Golub-score feature selection (7):

$$F(x_i) = \left| \frac{\mu_i^+ - \mu_i^-}{\sigma_i^+ + \sigma_i^-} \right|$$

JavaScript, OpenAction, JS, obj_mis, AcroForm, Page, trailer

Generic feature selection GeFS (5)·

JavaScript, JS, startxref, Page, trailer

$$GeFS(x) = \frac{a_o + \sum_{i=1}^{n} A_i(x)\ x_i}{b_o + \sum_{i=1}^{n} B_i(x)\ x_i}$$

Computational Forensics

Exp 1: Feature & Classifier Selection

Tested perfomance using 5 different classifiers:

	BayesNet			C45/J48			RBFNet		
	18	**7**	**5**	**18**	**7**	**5**	**18**	**7**	**5**
Bal succ	0.973	0.94	0.976	0.995	0.995	0.975	0.718	0.797	0.874
Auc	0.996	0.995	0.996	0.997	0.998	0.994	0.879	0.922	0.926
	MLP			**SVM**					
	18	**7**	**5**	**18**	**7**	**5**			
Bal succ	0.96	0.966	0.920	0.995	0.995	0.977			
Auc	0.985	0.987	0.978	0.995	0.996	0.974			

Choose **7 features** from Golub-score selection,
SVM* classifier for further experimentation.

*SVM - Support Vector Machine
*Bal succ - Balanced Successrate *AUC - Area Under (ROC) Curve

Computational Forensics

Discussion and Summary

- The dataset
  - Difficulties controlling factors
  - Best solution: MD5, generalization experiment, big dataset from many sources.
- Changes over time
  - Need for re-learning
  - Online learning
- Detecting malicious PDF documents is feasible
  - using reduced expert feature set, javascript features, SVM
- Aquired knowledge & lessons learned:
  - A PDF dataset (**16.296 / 7,454**) for future reseach.
  - Knowledge on significant features for PDF classification.
  - A method for automated detection of malicious PDF in network traffic.
  - A starting point for future research on malicious javascript detection.

Concluding Remarks

- Computational forensics holds the potential to greatly benefit all of the forensic sciences.
- For the computer scientist it poses a new frontier where new problems and challenges are to be faced.
- The potential benefits to society, meaningful inter-disciplinary research, and challenging problems should attract high quality students and researchers to the field.

Further Reading

- NAS Report: *Strengthening Forensic Science in the United States: A Path Forward* http://www.nap.edu/catalog/12589.html
- van der Steen, M., Blom, M.: *A roadmap for future forensic research*. Technical report, Netherlands Forensic Institute (NFI), The Hague, The Netherlands (2007)
- M. Saks and J. Koehler. *The coming paradigm shift in forensic identification science.* Science, 309:892-895, 2005.
- Starzecpyzel. *United states vs. Starzecpyzel. 880 F. Supp. 1027 (S.D.N.Y),* 1995.
- http://en.wikipedia.org/wiki/Daubert_Standard
- C. Aitken and F. Taroni. *Statistics and the Evaluation of Evidence for Forensic Scientists. Wiley,* 2nd edition, 2005.
- K. Foster and P. Huber. *Judging Science.* MIT Press, 1999.
- Franke, K., Srihari, S.N. (2008). *Computational Forensics: An Overview*, in Computational Forensics - IWCF 2008, LNCS 5158, Srihari, S., Franke, K. (Eds.), Springer Verlag, pp. 1-10.

- Our research center: www.nislab.no
- Our research-lab pages: Testimon Forensics Lab: http://goo.gl/YHMSf
- Our latest publications: http://goo.gl/R58SL

Thank you for your consideration of comments!

Getting in touch

WWW: kyfranke.com

Email: kyfranke@ieee.org

Skype/gTalk: kyfranke