Angad

Malware Detection via Multi-Dimensional Visualization

Ankur Tyagi (@7h3rAm)
whoami

- Sr. Malware Research Engineer @ Qualys
- @7h3rAm on the Interweb
Outline

- Issues with Malware Analysis
- What is Multi-Dimensional Visualization?
- Context-aware Multi-Dimensional Visualization
- Usecases
Issues with Malware Analysis

- Nymaim (is a trojan, downloads additional malware) CFGs
- pre deobfuscation (#1): 200+ blocks
- post deobfuscation (#2): 8 blocks

Nymaim obfuscation and cyber variant of "knock, knock, who’s there?" (@Avira)
Issues with Malware Analysis

- Malware Analysis is difficult:
  - Obfuscation, anti-* (disassembly, debugging, virtualization)
  - Volume of new unknowns is huge (250k+ unknowns daily, 90%+ clean)

- Lots of interesting proposals to solve these issues over last many years

- Vendors use proprietary (closed/IP-protected) solutions

- Enthusiasts build and use “personal” projects (difficult to gain traction)
Issues with Malware Analysis

- Issues with current automated techniques:
  - packers/virtualizers/ compressors (themida, vmpack, aspack)
  - runtime dependencies
  - trigger conditions
  - sandbox detection
  - user interaction

What Does Obfuscated Software Look Like? (The University of Arizona)
Issues with Malware Analysis

- We need to focus on generic signatures to reduce overhead and increase future coverage.
- Note: Excessive generalization in detection methods can increase FPs.
- We still need specialized detection methods for threat attribution.
Issues with Malware Analysis

- Lots of interesting research has been done towards applying visualization techniques for malware analysis
  - Malware Analysis Using Visualized Image Matrices (Han et al.)
  - Malware Images: Visualization and Automatic Classification (Nataraj et al.)
  - Visual Analysis of Malware Behaviour Using Treemaps and Thread Graphs (Holz et al.)
  - and many more ...
What is Multi-Dimensional Visualization?

- A **Hilbert curve** is a continuous fractal space-filling curve
- Recursive, self-similar, space-filling
- Fills a square area using **Gray code** traversal
- Unit dimension is 2, images are always a square
What is Multi-Dimensional Visualization?

- IPv4: $2^{32} \approx 4$ billion addresses

xkcd 195: Map of the Internet

Mapping the whole Internet with Hilbert curves (@benjojo12)
What is Multi-Dimensional Visualization?

- Important characteristics:
  - Clustering (locality preserving in 2D)
  - Spatial indexing

- Can be indexed and matched against unknown input for clustering and classification

- Can be used as a visual hash (contrast with cryptographic hashes)
What is Multi-Dimensional Visualization?

- Let's discuss how to use Hilbert curves to visualize malware structure and APIs

- **Aldo Cortesi (@cortesi)** has posted about this in depth on his blog

- Check out his project **binvis.io** for more details
What is Multi-Dimensional Visualization?

- Hilbert curves created using scurve library from @cortesi
- All files have unique hashes but (structurally) similar Hilbert curve representations
What is Multi-Dimensional Visualization?

- Difficult to identify offsets for prominent structures
- Unable to retain shape for similar content at different offsets
Context-aware Multi-Dimensional Visualization

- Daniel Plohmann has used Hilbert curve to visualize import APIs
- Check out the Malpedia service to see how imported APIs from PE files are used to create a visual hash (ApiQR)
Context-aware Multi-Dimensional Visualization

- Use file-format specific context to create Hilbert curves

- For PE files, visualize:
  - Import APIs from IAT
  - System Calls from behavior report

ByteAtlas: ApiQR representation: Hilbert curve for our 1024 bit ApiVector with the semantic categories
Context-aware Multi-Dimensional Visualization

- APIs from IAT and behavior report:
  - APIs are extracted from IAT and sandbox execution report
  - Names are normalized (eg: `RegOpenKey` <- `RegOpenKeyExW`)
  - A 32x32 bit vector (1024 cells) is created and mapped on Hilbert curve
  - A sliding window scheme is used to create animated Hilbert curve from extracted APIs
Usecase: #1 APT33

---

Use Case 33: The APT33 campaign involves the targeting of specific organizations and sectors, often using advanced persistent threats (APTs) that employ sophisticated techniques to evade detection.

The APT33 campaign has been associated with the theft of sensitive information, financial gain, and the disruption of critical infrastructure. The attackers use a combination of social engineering, malware, and other techniques to infiltrate and maintain access to their targets.

The campaign is characterized by the use of custom malware, such as the Duqu and Caracal malware families, which have been used to steal sensitive information from targeted organizations.

The APT33 campaign has been active since 2009 and has targeted a wide range of organizations, including government agencies, financial institutions, and technology companies.

---

@7h3rAm
### Usecase: #1 APT33

<table>
<thead>
<tr>
<th>SHA256</th>
<th>ByteView</th>
<th>IATView</th>
<th>BehaviorView</th>
</tr>
</thead>
<tbody>
<tr>
<td>0f80b73706df263d337c4da52aad67c3699d1deea00aafe78e604d61a54c649d</td>
<td><img src="image1" alt="ByteView" /></td>
<td><img src="image2" alt="IATView" /></td>
<td><img src="image3" alt="BehaviorView" /></td>
</tr>
<tr>
<td>Packer: NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130aa7bd89aa4b68f1561d33bbd0068ad96abc0cd78c74cd3eb89cf19076916</td>
<td><img src="image4" alt="ByteView" /></td>
<td><img src="image5" alt="IATView" /></td>
<td><img src="image6" alt="BehaviorView" /></td>
</tr>
<tr>
<td>Packer: NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>285aa5fe83503fee229bb4a1ab861427933c7ab047f63472543f75d8872735a9</td>
<td><img src="image7" alt="ByteView" /></td>
<td><img src="image8" alt="IATView" /></td>
<td><img src="image9" alt="BehaviorView" /></td>
</tr>
<tr>
<td>Packer: NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28cb4114ee5615e9fa039c913d41db660c089b206565d25a4342eeaf71d9b7f9</td>
<td><img src="image10" alt="ByteView" /></td>
<td><img src="image11" alt="IATView" /></td>
<td><img src="image12" alt="BehaviorView" /></td>
</tr>
<tr>
<td>Packer: NA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Usecase: #2 Dorv

---

```python
inf -c [Toolbox/misc/angad [Master L1]
$ python angad.py -l input.jsonp
```

Angad (v0.1) - Ankur Tyagi (g7h3rAm)

```
[env] $ [Toolbox/misc/angad [Master L1]
$ python angad.py -l input.jsonp
```

```
 Angad (v0.1) - Ankur Tyagi (g7h3rAm)
[env] $ [Toolbox/misc/angad [Master L1]
```

```
./toolbox/misc/angad [Master L1]
```

Use case of #2 Dorv
## Use Case: #2 Dorv

<table>
<thead>
<tr>
<th>#</th>
<th>SHA256</th>
<th>ByteView</th>
<th>IATView</th>
<th>BehaviorView</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0275acf5332c354ccda0a3daea804005e42219c4500edd23dcd7091d277d0434</td>
<td><img src="image1.png" alt="ByteView" /></td>
<td><img src="image2.png" alt="IATView" /></td>
<td><img src="image3.png" alt="BehaviorView" /></td>
</tr>
<tr>
<td></td>
<td><strong>Packer:</strong> UPX 2.90 [LZMA] -&gt; Markus Oberhumr, Laszlo Molnar &amp; John Reiser</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>038f68c4afa8213c38fc61f6e854a2d9ef636f3f3ab86003721f2af61e1fcd7</td>
<td><img src="image4.png" alt="ByteView" /></td>
<td><img src="image5.png" alt="IATView" /></td>
<td><img src="image6.png" alt="BehaviorView" /></td>
</tr>
<tr>
<td></td>
<td><strong>Packer:</strong> UPX 2.90 [LZMA] -&gt; Markus Oberhumr, Laszlo Molnar &amp; John Reiser</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>03fc48e75f754ad29e4420ebf6f2f97ce870e5c59cf6f2f80b7e4daedd19a31ff6</td>
<td><img src="image7.png" alt="ByteView" /></td>
<td><img src="image8.png" alt="IATView" /></td>
<td><img src="image9.png" alt="BehaviorView" /></td>
</tr>
<tr>
<td></td>
<td><strong>Packer:</strong> UPX 2.90 [LZMA] -&gt; Markus Oberhumr, Laszlo Molnar &amp; John Reiser</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Usecase: #2 Dorv

<table>
<thead>
<tr>
<th>SHA256</th>
<th>ByteView</th>
<th>IATView</th>
<th>BehaviorView</th>
</tr>
</thead>
<tbody>
<tr>
<td>03fc48e75f754ad29e4420ebfd2ec87ce5c59cf6f2fcee0b7e4daedd19a31ff6</td>
<td><img src="image1.png" alt="ByteView Image" /></td>
<td><img src="image2.png" alt="IATView Image" /></td>
<td></td>
</tr>
<tr>
<td>02f135ec7a39f40c26973f36e0209401a8f0343aecd6b743164f2e621d45fb7aa</td>
<td><img src="image3.png" alt="ByteView Image" /></td>
<td><img src="image4.png" alt="IATView Image" /></td>
<td></td>
</tr>
<tr>
<td>Pack: PECompact 2.xx → BitSum Technologies</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03ddace7f36bc9c57e679237b2c5f0fc31f3763bc467bcfebce7132717b69faa</td>
<td><img src="image5.png" alt="ByteView Image" /></td>
<td><img src="image6.png" alt="IATView Image" /></td>
<td></td>
</tr>
<tr>
<td>Pack: PECompact 2.xx → BitSum Technologies</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03fd30ac12245f5d135d596dca97800893507df8aad973b8c01f02a637d99519</td>
<td><img src="image7.png" alt="ByteView Image" /></td>
<td><img src="image8.png" alt="IATView Image" /></td>
<td></td>
</tr>
<tr>
<td>Pack: PECompact 2.xx → BitSum Technologies</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Usecase: #3 Moogkel
## Use Case: #3 Mooqkel

<table>
<thead>
<tr>
<th>SHA256</th>
<th>ByteView</th>
<th>IATView</th>
<th>BehaviorView</th>
</tr>
</thead>
</table>
| 1. 01b6ffe35f9ca26d6486226faca1eca371d535fc72b9fa13c9a3a2706b8dd1af5  
  Packer: UPX_LZMA, UPX | ![ByteView](1) | ![IATView](1) | ![BehaviorView](1) |
| 2. 043dee2920046a75ce4909e340e4ab31712664ae792f570b8dca169600770c78  
  Packer: UPX_LZMA, UPX | ![ByteView](1) | ![IATView](1) | ![BehaviorView](1) |
| 3. 0a45ee3c3d12af3571d80bd4cd60f37801b416044e411786aa5ab343836702e1  
  Packer: UPX_LZMA, UPX | ![ByteView](1) | ![IATView](1) | ![BehaviorView](1) |
| 4. 01d09cd684edfae388e23743f98e27881d2b501abf7413f1e0351db5ab7b8add  
  Packer: UPX | ![ByteView](1) | ![IATView](1) | ![BehaviorView](1) |
| 5. 098649ebd5451c6bee0821a5a42837263feae15035d94213f338a5db141ec6c6  
  Packer: UPX | ![ByteView](1) | ![IATView](1) | ![BehaviorView](1) |
Ending Notes / Q&A

-- -- --

- https://github.com/7h3rAm/angad/

- @7h3rAm

- Thanks:
  - Aldo Cortesi: @cortesi
  - Daniel Plohmann: @push_pnx
  - xkcd :)

---