Malware Removal – Beyond Content and Context Scanning

Tom Brosch, Maik Morgenstern
AV-Test GmbH, Magdeburg, Germany

http://www.av-test.org

Presented at the Virus Bulletin 2007 Conference in Vienna, Austria
Table of Contents

• Malware Removal
  – Why?
  – Content and Context scanning
  – Test Results
• Generic Malware Removal
  – Overview
  – Sandbox based Removal
  – Problems and solutions
• Further Concepts
• Conclusion
• Q&A
Malware Removal

• Why...
  – … is Malware Removal necessary?
  – … is comprehensive Malware Removal necessary?
  – … is Malware Removal a lot of work and a problem?
Why is Malware Removal necessary?

• Systems still get infected for different reasons
  – Users install an Anti-Malware software when it is too late …
  – Users update an Anti-Malware software when it is too late …
  – Anti-Malware vendors react when it is too late …
• These systems have to be cleaned
Why is Comprehensive Malware Removal Necessary?

• Comprehensive Malware Removal?
  – Malicious processes should be terminated and the related executables be removed
  – What about Run keys in the Registry?
  – What about settings changed by the malware?
  – What about other components, like image files or configuration files used by the malware?

• Why care?
  – Because the user cares
  – They are looking for a “really clean” system, since that’s what they pay for
  – Risk of reinfection when missing components or system changes
  – Another security product might “detect” the leftover components and leave the user in an uncertain state
  – Rogue Anti-Spyware products are producing false positives at the moment, they might happily switch to the leftover components
Why is Malware Removal a Lot of Work and a Problem?

- To have proper removal routines in place, a lot of analysis work by the Anti-Malware vendor is required
  - Different behavior of malware on different systems
  - Behavior of malware may change over time (downloaded components)
  - Threats are way more complex today
- The increasing amount of malware is not going to make it better
- Bad removal routines indicate a bad analysis, which doesn’t increase the trust of users in the software
Content and Context Scanning

• Content scanning uses signatures to identify malicious components
• Context scanning uses context rules to identify linked malicious components
• A combination of both is required to cope with today’s complex threats
• Both approaches require an analysis
• Several issues have to be considered: random file names, rootkits, anti-removal techniques, shared components, pre-infection settings, changing behavior of malware
• Simple fixes and workarounds are available for most problems
Content and Context Scanning

- Response times

Malware appears at users PC
AV company receives malware
Proactive detection
Signature based detection
Removal routine available
Malware Removal – Test Results

- Response Times and proactive detection

<table>
<thead>
<tr>
<th>Vendor / Product</th>
<th>Average response time range, including proactive detections</th>
<th>Proactive detection (based on different tests)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avira AntiVir</td>
<td>2 to 4 hours</td>
<td>20 to 50%</td>
</tr>
<tr>
<td>Alwil Avast</td>
<td>6 to 8 hours</td>
<td>5 to 35%</td>
</tr>
<tr>
<td>Grisoft AVG</td>
<td>6 to 8 hours</td>
<td>5 to 35%</td>
</tr>
<tr>
<td>BitDefender</td>
<td>2 to 4 hours</td>
<td>25 to 60%</td>
</tr>
<tr>
<td>F-Secure</td>
<td>less than 2 hours</td>
<td>20 to 50%</td>
</tr>
<tr>
<td>Kaspersky</td>
<td>less than 2 hours</td>
<td>20 to 50%</td>
</tr>
<tr>
<td>McAfee</td>
<td>14 to 16 hours</td>
<td>25 to 45%</td>
</tr>
<tr>
<td>Microsoft</td>
<td>38 to 40 hours</td>
<td>5 to 15%</td>
</tr>
<tr>
<td>Eset Nod32</td>
<td>4 to 6 hours</td>
<td>30 to 70%</td>
</tr>
<tr>
<td>Panda</td>
<td>4 to 6 hours</td>
<td>20 to 50%</td>
</tr>
<tr>
<td>Symantec Norton</td>
<td>6 to 8 hours</td>
<td>15 to 50%</td>
</tr>
<tr>
<td>Trend Micro</td>
<td>6 to 8 hours</td>
<td>15 to 45%</td>
</tr>
</tbody>
</table>
Malware Removal – Test Results

• Detection of dropped components
• Ad- and Spyware vs. WildList Malware
Malware Removal – Test Results

• Removal Results

<table>
<thead>
<tr>
<th></th>
<th>Files created</th>
<th>Registry keys created</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdWare.Hotbar</td>
<td>183</td>
<td>789</td>
</tr>
<tr>
<td><strong>Files removed</strong></td>
<td><strong>Registry keys removed</strong></td>
<td></td>
</tr>
<tr>
<td>Product A (AV)</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>Product B (AV)</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>Product C (AV)</td>
<td>26</td>
<td>43</td>
</tr>
<tr>
<td>Product D (AS)</td>
<td>182</td>
<td>778</td>
</tr>
</tbody>
</table>
Malware Removal – Conclusion

• Certain threats are handled very well (e.g. WildList malware)
• Other threat categories could need some more attention
• Proactive detection is far from 100%
• Response times still go up to several days
• It takes some time until removal routines are in place and no product is 100% perfect
Generic Malware Removal

• Overview
  – What is needed?
    • Reduce the response time where no sufficient disinfection routine is available
    • Disinfection without a dedicated analysis done by the vendors
  – Alternative times for an analysis
    • When the malware is first run on the users pc
    • When the malware is detected by the antivirus product
Generic Malware Removal

• Overview
  – How to analyze on detection time?
    • Run the malware again and monitor the changes it makes to the system
    • Run the malware without damaging the system
  - Sandbox-based disinfection
    - What is it?
    - How does it work?
    - How well does it work and what are the problems?
Generic Malware Removal

• Sandbox based Removal
  – The Idea
    • Emulate the malware and report all system changes
    • Perform a removal based on this report
  – What is a sandbox?
    • Virtual environment separated from the system
    • Executable files can be testdriven to analyse their behaviour
Generic Malware Removal

- Sandbox based Removal
  - Single-stage approach

![Diagram of malware removal process]
Generic Malware Removal

• Sandbox based Removal
  – Test results
    • Comparison of manual analysis with the sandbox analysis
    • Only few files and registry entries found
      – Example: Admedia
      – 24 of 48 files found
      – 6 of 178 Registry entries found
    ⇒ Single-stage approach not suitable for real malware
Generic Malware Removal

- Sandbox based Removal
  - Multi-stage approach
Generic Malware Removal

• Sandbox based Removal
  – Test results
    • More files and registry entries found
      – Example: Win32/Admedia
        – Increase from 24 to 32 of 48 files found
        – Increase from 6 to 10 of 178 Registry entries found
    • Multi-stage approach better but far from good
Generic Malware Removal

• Problems and solutions
  – Related to the sandbox (same for many malware samples)
    • The native API
    • Different behaviour in virtual environments
  – Related to the malware (can not be solved by improvements of the sandbox)
    • User interaction
    • Downloaded files from the internet during infection
    • Scheduled tasks, infection after reboot, etc.
Generic Malware Removal

• Problems and solutions
  – Related to the malware
    • Different behavior on an infected system
    • Random filenames
    • Pre-infection settings
  – Some worst case scenario
    • Inactive sample triggers the removal routine
    • The malware breaks out of the sandbox (exploiting some vulnerability) during emulation
    • Infection instead of disinfection
Further Concepts – Supervision

• Log the system changes done by a certain application
• As soon as it is known that this application is malicious, all the changes can easily be reverted
• Solves the problem of pre-infection settings or different behavior in sandbox and real pc
• There are other problems coming up:
  – Which applications should be supervised?
  – Which system changes are malicious and should be reverted?
  – Applications might evade the supervision
Further Concepts – Supervision

• Similar concepts are already used in current software:
  – Guards which monitor system areas and block all changes or ask the user whether to allow or block
  – Behavior based detection/prevention/blocking, which is a far better approach, because it takes the whole behavior and not only single actions into account and can, in the best case, decide by itself
## Conclusion

<table>
<thead>
<tr>
<th></th>
<th>Content and Context Scanning (Manual Analysis)</th>
<th>Sandbox based approach</th>
<th>Supervision approach</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Availability</strong></td>
<td>- (minutes to days)</td>
<td>+ (instantly)</td>
<td>+ (instantly)</td>
</tr>
<tr>
<td><strong>(Response time)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Different behavior in sandboxes</strong></td>
<td>+ (depends on the quality of the analysis)</td>
<td>- (obviously a problem)</td>
<td>+ (no problem)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Performance impact</strong></td>
<td>+ (none)</td>
<td>+ (nearly none)</td>
<td>- (rather much)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Handling of pre-infection settings</strong></td>
<td>- (resetting default values in the best case)</td>
<td>- (resetting default values in the best case)</td>
<td>+ (no problem)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Decision whether changes are malicious or not</strong></td>
<td>+ (depends on the quality of the analysis)</td>
<td>- (hard to do)</td>
<td>- (hard to do)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Catch all (relevant) changes</strong></td>
<td>+ (depends on the quality of the analysis)</td>
<td>- (problematic, as seen)</td>
<td>- (can be a problem)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions & Answers

• ???

• Note: Many testing papers can be found at:
http://www.av-test.org → Publications → Papers