Malware mining

Dr. Igor Muttik – McAfee Labs™
• What makes a great AV product
• What is data-mining?
• Data-mining methods
  – Decision trees
  – Support Vector Machines
  – ROCs
• Data-mining as a heuristic/generic method
• Extracting features
• Practical uses of malware data-mining
• Conclusions and questions
What would make a great AV product?

• Proactively detects as many threats as possible
• Creates a very low number of false alarms
• Requires as little maintenance as possible
• Runs quickly and introduces little overhead
But how can this be done?

- Traditional (specific) detection methods (strings, hashes)
- Generic detection methods
- Heuristic detection methods
- Cloud-based methods
- Reputation methods
- Behavioural methods

<table>
<thead>
<tr>
<th>Detection method</th>
<th>Proactive capability</th>
<th>Cost to update</th>
<th>Available for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific (manual)</td>
<td>None</td>
<td>Medium</td>
<td>Many years</td>
</tr>
<tr>
<td>Specific (automated)</td>
<td>None</td>
<td>Low</td>
<td>Many years</td>
</tr>
<tr>
<td>Generic (manual)</td>
<td>High</td>
<td>High</td>
<td>Many years</td>
</tr>
<tr>
<td>Generic (automated)</td>
<td>Medium</td>
<td>Medium</td>
<td>Many years</td>
</tr>
<tr>
<td>Heuristic (manual)</td>
<td>Medium</td>
<td>High</td>
<td>Many years</td>
</tr>
<tr>
<td>Heuristic (automated)</td>
<td>Medium</td>
<td>Low</td>
<td>???</td>
</tr>
</tbody>
</table>
What is data-mining?
Data Mining—An Example

Measurements
Height Versus Weight

<table>
<thead>
<tr>
<th>Height (in inches)</th>
<th>Weight (in pounds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>65</td>
<td>150</td>
</tr>
<tr>
<td>70</td>
<td>200</td>
</tr>
<tr>
<td>75</td>
<td>250</td>
</tr>
<tr>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>
Height Versus Weight + 1 Boolean feature

Weight (in pounds)

60 65 70 75 80

Women

Men
One Dimension Only (1 feature)
Putting Weight and Height Into Perspective
Best Guess for Gender

- 100% male
- 0% female
- 50% male
- 50% female
- 0% male
- 100% female

Height (in inches)
Weight (in pounds)

Best Guess
Buttock Circumference: The circumference of the body measured at the level of the maximum posterior protuberance of the buttocks.
Buttock Circumference: The circumference of the body measured at the level of the maximum posterior protuberance of the buttocks.
Further Improving the Separation

**Signal to Noise**
Features with very different distribution per class

**Correlation**
Features with low correlation

**Dimensionality**
Consider more features at the same time
Data-mining methods
Classification Algorithms

- Decision Trees
- Decision Forests
- Neural Networks
- Support Vector Machines

Final Verdict
ROCs graphically represent the quality of separation achieved by data-mining.
Primitive decision tree

Is the file packed?

- Yes: Emulate to unpack
- No: Scan without emulation
Manually-constructed decision tree (logic)

Is the file packed?

Yes:
- Does the file have a trusted digital signature?
  - No: Emulate to unpack
  - Yes: Do not scan

No: Scan without emulation
Feature extraction for PE files

- Is file packed = Boolean
- Is file a DLL = Boolean
- HLL language = Integer {1=VB, 2=MSVC, 3=Delphi, 4=.NET, 5=…}
- How many PE sections = Integer
- Are section names standard = Boolean
- Are characteristics of sections standard = Boolean
- Is entry point in the 1st section = Boolean
- Is file digitally signed = Boolean
- Is DOS header standard = Boolean
- Is PE timestamp old = Boolean
- How many resources = Integer
- How many languages in resources = Integer
- Do timestamps in PE header/resources/digisig match = Boolean
- Number of imports = Integer
- Number of exports = Integer
- Number of bound imports = Integer
- Number of imports from WSOCK32.DLL
Steps to build a decision tree

Build collections
- Build two TP and FP sets (training sets)
- Setting aside collections for testing (e.g. 10-25%)

Extract features
- From TP set
- From FP set

Model
- Can be expensive for large collections and lots of features
- Pruning (if there is overfitting)

Test
- If the result is not good – make changes and repeat

Convert into usable form
- Decision tree logic into C or C++
ROC for a decision trees we use

- Decision tree (DT) for PE files with 20 features

When built

In 6 months
Converting logic into code

- DT visualized in Weka

- DT as C code

```c
if (LSEX <= 0) {
    if (PACK <= 0) {
        if (TIME <= 1174454487) {
            if (TIME <= 708992669) {
                if (TIME <= 708992669) {
                    if (DLLB <= 0) {
                        if (SIZE <= 1043955) {
                            if (CONS <= 0) {
                                if (RESD <= 40) {confidence=9892; }
                            } else {
                                if (URLD <= 0) {
                                    if (WNET <= 0) {
                                        if (OVLY <= 905728) {
                                            if (SECS <= 8) {
                                                if (WSCK <= 0) {
                                                    if (IMSZ <= 212) {
                                                        if (OVLY <= 399050) {
                                                            if (SIZE <= 66240) {confidence=-5500;}
                                                        } else { confidence=9255;}
                                                    ...
```
Practical uses of malware data-mining
Practical uses

- To prioritize and de-prioritize samples in research queues
- To drive the depth of the sample analysis on the endpoint, gateway or a server
- To check the most suspicious samples using cloud-based security
- To exclude less suspicious samples from cloud communication
- Applies to anti-malware, anti-spam
- It is an automatic heuristic!
Conclusions

• Data-mining is very useful in malware analysis

• Easy to automate

• Can be used for both “strong” and “weak” heuristic verdicts
  – Detections
  – Limiting
  – Prioritization