Experimental Study of Fuzzy Hashing in Malware Clustering Analysis

Yuping Li¹, Sathya Chandran Sundaramurthy¹
Alexandru G. Bardas¹, Xinming Ou¹
Doina Caragea¹, Xin Hu², Jiyong Jang²
1) Kansas State University 2) IBM Research
Motivation

• Huge volume of malware samples:

Why to use fuzzy hashing?

• Use fuzzy hashing to identify the new specie - “cat” for prioritized analysis
Cryptographic Hash Overview

Input

“The quick brown fox jumps over a lazy dog”

Hash generation function (e.g. md5)

Hash value (digest)

30ded807d65ee0370fc6d73d6ab55a95

“The quick brown fox jumps over A lazy dog”

Hash generation function (e.g. md5)

f8d517f0d3b6e44256aaeb2a265811e3

Comparison

Different (or 0 similarity)
Fuzzy Hash Overview

Input

“The quick brown fox jumps over a lazy dog”

“The quick brown fox jumps over A lazy dog”

Hash value (fingerprint)

Fuzzy hash fingerprint generation function

dfcaf6129c45f8827

Fuzzy hash fingerprint generation function

dfcaf6129c45f88a4

Comparison

Fuzzy hash fingerprint comparison function

Similarity: 95%
Previous Fuzzy Hashing Applications

• Using fuzzy hashing techniques to identify malicious code (ShadowServer, 2007)

• VirusTotal incorporated SSDeep hashes into their data set (VirusTotal, 2012)

• Fuzzy hashing techniques in applied malware analysis (CMU CERT, 2011)
Our Contributions

• We identify two design flaws within some of existing fuzzy hashing algorithms.

• We design and implement a generic experimentation framework for evaluating the performances of different fuzzy hash functions.

• We show that current fuzzy hashing algorithms can be further improved and proposed a new fuzzy hash function.
Design Flaw 1: Asymmetric Distance Computation

Input A

Input B

Fingerprint A

Fingerprint B

Fuzzy hash fingerprint generation function

Fuzzy hash fingerprint comparison function

Comparison

Sim(A, B) ≠ Sim(B, A)
Design Flaw 2: Incompatible Interpretations

- Containment analysis, expected optimal score: 1.0
  - Explanation: A is 100% contained in B

- Similarity analysis, expected optimal score: 0.5
  - Explanation: A and B have an overall similarity of 50%

Input A: [a, b, c]
Input B: [a, b, c, d, e, f]
Generic Experimentation Framework For Fuzzy Hash Evaluation

- Get unpacked samples
- For each malware, apply genHash()
- For each malware pair, apply compareHash()
- For each threshold, apply hacluster()
Malware Data Preparation

• Preparing malware dataset that are reliable:
  1) Prepare unpacked samples
     - Classification of packed and unpacked samples
  2) Prepare samples with accurate family name
     - Majority vote of VirusTotal labels
# Dataset Statistics (from VirusShare)

<table>
<thead>
<tr>
<th>Family</th>
<th>Size</th>
<th>Family</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viking</td>
<td>31</td>
<td>Vilsel</td>
<td>185</td>
</tr>
<tr>
<td>Fesber</td>
<td>57</td>
<td>Jeefo</td>
<td>36</td>
</tr>
<tr>
<td>Neshta</td>
<td>39</td>
<td>Turkojan</td>
<td>22</td>
</tr>
<tr>
<td>Skintrim</td>
<td>41</td>
<td>Bettersurf</td>
<td>300</td>
</tr>
<tr>
<td>Ramnit</td>
<td>38</td>
<td>Koutodoor</td>
<td>30</td>
</tr>
<tr>
<td>Zenosearch</td>
<td>99</td>
<td>Zbot</td>
<td>22</td>
</tr>
<tr>
<td>Hupigon</td>
<td>28</td>
<td>Fosniw</td>
<td>22</td>
</tr>
<tr>
<td>Domaiq</td>
<td>147</td>
<td>Wabot</td>
<td>27</td>
</tr>
<tr>
<td>Xpaj</td>
<td>22</td>
<td>Total</td>
<td>1146</td>
</tr>
</tbody>
</table>
Clustering Accuracy Measurement

• **Precision** is to capture how well the clustering algorithm separates samples of different families to different clusters.

• **Recall** is to capture how well the clustering algorithm assigns samples of same family to the same cluster.

• **Intersection point** between precision and recall can be seen as good balance between the two measurements.
Precision & Recall of SSDeep

• Modification of SSDeep:
  – Let similarity equals 0 if two SSDeep fingerprints can not be meaningfully compared

(0.964, 0.797)
Precision & Recall of sdHash

- Modification of sdHash:
  - Fix the asymmetric distance computation problem

\[
\text{Precision} \quad \text{Recall} \quad \text{of sdHash}
\]

\[
\text{Modification of sdHash:}
\]

\[
\text{– Fix the asymmetric distance computation problem}
\]

\[
8/10/15 
\]

\[
\text{CSET15} 
\]

\[
(0.469, \ 0.807) 
\]
Precision & Recall of mvHash-B

• Modification of mvHash-B:
  – Fix the asymmetric distance computation problem
Key Elements of Fuzzy Hashing Algorithms

• Features (characteristics of input):
  – hash values of substrings, entropy values of substrings, \textit{ngrams}

• Fingerprint (representation of features):
  – dynamic length strings, multiple bit-vectors, \textit{single bit-vector}

• Distance function (comparison of fingerprints):
  – edit distance, customized distance, jaccard distance, etc
New Fuzzy Hash Function:
nextGen-hash

- **Input:** 2f 2a 0a 20 2a 20 43 6f 70 79 72 69 67 68 74 20

- **4-gram Features:** 2f2a0a20, 2a0a202a, 0a202a20, 202a2043, ...

- **Fingerprint:**
  
  101010010100000101001010100100010100010100

- **Comparison:**
  
  \[
  \text{similarity} = \frac{\text{bitcount}(f_a \wedge f_b)}{\text{bitcount}(f_a \vee f_b)}
  \]
Precision & Recall of nextGen-hash

• Highlights of nextGen-hash:
  – Use ngrams as features
  – Use bit-vector as final fingerprint

(0.271, 0.791)
Further Improve Fuzzy Hashing Algorithms

<table>
<thead>
<tr>
<th>Algorithm Name</th>
<th>Whole Sample as Input (f-score)</th>
<th>Code Section as Input (f-score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssdeep</td>
<td>0.797</td>
<td>0.872</td>
</tr>
<tr>
<td>sdHash</td>
<td>0.807</td>
<td>0.877</td>
</tr>
<tr>
<td>mvHash-B</td>
<td>0.792</td>
<td>0.893</td>
</tr>
<tr>
<td>nextGen-hash</td>
<td>0.791</td>
<td>0.919</td>
</tr>
</tbody>
</table>
Summary

• We identify several design flaws within existing fuzzy hashing algorithms and provide algorithm and suggestion to fix the problems.

• We design an evaluation framework and demonstrate that it can be used to compare the effectiveness of different fuzzy hash functions.

• We propose a new fuzzy hashing algorithm and show that performances of existing fuzzy hash functions can be further improved.