BareCloud: Bare-metal Analysis-based Evasive Malware Detection

Dhilung Kirat, Giovanni Vigna, and Christopher Kruegel, University of California, Santa Barbara

Presented by: Kevin Coia

CS563 – Fall 2018
It can't hurt to open one little attachment. Can it?...
Evasive Malware
Approach presented in paper

Figure 1: Overview of the system
High level approach:

• Identify interesting malware samples that they would like to run in the BareCloud system.

• Run the chosen malware sample at the same time with identical setups in several different environments to collect the behavioral profiles for each environment. The environments used were as follows:
  1. Bare-metal
  2. Ether (Xen hypervisor based analysis environment)
  3. Anubis (Emulated environment based on Qemu)
  4. VirtualBox (Type 2 hypervisor)

• Compare behavioral profiles to identify how similar each of the Virtualized / Emulated environments compared to the Bare-metal’s profile, and if this difference is above a threshold classify it as Evasive Malware
Behavioral Profile Data Collection
Behavioral Profile Data Collection

Will this data collection be sufficient to build an accurate profile of the malware's behavior?

Does this approach even prevent an adversary from fingerprinting the Bare metal system to detect that it's being run in this monitored environment?

Can an adversary hide their persistent file operations in a way to look identical to the normal background file operations of the running OS?
Behavioral Profile Comparison

\[ \text{Jaccard Similarity} = \frac{A \cap B}{A \cup B} \]
Behavioral Profile Comparison

Profile A
- Create file X
- Create file Y
- Create file Z

Profile B
- Create file X
- Create file Y
- Modify file Z

Profile C
- Create file X
- Create file Y
- Connect to C&C

\[ \text{JaccardSimilarity}(A, B) = \frac{2}{4} = \text{JaccardSimilarity}(A, C) \]
Hierarchical similarity

Figure 2: Behavior similarity hierarchy
Hierarchical similarity

Profile A

Candidate Sets

Sim_1 = 1/2

Profile C
Hierarchical similarity

Profile A

Object Type: file
Object Name: C:\X, C:\Y, C:\Z
Operation Name: create, create, create
Operation Attribute: size, size, size

Profile C

Object Type: file
Object Name: C:\X, C:\Y
Operation Name: create, create
Operation Attribute: size, size

Candidate Sets

$\text{Sim}_1 = \frac{1}{2}$
Hierarchical similarity

\[ \text{Sim}(A, C) = \text{AVG}(\text{Sim}_1 \ldots \text{Sim}_4) = 0.79 \]
Hierarchical similarity

\[
\text{Sim}(A, B) = \text{AVG}(\text{Sim}_1 \ldots \text{Sim}_4) = 0.87
\]
Profile Similarity Comparison

\[ \text{JaccardSimilarity}(A, B) = \text{JaccardSimilarity}(A, C) \]

\[ \text{HierarchicalSim}(A, B) > \text{HierarchicalSim}(A, C) \]

0.87 > 0.79
Scoring Deviation from Bare-Metal behavior

Deviation Score

• Behavior Distance
  \[ \text{Distance}(A, B) = 1 - \text{Sim}(A, B) \]

• Deviation Score \( D \)
  Quadratic mean of the behavior distances with respect to the bare-metal analysis

• Deviation Threshold \( t \)
  - Evasive if \( D > t \)
Hierarchical vs Jaccard similarity

Using a sample of 111 evasive and 119 non-evasive samples

Ultimately led to them concluding that their Hierarchical similarity method is better at quantizing the similarity between two behavioral profiles
Hierarchical similarity

Is Hierarchical similarity a good approach to quantizing how similar two program behaviors were from their behavior profile?

Could this approach be improved further to better handle similar operations, for example creating a randomly named temp file?
Threshold Determination

![Graph showing precision and recall against threshold (t)](image-url)

- Threshold (t) = 0.84
Large-scale Evaluation

Ran the BareCloud system on 110,005 samples during a 4 months period starting in July 2013

Had the following selection criteria to decide which samples to test their system with:

- Low system and low network activity
- High system and high network activity
- High system but low network activity
- Low system but high network activity
Table 2: Evasion detection on different environments

<table>
<thead>
<tr>
<th>Environment</th>
<th>Detection count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anubis</td>
<td>4,947</td>
<td>84.78</td>
</tr>
<tr>
<td>Ether</td>
<td>4,562</td>
<td>78.18</td>
</tr>
<tr>
<td>VirtualBox</td>
<td>3,576</td>
<td>61.28</td>
</tr>
<tr>
<td>All</td>
<td>2,530</td>
<td>43.35</td>
</tr>
<tr>
<td>Total</td>
<td>5,835</td>
<td></td>
</tr>
</tbody>
</table>
What can we actually pull from this?

- Did they check that these detected evasive malwares, truly were evasive malware? Was there any misidentification?
- With 110,005 samples, what does this mean if anything since there is no ground truth for this data set?
- What would happen if you applied this to non malware? Can we verify that normal applications (or simply normal non-evasive malware) wouldn’t be mis-identified as “Evasive Malware”?

<table>
<thead>
<tr>
<th>Environment</th>
<th>Detection count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anubis</td>
<td>4,947</td>
<td>84.78</td>
</tr>
<tr>
<td>Ether</td>
<td>4,562</td>
<td>78.18</td>
</tr>
<tr>
<td>VirtualBox</td>
<td>3,576</td>
<td>61.28</td>
</tr>
<tr>
<td>All</td>
<td>2,530</td>
<td>43.35</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>5,835</strong></td>
<td></td>
</tr>
</tbody>
</table>
Questions