Detecting Environment-Sensitive Malware

Martina Lindorfer, Clemens Kolbitsch and Paolo Milani Comparetti

COMET Competence Centers for Excellent Technologies
www.ffg.at/comet
secure sba-research.org
iseclab

Problem Outline

- Thousands of new malware samples surface every day
- Automation of analysis is necessary → Dynamic malware analysis
- Sample is executed in a monitored environment (emulator, virtual machine)
- Secure Systems Lab developed Anubis ("Analyzing Unknown Binaries")
- Public malware analysis sandbox: http://anubis.iseclab.org/
- **BUT:** Malware can discover that it is being analyzed
- Environment-sensitive malware checks for characteristics of the sandbox: CPU bugs, timing, Windows product ID, username, hardware serials, ...
- Malware exhibits no malicious activity in the sandbox ("analysis evasion")

→ **How can we detect analysis evasion?**

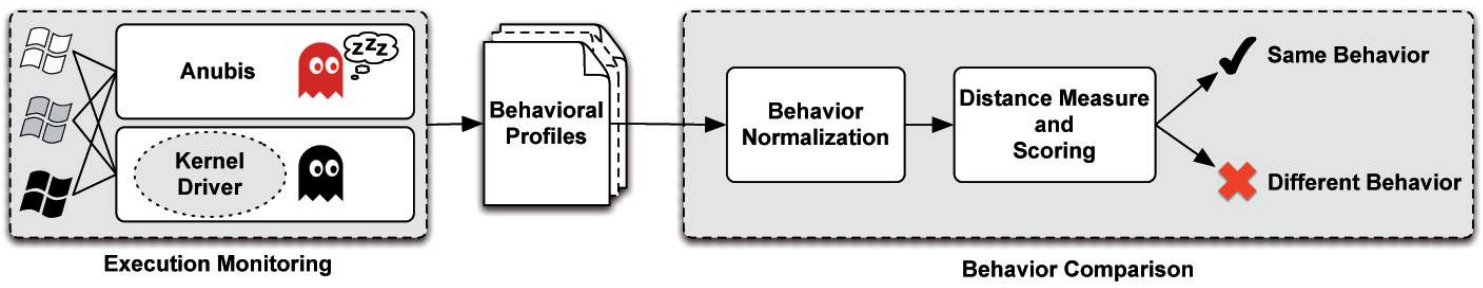

Figure 1: System Overview

Approach

- Build a verification system for Anubis to uncover evasion techniques

Execution Monitoring

- Windows kernel driver intercepts and logs system calls on a real host
- Logs are converted to behavioral profiles:
  Malware behavior as a set of operations on operating system resources

```
file|C:\foo.exe|write:1
process|C:\Windows\foo.exe|create:0
network|tcp_conn_attempt_to_host|www.foobar.com
registry|HKLM\System\CurrentControlSet\Services|set_value('xy'):1
```

Behavior Comparison

- Comparison of behavior in Anubis and on real host with driver
- Different Windows installations → normalize behavior
  1. Remove noise
  2. Generalize username
  3. Generalize environment (hardware, language)
  4. Randomization detection
  5. Repetition detection (file infectors)
  6. Filesystem and registry generalization (ignore missing resources)
- 3 executions in each sandbox (Anubis and real host)
- Intra-sandbox distance = variations between executions
- Inter-sandbox distance = variations between sandboxes
- Inter-sandbox distance – Intra-sandbox distance = evasion score [0,1]
- If evasion score ≥ threshold → different behavior; else same behavior
- Use findings to improve Anubis and prevent analysis evasion

Evaluation

Experiments with 4 different sandboxes

→ Anubis, Driver with Anubis image, Driver with German image, Driver with other image (different user, .NET, ...)

Training Dataset

- 185 malware samples
- Used to optimize normalization and scoring
- Manual classification
- Reached 99.5% accuracy @ threshold 0.4

Test Dataset

- 1686 malware samples
- Used to verify our system
- 25.56% samples above threshold
- Spot tests to find reasons for evasion

→ **Several new Anubis evasion techniques detected**
→ **Configuration flaws and missing software in Anubis (.NET, JRE, Microsoft Office, etc.)**
→ **Driver vulnerable to bypassing, but we can fix it**
→ **We can use these results to improve Anubis in order to observe a wider variety of malware behavior and thwart evasion!**

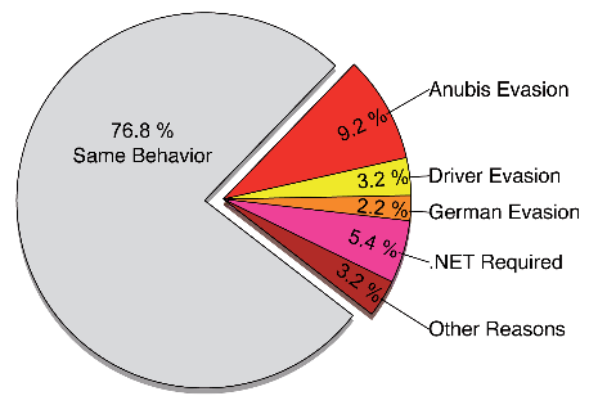

Figure 2: Manual classification of samples in the training dataset

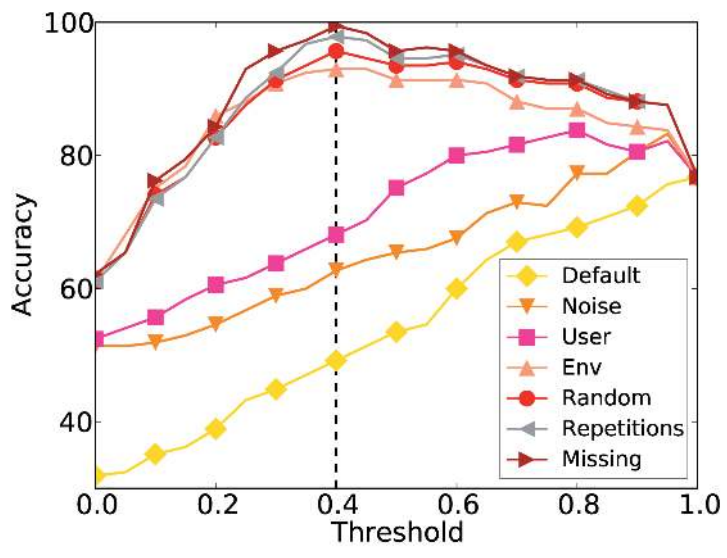

Figure 3: Efficiency of behavior normalization measured by result accuracy

Reference
M. Lindorfer, C. Kolbitsch, Paolo M. C.: "Detecting Environment-Sensitive Malware" in International Symposium on Recent Advances in Intrusion Detection (RAID 2011), 2011

Das Kompetenzzentrum SBA Research wird im Rahmen von COMET – Competence Centers for Excellent Technologies durch BMVIT, BMWFJ, das Land Wien gefördert. Das Programm COMET wird durch die FFG abgewickelt.