Improving the Efficiency of Dynamic Malware Analysis

Ulrich Bayer, Engin Kirda, Christopher Kruegel

Yang Yang

CISC850
Cyber Analytics
1. Introduction

- mutations of only a few malware programs
- reduce time
- 10,922 randomly chosen executable files
2. BACKGROUND: ANALYSIS TIME

\[ \text{Overall Analysis Time} = (|B| \cdot \sum_{b \in B} t_a(b)) / I \]

\[ t_a(b) = t_s(b) + t_e(b) + t_p(b) \]
3. REDUCING THE OVERALL ANALYSIS TIME

- Checkpoint time $T_c$
- $t_e(b)$: $T_c << t_e(b)$

- $t_{pre-empted}(b) = t_s(b) + T_c$
- $t_a(b) - t_{pre-empted}(b)$
3.1 Behavioral Profiles

- Timing information (timestamp value)
3.2 Comparison

- \( \text{dist}(bp(a), \ bp(b)) < d \)

- Jaccard distance:
  \[
  J(a, b) = 1 - \frac{|a \cap b|}{|a \cup b|}
  \]

- Extended Jaccard Distance
3.3 Efficient Nearest Neighbor Search

- Locality Sensitive Hashing (LSH)

\[ Pr[\text{collision}(a, b)] = 1 - (1 - (\text{sim}(a, b)^k)^l) \]
3.4 The Analysis Process

- Analyze a binary $b$ for time $T_c$
- Create a behavioral profile $bp(b)$
- Find behavioral profile $p$ with $\text{dist}(p, bp(b)) < d$
- Found no near profile
  - Store LS hashes
  - Continue analysis
- Found $p$
  - Stop Analysis
  - Return analysis report of $p$
4.1 Prototype Implementation

- On-the-fly generation of the behavioral profile
- Timestamps
- LSH
- Mapping feature strings to integer values
- LSH configuration
4.2 Experiment with a Reference Set

- Virut
- Allaple.1
- Allaple.2
- Trojan-PWS.Win32.LdPinch
Figure 2: False Positives
Figure 3: False Negatives
### 4.3 Real-World Experiments

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Pre-empted files</th>
<th>Time saved/pre-emption</th>
<th>Total time saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>45s, 0.12</td>
<td>3,087 (28.26%)</td>
<td>265s</td>
<td>227.2 hours</td>
</tr>
<tr>
<td>60s, 0.12</td>
<td>2,747 (25.15%)</td>
<td>250s</td>
<td>190.8 hours</td>
</tr>
<tr>
<td>60s, 0.12, $J_e$</td>
<td>3,659 (33.5%)</td>
<td>250s</td>
<td>284.1 hours</td>
</tr>
<tr>
<td>60s, 0.08</td>
<td>1,653 (15.13%)</td>
<td>250s</td>
<td>114.8 hours</td>
</tr>
<tr>
<td>60s, 0.08, $J_e$</td>
<td>2,539 (23.24%)</td>
<td>250s</td>
<td>176.2 hours</td>
</tr>
</tbody>
</table>

Table 1: Results of testing our approach in different configurations on a set of 10,922 binaries
Figure 4: CDF in [%] of distances $J(b_i, s_i)$ at time $t_e$
5. LIMITATIONS

• do not reveal true behavior during the short period

• against specific attacks
6. CONCLUSIONS

• 10,922 randomly chosen executable files

• 2,747 files (25.25%)

• 190.8 hours saved
Thank you!