FLARE: Fuzzy Similarity Hashing Framework

9 Oct. 2019
Hello!

Hossein Jazi

Senior Threat Researcher
@ Cysiv/Trend Micro

- APT campaigns Tracker
- Malware Reverse Engineer
- Threat Intelligence Analyst
Agenda

◎ Introduction to Fuzzy hashing
◎ Overview of fuzzy hashing algorithms
◎ Comprehensive experiments
Introduction to Fuzzy hashing

- Cryptographic hashing: MD5, Sha256
- Binary similarity:
  - Approximate Matching/Fuzzy Similarity
Algorithms

Overview of the known fuzzy hashing algorithms
Fuzzy hashing basics

<table>
<thead>
<tr>
<th>sector</th>
<th>c</th>
<th>o</th>
<th>n</th>
<th>f</th>
<th>1</th>
<th>9</th>
<th>talk</th>
</tr>
</thead>
<tbody>
<tr>
<td>115</td>
<td>101</td>
<td>99</td>
<td>116</td>
<td>111</td>
<td>114</td>
<td>99</td>
<td>111</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sector</th>
<th>c</th>
<th>o</th>
<th>n</th>
<th>f</th>
<th>1</th>
<th>9</th>
<th>talk</th>
</tr>
</thead>
<tbody>
<tr>
<td>115</td>
<td>101</td>
<td>99</td>
<td>116</td>
<td>111</td>
<td>114</td>
<td>99</td>
<td>111</td>
</tr>
</tbody>
</table>
Fuzzy hashing basics

<table>
<thead>
<tr>
<th>sector</th>
<th>core</th>
<th>config</th>
<th>19</th>
<th>fuzz</th>
<th>talk</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>115</td>
<td>101</td>
<td>99</td>
<td>116</td>
<td>111</td>
</tr>
<tr>
<td>435</td>
<td>440</td>
<td>372</td>
<td>398</td>
<td>443</td>
<td>107</td>
</tr>
<tr>
<td>z</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>r</td>
</tr>
</tbody>
</table>

- **Vz+ps and z40o7r**
- **Problem**: Fixed size blocks
- **Solution**: Use boundaries in another way
SSDEEP (Window = 3, Trigger Point= 5, Modulo = 8)

<table>
<thead>
<tr>
<th>115</th>
<th>216</th>
<th>315</th>
<th>316</th>
<th>326</th>
<th>341</th>
<th>224</th>
<th>324</th>
<th>320</th>
<th>323</th>
<th>261</th>
<th>208</th>
<th>208</th>
<th>276</th>
<th>341</th>
<th>361</th>
<th>360</th>
<th>335</th>
<th>321</th>
<th>313</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td><strong>5</strong></td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td><strong>5</strong></td>
<td>0</td>
<td>0</td>
<td>4</td>
<td><strong>5</strong></td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>s</td>
<td>e</td>
<td>c</td>
<td>t</td>
<td>o</td>
<td>r</td>
<td>c</td>
<td>o</td>
<td>n</td>
<td>f</td>
<td>1</td>
<td>9</td>
<td>f</td>
<td>u</td>
<td>z</td>
<td>z</td>
<td>t</td>
<td>a</td>
<td>l</td>
<td>k</td>
</tr>
<tr>
<td>115</td>
<td>101</td>
<td>99</td>
<td>116</td>
<td>111</td>
<td>114</td>
<td>99</td>
<td>111</td>
<td>110</td>
<td>102</td>
<td>49</td>
<td>57</td>
<td>102</td>
<td>117</td>
<td>122</td>
<td>122</td>
<td>116</td>
<td>97</td>
<td>108</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>656</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>471</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>398</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Q</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>120</th>
<th>235</th>
<th>336</th>
<th>315</th>
<th>316</th>
<th>326</th>
<th>341</th>
<th>224</th>
<th>324</th>
<th>320</th>
<th>323</th>
<th>261</th>
<th>208</th>
<th>208</th>
<th>276</th>
<th>341</th>
<th>361</th>
<th>360</th>
<th>335</th>
<th>321</th>
<th>313</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td><strong>5</strong></td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td><strong>5</strong></td>
<td>0</td>
<td>0</td>
<td>4</td>
<td><strong>5</strong></td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>s</td>
<td>e</td>
<td>c</td>
<td>t</td>
<td>o</td>
<td>r</td>
<td>c</td>
<td>o</td>
<td>n</td>
<td>f</td>
<td>1</td>
<td>9</td>
<td>f</td>
<td>u</td>
<td>z</td>
<td>z</td>
<td>t</td>
<td>a</td>
<td>l</td>
<td>k</td>
</tr>
<tr>
<td>120</td>
<td>115</td>
<td>101</td>
<td>99</td>
<td>116</td>
<td>111</td>
<td>114</td>
<td>99</td>
<td>111</td>
<td>110</td>
<td>102</td>
<td>49</td>
<td>57</td>
<td>102</td>
<td>117</td>
<td>122</td>
<td>122</td>
<td>116</td>
<td>97</td>
<td>108</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>776</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>471</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>398</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>235</th>
<th>336</th>
<th>315</th>
<th>316</th>
<th>326</th>
<th>341</th>
<th>224</th>
<th>324</th>
<th>320</th>
<th>323</th>
<th>261</th>
<th>208</th>
<th>208</th>
<th>276</th>
<th>341</th>
<th>361</th>
<th>360</th>
<th>335</th>
<th>321</th>
<th>313</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td><strong>5</strong></td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td><strong>5</strong></td>
<td>0</td>
<td>0</td>
<td>4</td>
<td><strong>5</strong></td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>s</td>
<td>e</td>
<td>c</td>
<td>t</td>
<td>o</td>
<td>r</td>
<td>c</td>
<td>o</td>
<td>n</td>
<td>f</td>
<td>1</td>
<td>9</td>
<td>f</td>
<td>u</td>
<td>z</td>
<td>z</td>
<td>t</td>
<td>a</td>
<td>l</td>
<td>k</td>
</tr>
<tr>
<td>120</td>
<td>115</td>
<td>101</td>
<td>99</td>
<td>116</td>
<td>111</td>
<td>114</td>
<td>99</td>
<td>111</td>
<td>110</td>
<td>102</td>
<td>49</td>
<td>57</td>
<td>102</td>
<td>117</td>
<td>122</td>
<td>122</td>
<td>116</td>
<td>97</td>
<td>108</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>776</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>471</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>398</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8
SDHASH

Features with high entropy

SHA1

BF1

SHA1

BF1

SHA1

BF1

Low entropy

0000000

High entropy

10010111010
TLSH

Locality Sensitive Hashing

Pearson Hash: X G 5 X L

Buckets: X:2 G:1 5:1 L:1

Define quartiles

Construct digest header

Construct digest body

75% of the bucket counts are \( \geq q_1 \)
50% of the bucket counts are \( \geq q_2 \)
25% of the bucket counts are \( \geq q_3 \)

header = checksum(bytes) + log(bytes) + q1_ratio + q2_ratio

for \( i = 0 \) to 127
  if bucket[i] \( \leq q_1 \) Emit(00)
  else if bucket[i] \( \leq q_2 \) Emit(01)
  else if bucket[i] \( \leq q_3 \) Emit(10)
  else Emit(11)
MRSH-V2

- Combination of SSDEEP and SDHASH
- Fixed Window size (7)
- This is a Security Conference

Reach to Trigger Point

FLV hash

BF
Fuzzy Similarity Process

1. Sample 1
   - Feature Extraction
   - Digest Generation
   - Similarity Score

2. Sample 2
   - Feature Extraction
   - Digest Generation
Experiments

Real world use cases
Experiments Process

- Over 7000 samples
- Compare all files against each others and calculate the overage
- Normalize the algorithms that have different scales
- Define 4 measurement parameters: TP, PTP, PFN, FN
- Tested of different malware behaviors
Use Case 1: VxCrypter
VxCrypter
.Net Ransomware
Use case 2: Separ Info Stealer

Image is 32bit executable
RES/OVL: 2 / 86% 2013

Lamer Info - Help Hint - Unpack info
try: internal Zip Ripper

Rar Sfx Archive, Overlay: RAR archive > Offset: 0000h -> SOLID L
Lamer Info - Help Hint - Unpack info
try WinRar.exe to view files - www.rarlab.com
Use case 3: GoldBrute Botnet
GoldBrute Botnet
Use Case 4: RougeRobin Trojan

- Trojan used by DarkHydrus APT
- It has 2 different variants: PowerShell, C#
- The Analysis showed similarities between these variants. The actor just ported their code to a compiled version.
  - Using WMI to check BIOS version, number of CPU cores, size of physical memory and enumerate running processes for Wireshak and Sysinternals
  - Communicate with its configured C&C using a custom DNS tunneling protocol
RougeRobin

The .Net variant used Google API as its C&C
RougeRobin Trojan
Use case 5: APT Analysis

Analysis of effects of Fuzzy hashing on finding similarities within APTs
APT28 (Fancy Bear)

- Politically motivated threat actor group that has been active since at least 2007 and is believed to be Russian-sponsored.
- Targets: Defense contractors, Embassies, Government Hospitality, Military, Media
- Use spear phishing as initial infection vector

The group is unique because of:
- Sophistication of phishing content
- Custom toolsets and platforms
- Network of phishing websites

Malware families:
- Autoit Backdoor, AZZY, Cannon, Delphi Backdoor, Downdelph, Komplex, Lojax, Olympic Destroyer, Seduploader, Xagent, Xtunnel, Zekapab, Zebrocy
FIN6 (Skeleton Spider)

- Financially motivated Russian-speaking criminal group that has been active since at least 2014.
- Use spear phishing and stolen credentials as Initial infection vector
- Focus: Steal credit card data and sell
- Targets: hospitality and retail sectors
- Malware families:
FIN6
FIN7 (Carbon Spider)

- Russian-speaking financially motivated threat actor group that has primarily targeted the U.S. retail, restaurant, and hospitality sectors since 2015.
- Spear phishing with link or macro as initial infection vector
- Primary malware type: PoS
- Malware families:
  - Babymetal, Bateleau, Birdodg, Carbanak, DNSRat, Driftpin, Halfbaked, Powersource, SqlRat, Tinimet
APT19 (Deep Panda)

- Deep Panda is state sponsored Chinese threat group consist of freelancers.
- Key identifications:
  - Custom malware
  - Access to several Zero days, exploits
  - Its own infrastructure

Malware families

Derbusi and Sakurel
TA505

- TA505 is a financially motivated threat actor this is known to perform large malicious spam
- TA505 has been active since 2014 by distributing Dridex in mass scale.
- Key differentiations:
  - Extraordinary volume of messages
  - The variety of malware families
- Malware families:
  - AdvisorBot, Dedex, Dridex, Locky, Pusadvisor, FlawedAmmey
Use case 6: IOT

- IOT threats target IOT devices such as IP cameras, home routers, storage devices and etc.
- Eight different IOT malware: Bashlite, Amnesia, Mirai, Okne, Persirai, Satori, Silex and VPNFilter
- VPNFilter: VPNFilter is malware infecting a number of different kinds of network routers and storage devices.
- Mirai: Target IP cameras and home routers to add them to its botnet network.
- Silex: A type of malware that wipe the firmware of IoT devices.
- Amnesia: A botnet that targets DVR by exploiting unpatched RCE vulnerability
Use case 7: Mobile Malware

- Three families have been analyzed.
- Advertising dropper: A dropper that hide itself behind CamScanner app
- Clicker Trojan: 33 Android Apps in the Google Play Store with over 100 million installations
  - performs ad fraud
- Joker: 24 Android apps with more than 472,000 downloads in total.
  - Downloaded as a second stage component
  - Harvests sensitive data such as contact lists, text messages, and device information.
Use case 8: Packers, Protectors, Crypters

Experimented effects of 18 different packers finding malware similarity.

### SDHASH

<table>
<thead>
<tr>
<th>Tool</th>
<th>TP</th>
<th>FTP</th>
<th>FPN</th>
<th>PN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confuser</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASAdo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASPack</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CryptoObfuscator</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Reactor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smart Assembly</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPX</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enigma</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exe32pack</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPRESS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSPACK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obfuscium</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pecompact</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pelock</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Themida</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unpack</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yoda</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### TLSH

<table>
<thead>
<tr>
<th>Tool</th>
<th>TP</th>
<th>FTP</th>
<th>FPN</th>
<th>PN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confuser</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASAdo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASPack</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CryptoObfuscator</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Reactor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smart Assembly</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPX</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enigma</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exe32pack</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPRESS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSPACK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obfuscium</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pecompact</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pelock</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Themida</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unpack</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yoda</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Efficiency & False Positive Analysis
Efficiency
False Positive Rate
Attacks against Fuzzy hashing Algorithms

SSDEEP
- Adding 7 to 10 random byte
- Target trigger values

Sdhash, Mrsh
- Adding data with 64 artificial features to the front of the file to be hashed causing the Bloom filters to overlap in a different place

TLSH
- TLSH is extremely robust to random manipulations throughout the file, such as insertions and deletions; however, TLSH produces a high false positive.
- Containment detection is low

Myhash-B
- Random edits
Conclusion

- Highest accuracy: TLSH
- False Negative rates: TLSH and LZJD
- Fuzzy hashing can help spotting new samples of an APT
- Fuzzy hashing methods are not reliable for IOT threats
- .Net obfuscators preserve the file structure
- Multi hashing avoids targeted attacks
Thanks!

Any questions?
Others

SFHU: Use feature identification of SSDEEP to find the trigger points and then convert each ASCII value to hex.

<table>
<thead>
<tr>
<th>Secto r</th>
<th>83</th>
<th>101</th>
<th>99</th>
<th>116</th>
<th>111</th>
<th>114</th>
<th>99</th>
<th>111</th>
<th>110</th>
<th>49</th>
<th>57</th>
</tr>
</thead>
<tbody>
<tr>
<td>5365-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3: 5365-2</td>
<td>63746F72-4</td>
<td>636F-2</td>
<td>6E3139-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PeHash: Generate hash based on file header structural data and data section structural information
Others

- LZJD: converts a byte string $b$ into a set of sub-strings $s_b$, using a simplified version of the Lempel-Ziv77 algorithm and then convert them to digest.

- Mvhash-B: Use majority vote on bit level to transforms an input byte sequence into long runs of 0s and 1s. Next, these runs are encoded by Run Length Encoding (RLE).