Detecting Environment-Sensitive Malware

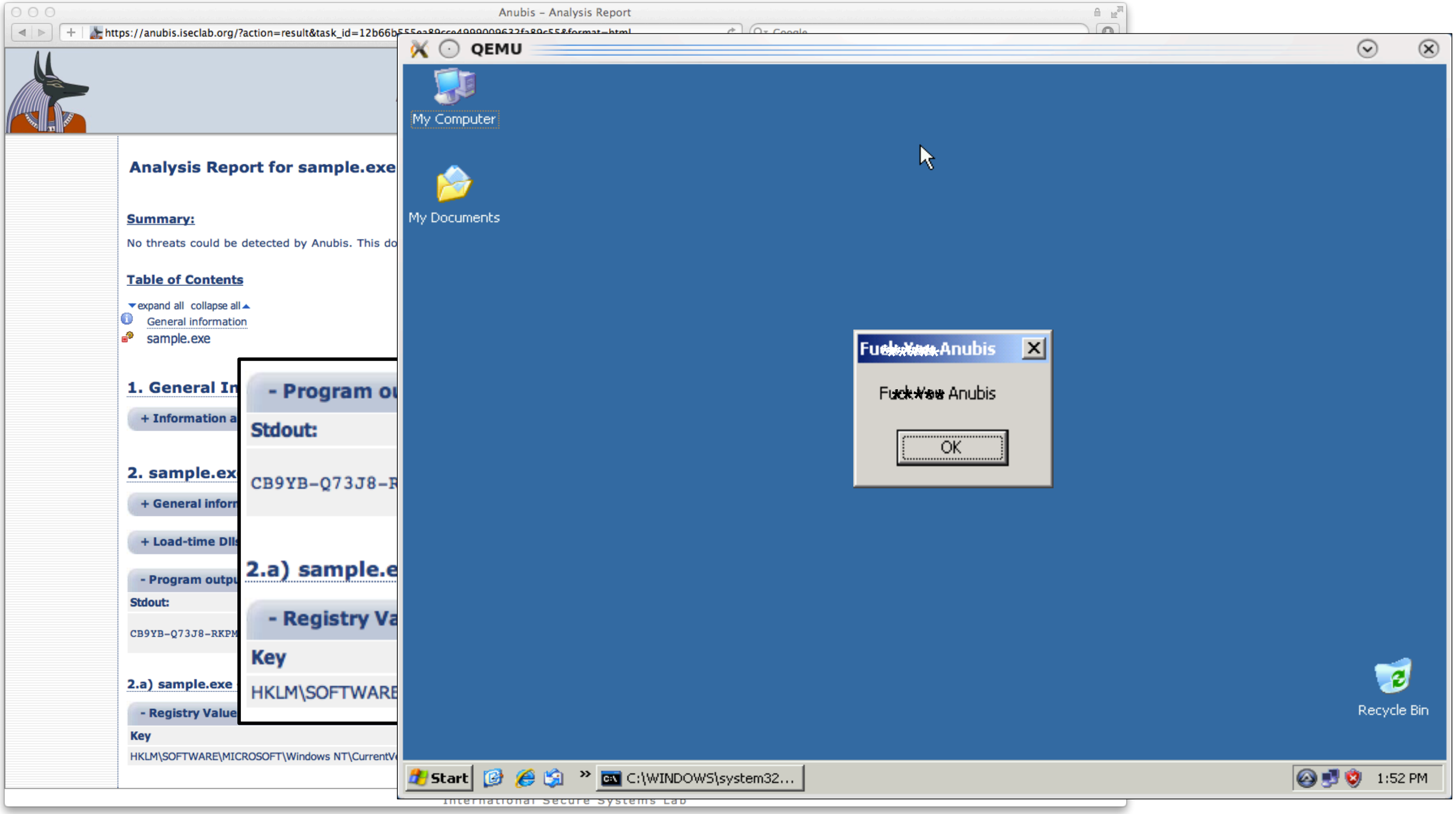
Martina Lindorfer
Clemens Kolbitsch
Paolo Milani Comparetti

Vienna University of Technology

Motivation

- Sandboxes widely used to observe malicious behavior
- Anubis: Dynamic malware analysis sandbox
  - Online since February 2007
  - Over 2,000 distinct users
  - Over 10,000,000 samples analyzed
- Malware tries to differentiate sandbox from real system
- No malicious activity in sandbox → analysis evasion
- Attackers can use samples to perform reconnaissance

Motivation



Analysis Report for sample.exe

Summary:

No threats could be detected by Anubis. This do

Table of Contents

expand all collapse all

General information

sample.exe

1. General In

+ Information a

2. sample.ex

+ General infor

+ Load-time Dlls

- Program outpu

Stdout:

```
CB9YB-Q73J8-RKPM
```

2.a) sample.exe

- Registry Value

Key

HKLM\SOFTWARE\MICROSOFT\Windows NT\CurrentV

- Program ou

Stdout:

```
CB9YB-Q73J8-R
```

2.a) sample.e

- Registry Va

Key

HKLM\SOFTWARE

Fu***** Anubis

Fu***** Anubis

Evasion Techniques

- “Environment-sensitive” malware checks for
  - Characteristics of the analysis environment
  - Characteristics of the Windows environment

- Emulation/Virtualization detection
- Timing
- Unique identifiers
- Running processes
- Restricted network access
- Public IP addresses

Evasion Countermeasures

- Transparent Monitoring Platform (e.g. Ether)
  - “undetectable”
  - Vulnerable to timing attacks
  - Vulnerable to detection of the specific Windows environment

- Evasion Detection
  - Execute malware in multiple environments
  - Detect deviations in behavior and identify root cause
  - Modify analysis sandboxes to thwart evasion techniques

Our Approach

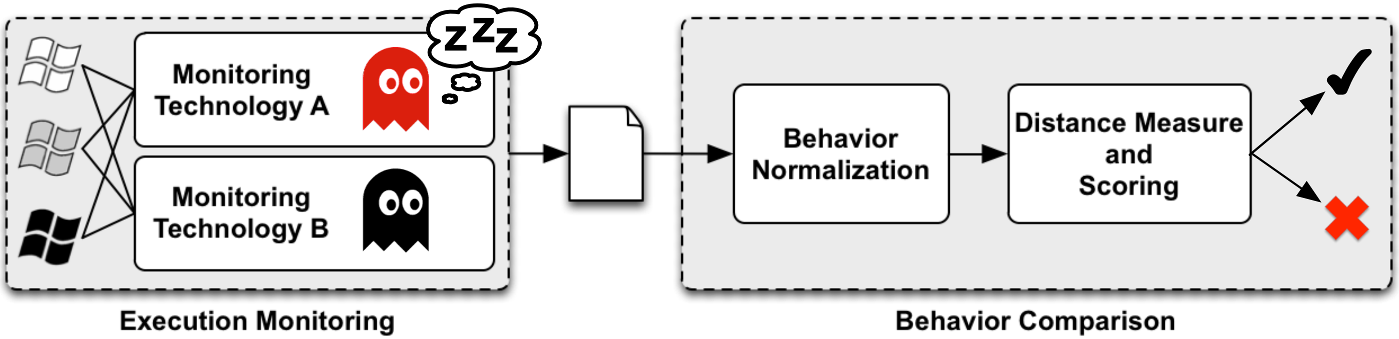
- DISARM
  "DetectIng Sandbox-AwaRe Malware"
  - Agnostic to root cause of divergence in behavior
  - Agnostic to employed monitoring technologies

- Automatically screen samples for evasive behavior
- Collect execution traces in different environments
- Eliminate spurious differences in behavior caused by different environments
- Compare normalized behavior and detect deviations
- Use findings to make sandbox resistant against evasion

Outline

- DISARM
- Evaluation
- Conclusion

DISARM



- Execution monitoring
  - Execute malware in multiple sandboxes
  - Different monitoring technologies & Windows installations
- Behavior comparison
  - Normalize behavior from different environments
  - Measure distance of behavior and calculate evasion score

Execution Monitoring

- Out-of-the-box monitoring
  - Anubis
  - modified version of Qemu emulator
  - Heavy-weight monitoring
- In-the-box monitoring
  - Light-weight monitoring → portable to any host
  - Windows kernel driver
  - Intercept system calls by SSDT hooking
- Multiple executions in each sandbox to compensate for randomness in behavior

Behavior Normalization

- Eliminate differences not caused by malware behavior
  - Differences in hardware, software, username, language, …

1. Remove noise
2. Generalize user-specific artifacts
3. Generalize environment
4. Randomization detection
5. Repetition detection
6. File system & registry generalization

Example Repetition Detection

File system Sandbox A

```
...
C:\WINDOWS\system32\w32tm.exe
C:\WINDOWS\system32\wdfmgr.exe
C:\WINDOWS\system32\wextract.exe
C:\WINDOWS\system32\wiaacmgr.exe
C:\WINDOWS\system32\winchat.exe
C:\WINDOWS\system32\WinFXDocObj.exe
C:\WINDOWS\system32\winhlp32.exe
C:\WINDOWS\system32\winlogon.exe
C:\WINDOWS\system32\winmine.exe
C:\WINDOWS\system32\winmsd.exe
C:\WINDOWS\system32\winspool.exe
C:\WINDOWS\system32\winver.exe
C:\WINDOWS\system32\wowdeb.exe
C:\WINDOWS\system32\wowexec.exe
C:\WINDOWS\system32\wpabaln.exe
C:\WINDOWS\system32\wpdshextautoplay.exe
C:\WINDOWS\system32\wpnpinst.exe
C:\WINDOWS\system32\write.exe
...
```

File system Sandbox B

```
...
C:\WINDOWS\system32\w32tm.exe
C:\WINDOWS\system32\wextract.exe
C:\WINDOWS\system32\wiaacmgr.exe
C:\WINDOWS\system32\winchat.exe
C:\WINDOWS\system32\winhlp32.exe
C:\WINDOWS\system32\winlogon.exe
C:\WINDOWS\system32\winmine.exe
C:\WINDOWS\system32\winmsd.exe
C:\WINDOWS\system32\winspool.exe
C:\WINDOWS\system32\winver.exe
C:\WINDOWS\system32\wmpstub.exe
C:\WINDOWS\system32\wowdeb.exe
C:\WINDOWS\system32\wowexec.exe
C:\WINDOWS\system32\wpabaln.exe
C:\WINDOWS\system32\wpnpinst.exe
C:\WINDOWS\system32\write.exe
...
```

```
C:\WINDOWS\system32\*.exe
```

Behavior Comparison

- Behavioral Profiles

```
file|C:\foo.exe|write:1
process|C:\Windows\foo.exe|create:0
network|tcp_conn_attempt_to_host|www.foobar.com
```

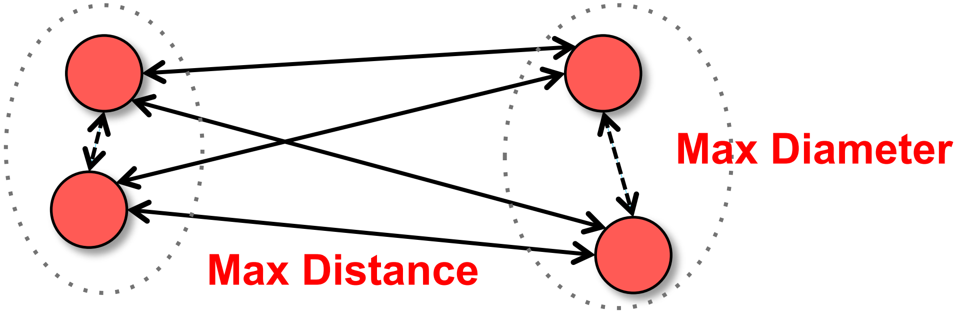
  – Set of actions on operating system resources

- Only persistent state changes
  – file/registry writes, network actions, process creations

- Distance between two profiles: Jaccard Distance

Evasion Score

- Evasion Score calculated in two steps:



1. Intra-sandbox distance (*diameter*) between executions in the same sandbox
2. Inter-sandbox distance (*distance*) between executions in different sandboxes

- If E ≥ threshold → classify as different behavior

Evaluation

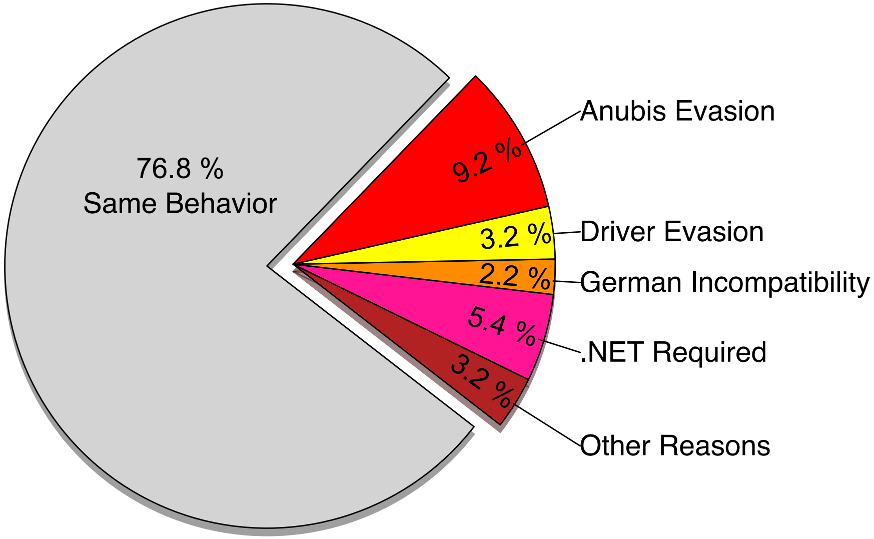
Setup

- 2 different monitoring technologies
- 3 different Windows images
- Driver inside Qemu to facilitate deployment

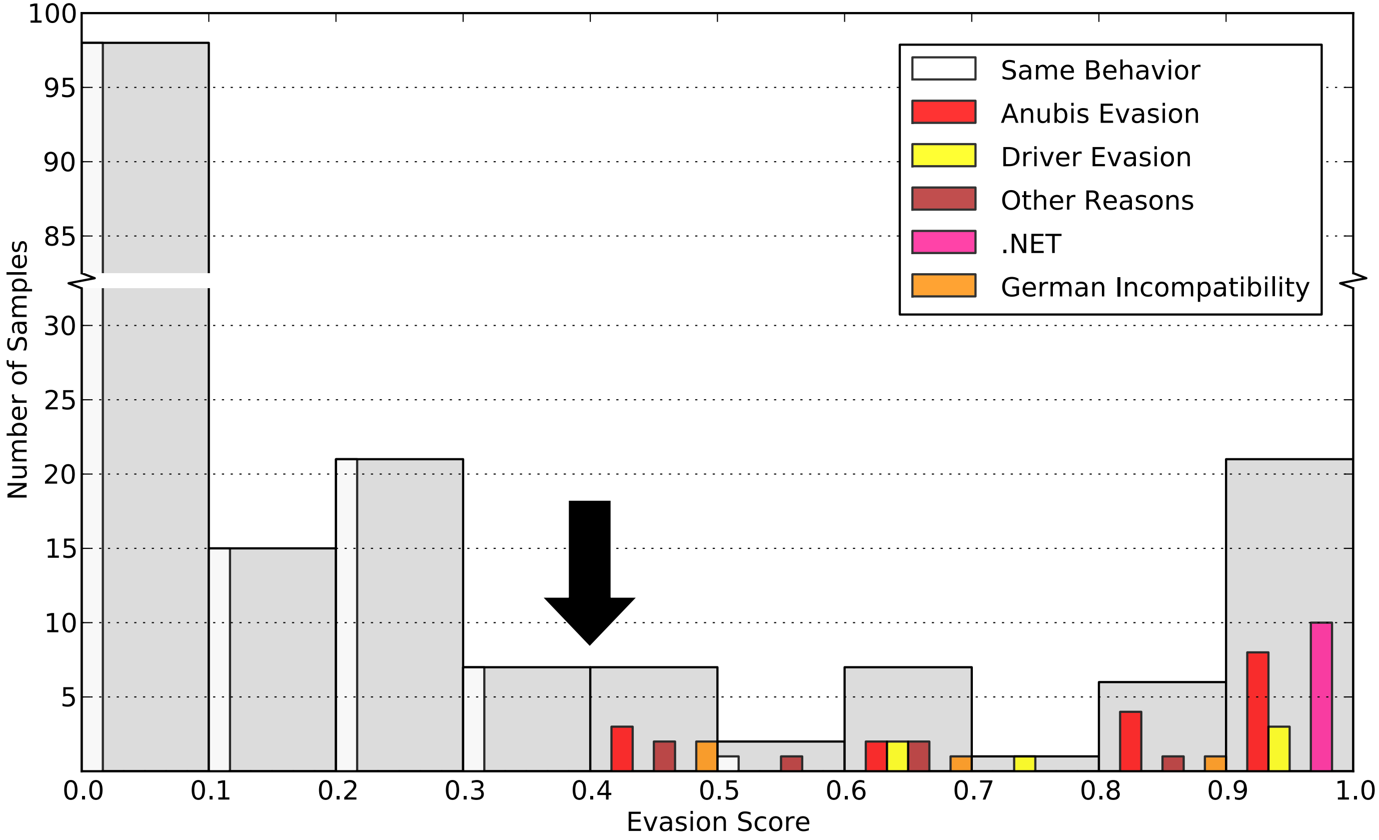
Sandbox	**Monitoring Technology**	**Image Characteristics**		
		Software	Username	Language
1	Anubis	Windows XP SP3, IE6	Administrator	English
2	Driver	Same as Anubis		
3	Driver	Windows XP SP3, IE7, JRE, .NET, Office	User	English
4	Driver	Windows XP SP2, IE6, JRE	Administrator	German

Training Dataset

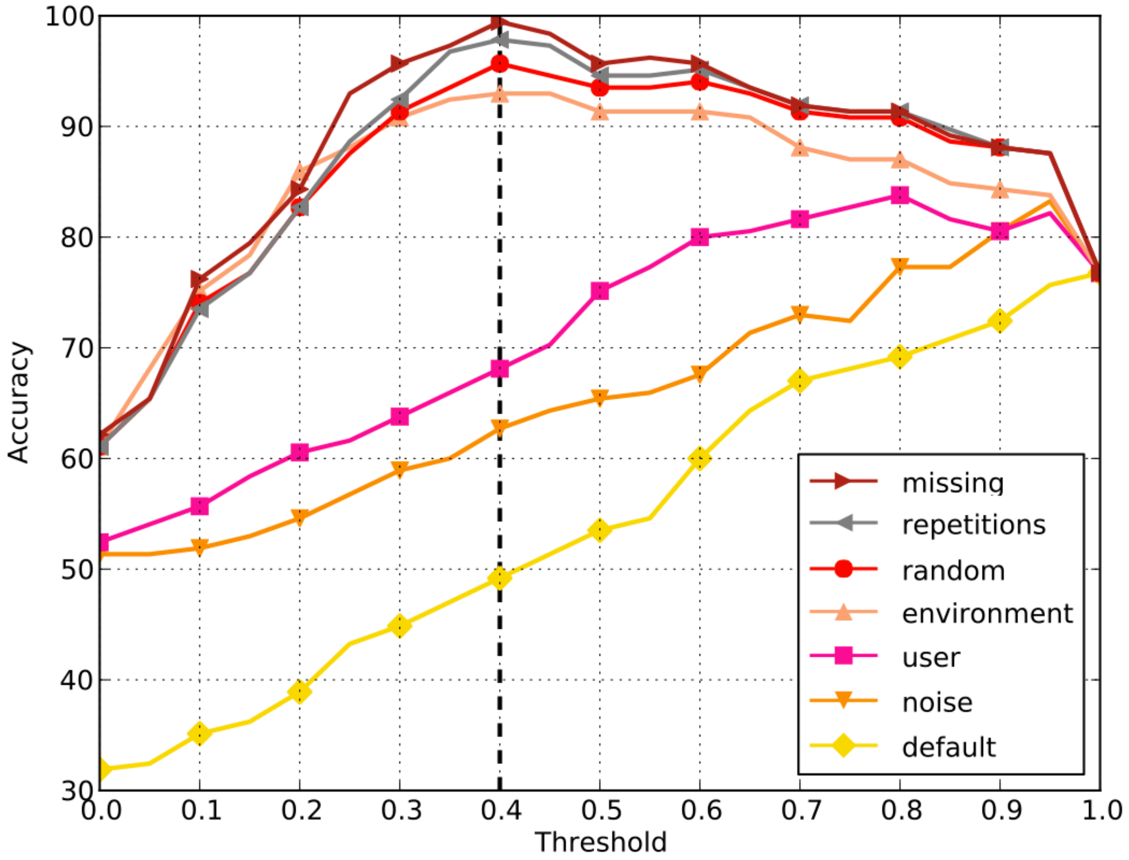
- 185 malware samples
  - Randomly selected from submissions to Anubis
  - Only one sample per malware family
- Optimize normalization and scoring
- Manual classification



Threshold Selection



Result Accuracy



- Proportion of correctly classified samples
- Each normalization improves results
- Accuracy > 90% for thresholds 0.3 – 0.6
- Max. accuracy 99.5 % for threshold 0.4

Test Dataset

- 1,686 malware samples
  - Selected from submissions to Anubis Dec 2010 – March 2011
  - Max. 5 samples per malware family
- Used threshold of 0.4 selected from training dataset
- 25.65 % of samples above threshold
- Manual examination of randomly selected samples
  - Discovered evasion techniques against Anubis
  - Discovered ways to improve the software configuration

Qualitative Results

Anubis Evasion

- Timing (Anubis 10x slower than driver in Qemu)
- Check for parent process
- Incomplete randomization of Anubis characteristics
  - Computer name
  - Machine GUID
  - Hard disk information

Driver Evasion

- Some samples restored SSDT addresses
  - Restrict access to kernel memory

Qualitative Results

Environment Sensitivity

- Configuration flaws in Anubis image
  - .NET environment
  - Microsoft Office
  - Java Runtime Environment (samples infect Java Update Scheduler)

False Positives

- *Sality* family creates registry keys and values dependent on username

Limitations

- Samples can evade DISARM by evading ALL sandboxes → eliminate shared sandbox characteristics
  - All sandboxes inside Qemu for our evaluation
  - Network configuration (restricted network access, public IPs)

- No automatic detection of root cause for evasion → use in combination with other tools:
  - Balzarotti et al.: Efficient Detection of Split Personalities in Malware (NDSS 2010)
  - Johnson et al.: Differential Slicing: Identifying Causal Execution Differences for Security Applications (Oakland 2011)

Conclusion

- Automatic screening of malware for evasive behavior
- Applicable to any analysis environment that captures persistent state changes
- Comparison of behavior across sandboxes
  - Different monitoring technologies & different Windows installations
  - Behavior normalization
- Light-weight in-the-box monitoring
  - Portable to any Windows XP environment (virtual or physical)
- Evaluation against large-scale test dataset
- Discovery of several new evasion techniques

Questions?

mlindorfer@iseclab.org

Related Work

- Chen et al.: Towards an Understanding of Anti-Virtualization and Anti-Debugging Behavior in Modern Malware (DSN 2009)
  - Comparison of single executions on plain machine, virtual machine and with debugger
  - Consider any difference in persistent behavior

- Lau et al.: Measuring virtual machine detection in malware using DSD tracer (Journal in Computer Virology 2010)
  - Focus on VM detection techniques in packers