ANUBIS

ANalyzing Unknown BINaries

The automatic Way

Thomas Mandl, Ulrich Bayer, Florian Nentwich

25.09.2009, v1.0.02, EN

Virus Bulletin Conference 2009, Geneva
People behind ANUBIS – Who are we?

Ulrich Bayer
- Currently PhD student at Vienna University of Technology
- Main developer and architect of ANUBIS

Florian Nentwich
- Senior malware analyst at Ikarus labs and maintainer of commercial ANUBIS version

Thomas Mandl
- Former CTO of Ikarus, now CEO of his own information security consulting company in Austria
- Still contributing to the ANUBIS project
ANUBIS’ Academic Research Members

Engin Kirda
- Assistant professor at EURECOM Communication Systems
- Former assistant professor at Vienna University of Technology
- http://www.eurecom.fr/people/kirda.en.htm

Christopher Kruegel
- Assistant professor at UCSB, Dept. of Computer Science
- Former assistant professor at Vienna University of Technology
- http://www.cs.ucsb.edu/~chris/
- Development of Wepawet Tools

See also
- http://www.iseclab.org/people.html
Automated Malware Analysis: Why?

Too many new malware samples per day

- ~25k - 35k samples per day (unique MD5) (peak up to 50k)
- Increasing number of malware uses runtime packers/code obfuscation methods to trick pattern matching AV
- Increasing FP rate, nobody can handle this load manually!
- Almost no in-house incident response process/RE due to its complexity (at least in Austria)
- Among others, this was our primary motivation to create ANUBIS!
A traditional Analysis Approach

- ~35k samples/day
- Manual analysis takes up to several days

- Limited human expert resources
- Experts should concentrate on novel malware

- Response time for signature creation is crucial
- How can we speed up this process?
What is ANUBIS?

Framework of several tools for **dynamic code analysis**
- We run a binary in an emulated PC environment (WinXP/SP3)
- We monitor its actions (SysCalls, Windows API functions, ...)
- We generate a detailed report of the sample’s behavior
- Fully automatically within 4 min. (**no human interaction**)
- Based on an ANUBIS report, a human expert can decide whether to manually analyze a sample in depth or not.

**Benefits of dynamic code analysis with ANUBIS**
- Scalable approach, unaffected by runtime packers, code obfuscation or anti-debug mechanisms of modern malware
- Can handle basic user interactions if required during analysis

**Community version heavily used by AV and AV researchers**
- [https://anubis.iseclab.org](https://anubis.iseclab.org) (public) with limited features
Architecture and Capabilities

ANUBIS has 5 primary building blocks

- **Web/DB server/HTTP(s) frontend (upload/admin)**
  - DB stores reports and references to samples (XML)
  - Enables us to generate *lots of statistics*!

- **Malware sample storage**
  - Archives uploaded and already analyzed samples

- **Report storage**
  - Archives report/result files (traffic dumps, downloaded files...)
  - Comprehensive Archive + 2nd stage malware!

- **Victim server**
  - Acts as *local honey pot* for certain services and keeps malicious traffic local!

- **Multiple Worker (VM)**
  - Snapshot technology! Revert to known state in a second!
ANUBIS Architecture

Sample Upload via HTTP

Web / DB SRV

Worker VMs

Victim SRV

ANUBIS FRAMEWORK

Report Store

Sample Store
Advanced Features of ANUBIS

Records and analyzes sample’s network traffic
- HTTP, FTP, SMTP, IRC, ... are available as PCAP file

Storage of analysis reports in relational DB
- Servers contacted, files created, modified, deleted, RegKeys manipulated, and short threat summary

Several report formats
- XML, HTML, MHT, PDF, TXT
- Integrates also static analysis with AV scanner/PE scan

URL analysis (early development stage)

ANUBIS was designed to support human experts
- Gives quick overview of a sample’s behavior within minutes
- What makes ANUBIS different from other sandbox solutions?
Anubis - Analysis Report

Analysis Report for nepenthes-65c242c013045c678974e3be0796188d-index.html

Summary:

<table>
<thead>
<tr>
<th>Description</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creates files in the Windows system directory: Malware often keeps copies of itself in the Windows directory to stay undetected by users.</td>
<td>📦</td>
</tr>
<tr>
<td>Performs Address Scan: The executable scans a range of IP Addresses. In most cases these scans identify more potential vulnerable targets.</td>
<td>🛡️</td>
</tr>
<tr>
<td>Performs File Modification and Destruction: The executable modifies and destructs files which are not temporary.</td>
<td>🚧</td>
</tr>
<tr>
<td>Spawns Processes: The executable produces processes during the execution.</td>
<td>🐱</td>
</tr>
<tr>
<td>Performs Registry Activities: The executable reads and modifies registry values. It may also create and monitor registry keys.</td>
<td>🍾</td>
</tr>
</tbody>
</table>

Table of Contents

- General information
  - nepenthes-65c242c013045c678974e3be0796188d-index.html
    - urdvxc.exe
    - urdvxc.exe
    - services.exe
    - urdvxc.exe
    - urdvxc.exe
## Report – Static findings

<table>
<thead>
<tr>
<th>SHA-1:</th>
<th>b616dcf0c05e539b317edd9d279a267a6fad01e</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Size:</td>
<td>131584 Bytes</td>
</tr>
<tr>
<td>Command Line:</td>
<td>&quot;C:\nepenthes-65c242c013045c678974e3be0796188d-index.html&quot;</td>
</tr>
<tr>
<td>Process-status at analysis end:</td>
<td>dead</td>
</tr>
<tr>
<td>Exit Code:</td>
<td>0</td>
</tr>
</tbody>
</table>

+ **Load-time Dlls**

+ **Run-time Dlls**

- **SigBuster Output**
  - Allaple_Polymorphic_Packer vna SN: 1647

- **Ikarus Virus Scanner**
  - Net-Worm.Win32.Allaple.b (Sig-Id:158175)
### 3.c) urdvxc.exe - Windows Service Activities

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSWindows</td>
<td>SERVICE_AUTO_START</td>
<td>&quot;C:\WINDOWS\system32\urdvxc.exe&quot; /service</td>
</tr>
</tbody>
</table>

**- Services Changed:**

- MSWindows
- MSWindows
7.c) urdvxc.exe - Network Activity

- **ICMP Traffic:**
  - ICMP Echo Requests sent to 26 hosts
  - ICMP Echo Replies received from 26 hosts
  - Scanned a Subnet: 61.229.0.0/16

- **Unknown TCP Traffic:**
  **from ANUBIS:1328 to 61.229.113.109:445**
  - State: Connection established, not terminated
  - Transferred outbound Bytes: 172
  - Transferred inbound Bytes: 0
  - Data sent:
    
    0000 00a8 ff53 4d42 7200 0000 0008 0140  ......SMBr......@

- **TCP Connection Attempts:**
  **from ANUBIS:1040 to 61.229.113.109:139**
  **from ANUBIS:1039 to 61.229.82.160:139**
  **from ANUBIS:1038 to 61.229.54.57:139**
  **from ANUBIS:1041 to 61.229.118.248:139**
  **from ANUBIS:1042 to 61.229.218.221:139**
Data Tainting in Anubis

Powerful technique for tracing data flows of a program

- E.g. how network data is processed by a program
- E.g. it enables us to find out if malware uses random file names for infection only during one single analysis run

How does tainting work?

- Performed on hardware level, invisible for analyzed malware
- Data elements of interest are labeled (tainted)
- When memory values are copied, taint labels (information) are maintained allowing us to identify the data flow process
Memory Tainting Example

Consider the following code fragment

ticks = GetTickCount()
filename = "c:\" + ticks + ".exe"
file = CreateFile(filename, ...)

Enhanced with tainting information

ticks = GetTickCount()

ticks <GetTickCount>
filename = "c:\" + ticks + ".exe"

filename <GetTickCount>
file = CreateFile(filename, ...)

=> CreateFile is called with a random filename
Resume so far

By now we have achieved the following

- We can automatically analyze single malware samples
- We known within 4 min. if this sample is malicious or not
- We can provide a non-obtrusive view from outside on our malware’s behavior

But we still have the following challenges

- How to structure thousands of generated analysis reports?
- Wouldn’t it be nice to know (for every new incoming sample) if it belongs to a well-known malware family?

ANUBIS can also provide this additional information

- This feature is called “clustering”
Scalable, Behavior-Based Malware Clustering

Malware Clustering: Find a partitioning of a given set of malware samples into subsets so that subsets share some common traits (i.e., find “virus families”)

![Diagram showing malware samples clustered into subsets: Blaster, Fishing, Slammer]
Malware Clustering – Features

Behavior-based
- Samples are clustered according to their behavior exhibited at runtime
- Requires prior analysis by Anubis

Scalable
- Use of LSH (Locality Sensitive Hashing) allows us to avoid computing all $n^2/2$ distances
- Suitable for clustering real-world malware collections

Details
- Ulrich Bayer, Paolo Milani, Clemens Hlauschek, Christopher Kruegel, and Engin Kirda: Scalable, Behavior-Based Malware Clustering, NDSS 2009, San Diego, February 2009
How about clustering 825k samples...

...in less than 8 hours?

- Most recent clustering run (August 16th 2009):
- [http://anubis.iseclab.org/?action=browse_clusters&task=299](http://anubis.iseclab.org/?action=browse_clusters&task=299)
Anubis Clustering Task 299

Cluster Task Id: 299
Create Time: 2009-08-16 10:11:24
Start Time: 2009-08-16 10:15:38
End Time: 2009-08-16 18:09:23
Run Time: 07:53:45
Peak Virtual Memory Size: 21.53 Gb
Peak Resident Set Size: 18.74 Gb

Samples were submitted between: 2007-02-07 13:44:00 - 2009-08-16 16:58:53

Anubis Families 1 - 10 of 91521

<table>
<thead>
<tr>
<th>Family Id</th>
<th># Samples</th>
<th>Top 3 A/V Labels</th>
<th>% Samples</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>6824791</td>
<td>116995</td>
<td>53% -unlabeled-</td>
<td>2%</td>
<td>trojan-dropper.agent</td>
</tr>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allaple</td>
<td>25%</td>
<td>-unlabeled-</td>
</tr>
<tr>
<td>6823539</td>
<td>49148</td>
<td>61% -unlabeled-</td>
<td>14%</td>
<td>trojan.generic</td>
</tr>
<tr>
<td>6789759</td>
<td>31984</td>
<td>50% -unlabeled-</td>
<td>10%</td>
<td>trojan-clicker.html iframe</td>
</tr>
<tr>
<td>6798627</td>
<td>25494</td>
<td>47% -unlabeled-</td>
<td>9%</td>
<td>trojan-dropper.agent</td>
</tr>
<tr>
<td>6830115</td>
<td>24639</td>
<td>82% net-worm.win32.allaple</td>
<td>17%</td>
<td>-unlabeled-</td>
</tr>
<tr>
<td>6830127</td>
<td>17440</td>
<td>98% -unlabeled-</td>
<td>0%</td>
<td>trojan-downloader.win32.autoit</td>
</tr>
<tr>
<td>6797651</td>
<td>16042</td>
<td>58% backdoor.win32</td>
<td>40%</td>
<td>-unlabeled-</td>
</tr>
<tr>
<td>6818183</td>
<td>15483</td>
<td>69% -unlabeled-</td>
<td>25%</td>
<td>trojan-downloader.win32</td>
</tr>
<tr>
<td>6788535</td>
<td>14495</td>
<td>91% net-worm.win32.allaple</td>
<td>9%</td>
<td>-unlabeled-</td>
</tr>
</tbody>
</table>

Number of Samples: 827377
Anubis Tasks: 998505
Unique Behavioral Profiles: 730539
Number of Clusters: 91521

Local Sensitive Hashing Parameters
Distance Threshold = 0.2
l = 87
k = 20
Anubis Clustering Task 299

Cluster Task Id: 299
Create Time: 2009-08-16 10:11:24
Start Time: 2009-08-16 10:15:38
End Time: 2009-08-16 18:09:23
Run Time: 07:53:45
Peak Virtual Memory Size: 21.53 Gb
Peak Resident Set Size: 18.74 Gb
Samples were submitted between: 2007-02-07 13:44:00 - 2009-08-16 16:58:53

Number of Samples: 827377
Anubis Tasks: 998505
Unique Behavioral Profiles: 730539
Number of Clusters: 91521

Local Sensitive Hashing Parameters
Distance Threshold = 0.2
l = 87
k = 20

Anubis Families 1 - 10 of 91521

<table>
<thead>
<tr>
<th>Family Id</th>
<th># Samples</th>
<th>Top 3 A/V Labels</th>
<th>next~</th>
</tr>
</thead>
<tbody>
<tr>
<td>6824791</td>
<td>116995</td>
<td>53% -unlabeled-</td>
<td>2% trojan-dropper.agent 2% virus.win32.virut</td>
</tr>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allaple</td>
<td>25% -unlabeled- 0% virus.win32.cheburgen</td>
</tr>
<tr>
<td>6823539</td>
<td>49148</td>
<td>61% -unlabeled-</td>
<td>14% trojan.generic 9% not-a-virus:webdriver</td>
</tr>
<tr>
<td>6789759</td>
<td>31984</td>
<td>50% -unlabeled-</td>
<td>10% trojan-clicker.html iframe 8% trojan-clicker.js.agent</td>
</tr>
<tr>
<td>6798627</td>
<td>25494</td>
<td>47% -unlabeled-</td>
<td>9% trojan-dropper.agent 7% virus.worm.win32.socks</td>
</tr>
<tr>
<td>6830115</td>
<td>24639</td>
<td>82% net-worm.win32.allaple</td>
<td>17% -unlabeled- 0% backdoor.rbot</td>
</tr>
<tr>
<td>6830127</td>
<td>17440</td>
<td>98% -unlabeled-</td>
<td>0% trojan-downloader.win32.autoit 0% packed.win32.klone</td>
</tr>
<tr>
<td>6797651</td>
<td>16042</td>
<td>58% backdoor.win32</td>
<td>40% -unlabeled-</td>
</tr>
<tr>
<td>6918163</td>
<td>15483</td>
<td>69% -unlabeled-</td>
<td>25% trojan-downloader.win32</td>
</tr>
<tr>
<td>6788535</td>
<td>14495</td>
<td>91% net-worm.win32.allaple</td>
<td>9% -unlabeled- 0% virus.win32.cheburgen</td>
</tr>
</tbody>
</table>
Anubis Clustering Task 299

Cluster Task Id: 299
Create Time: 2009-08-16 10:11:24
Start Time: 2009-08-16 10:15:38
End Time: 2009-08-16 18:09:23
Run Time: 07:53:45
Peak Virtual Memory Size: 21.53 Gb
Peak Resident Set Size: 18.74 Gb
Samples were submitted between: 2007-02-07 13:44:00 - 2009-08-16 16:58:53

Number of Samples: 827377
Anubis Tasks: 998505
Unique Behavioral Profiles: 730539
Number of Clusters: 91521

Local Sensitive Hashing Parameters
Distance Threshold = 0.2
l = 87
k = 20

Anubis Families 1 - 10 of 91521

<table>
<thead>
<tr>
<th>Family Id</th>
<th># Samples</th>
<th>Top 3 A/V Labels</th>
<th>next...</th>
</tr>
</thead>
<tbody>
<tr>
<td>6824791</td>
<td>116995</td>
<td>53% -unlabeled-</td>
<td>2% trojan-dropper.agent</td>
</tr>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allaple</td>
<td>25% -unlabeled-</td>
</tr>
<tr>
<td>6823539</td>
<td>49148</td>
<td>61% -unlabeled-</td>
<td>14% trojan.generic</td>
</tr>
<tr>
<td>6789759</td>
<td>31984</td>
<td>50% -unlabeled-</td>
<td>10% trojan-clicker.html.iframe</td>
</tr>
<tr>
<td>6798627</td>
<td>25494</td>
<td>47% -unlabeled-</td>
<td>9% trojan-dropper.agent</td>
</tr>
<tr>
<td>6830115</td>
<td>24639</td>
<td>82% net-worm.win32.allaple</td>
<td>17% -unlabeled-</td>
</tr>
<tr>
<td>6830127</td>
<td>17440</td>
<td>98% -unlabeled-</td>
<td>0% trojan-downloader.win32.autoit</td>
</tr>
<tr>
<td>6797651</td>
<td>16042</td>
<td>58% backdoor.win32</td>
<td>40% -unlabeled-</td>
</tr>
<tr>
<td>6918183</td>
<td>15483</td>
<td>69% -unlabeled-</td>
<td>25% trojan-downloader.win32</td>
</tr>
<tr>
<td>6788535</td>
<td>14495</td>
<td>91% net-worm.win32.allaple</td>
<td>9% -unlabeled-</td>
</tr>
</tbody>
</table>
Anubis Clustering Task 299

Cluster Task Id: 299
Create Time: 2009-08-16 10:11:24
Start Time: 2009-08-16 10:15:38
End Time: 2009-08-16 18:09:23
Run Time: 07:53:45
Peak Virtual Memory Size: 21.53 Gb
Peak Resident Set Size: 18.74 Gb
Samples were submitted between: 2007-02-07 13:44:00 - 2009-08-16 16:58:53

Anubis Families 1 - 10 of 91521

<table>
<thead>
<tr>
<th>Family Id</th>
<th># Samples</th>
<th>Top 3 A/V Labels</th>
<th>next...</th>
</tr>
</thead>
<tbody>
<tr>
<td>6824791</td>
<td>116995</td>
<td>2% trojan-dropper.agent, 25% -unlabeled, 9% trojan-dropper.agent</td>
<td></td>
</tr>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allaple, 61% -unlabeled, 14% trojan.generic, 50% -unlabeled</td>
<td>9% not-a-virus:.webtoolbars</td>
</tr>
<tr>
<td>6823539</td>
<td>49148</td>
<td>31984, 61% -unlabeled, 61% -unlabeled, 47% -unlabeled</td>
<td>9% trojan-dropper.ag</td>
</tr>
<tr>
<td>6789759</td>
<td>25494</td>
<td>24639, 50% -unlabeled, 47% -unlabeled, 47% -unlabeled</td>
<td>9% trojan-dropper.ag</td>
</tr>
<tr>
<td>6830115</td>
<td>24639</td>
<td>82% net-worm.win32.allaple, 82% -unlabeled, 74% -unlabeled</td>
<td>9% trojan-dropper.ag</td>
</tr>
<tr>
<td>6830127</td>
<td>17440</td>
<td>100% -unlabeled, 98% -unlabeled, 98% -unlabeled</td>
<td>9% trojan-dropper.ag</td>
</tr>
<tr>
<td>67797651</td>
<td>16042</td>
<td>58% backdoor.win32, 40% -unlabeled, 40% -unlabeled</td>
<td>9% trojan-dropper.ag</td>
</tr>
<tr>
<td>6818183</td>
<td>15483</td>
<td>69% -unlabeled, 69% -unlabeled, 69% -unlabeled</td>
<td>9% trojan-dropper.ag</td>
</tr>
<tr>
<td>6788535</td>
<td>14495</td>
<td>91% net-worm.win32.allaple, 91% -unlabeled, 91% -unlabeled</td>
<td>9% trojan-dropper.ag</td>
</tr>
</tbody>
</table>

Local Sensitive Hashing Parameters
Distance Threshold = 0.2
I = 87
k = 20
Anubis Clustering Task 299

Cluster Task Id: 299
Create Time: 2009-08-16 10:11:24
Start Time: 2009-08-16 10:15:38
End Time: 2009-08-16 18:09:23
Run Time: 07:53:45
Peak Virtual Memory Size: 21.53 Gb
Peak Resident Set Size: 18.74 Gb
Samples were submitted between: 2007-02-07 13:44:00 - 2009-08-16 16:58:53

Anubis Families 1 - 10 of 91521

<table>
<thead>
<tr>
<th>Family Id</th>
<th># Samples</th>
<th>Top 3 A/V Labels</th>
<th>next</th>
</tr>
</thead>
<tbody>
<tr>
<td>6824791</td>
<td>116995</td>
<td>53% -unlabeled-</td>
<td>2% trojan-dropper.agent</td>
</tr>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allaple</td>
<td>25% -unlabeled-</td>
</tr>
<tr>
<td>6823539</td>
<td>49148</td>
<td>61% -unlabeled-</td>
<td>14% trojan.generic</td>
</tr>
<tr>
<td>6789759</td>
<td>31984</td>
<td>50% -unlabeled-</td>
<td>10% trojan-clicker.html iframe</td>
</tr>
<tr>
<td>6798627</td>
<td>25494</td>
<td>47% -unlabeled-</td>
<td>9% trojan-dropper.agent</td>
</tr>
<tr>
<td>6830115</td>
<td>24639</td>
<td>82% net-worm.win32.allaple</td>
<td>17% -unlabeled-</td>
</tr>
<tr>
<td>6830127</td>
<td>17440</td>
<td>98% -unlabeled-</td>
<td>0% trojan-downloader.win32.autoit</td>
</tr>
<tr>
<td>6797651</td>
<td>16042</td>
<td>58% backdoor.win32</td>
<td>40% -unlabeled-</td>
</tr>
<tr>
<td>6818183</td>
<td>15483</td>
<td>69% -unlabeled-</td>
<td>25% trojan-downloader.win32</td>
</tr>
<tr>
<td>6788535</td>
<td>14495</td>
<td>91% net-worm.win32.allaple</td>
<td>9% -unlabeled-</td>
</tr>
</tbody>
</table>

IKARUS security software
## Anubis Clustering Task 299

<table>
<thead>
<tr>
<th>Cluster Task Id:</th>
<th>299</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Time:</td>
<td>2009-08-16 10:11:24</td>
</tr>
<tr>
<td>Start Time:</td>
<td>2009-08-16 10:15:38</td>
</tr>
<tr>
<td>End Time:</td>
<td>2009-08-16 18:09:23</td>
</tr>
<tr>
<td>Run Time:</td>
<td>07:53:45</td>
</tr>
<tr>
<td>Peak Virtual Memory Size:</td>
<td>21.53 Gb</td>
</tr>
<tr>
<td>Peak Resident Set Size:</td>
<td>18.74 Gb</td>
</tr>
<tr>
<td>Samples were submitted between:</td>
<td>2007-02-07 13:44:00 - 2009-08-16 16:58:53</td>
</tr>
</tbody>
</table>

### Anubis Families 1 - 10 of 91521

<table>
<thead>
<tr>
<th>Family Id</th>
<th># Samples</th>
<th>Top 3 A/V Labels</th>
<th>2% trojan-dropper.agent</th>
<th>2% virus.win32.virut</th>
</tr>
</thead>
<tbody>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allape 25% -unlabeled-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allape 25% -unlabeled-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allape 25% -unlabeled-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allape 25% -unlabeled-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allape 25% -unlabeled-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Anubis Clustering Task 299

<table>
<thead>
<tr>
<th>Cluster Task Id:</th>
<th>299</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Time:</td>
<td>2009-08-16 10:11:24</td>
</tr>
<tr>
<td>Start Time:</td>
<td>2009-08-16 10:15:38</td>
</tr>
<tr>
<td>End Time:</td>
<td>2009-08-16 18:09:23</td>
</tr>
<tr>
<td>Run Time:</td>
<td>07:53:45</td>
</tr>
<tr>
<td>Peak Virtual Memory Size:</td>
<td>21.53 Gb</td>
</tr>
<tr>
<td>Peak Resident Set Size:</td>
<td>18.74 Gb</td>
</tr>
<tr>
<td>Samples were submitted between:</td>
<td>2007-02-07 13:44:00 - 2009-08-16 16:58:53</td>
</tr>
</tbody>
</table>

#### Local Sensitive Hashing Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance Threshold</td>
<td>0.2</td>
</tr>
<tr>
<td>l</td>
<td>87</td>
</tr>
<tr>
<td>k</td>
<td>20</td>
</tr>
</tbody>
</table>

### Anubis Families 1 - 10 of 91521

<table>
<thead>
<tr>
<th>Family Id</th>
<th># Samples</th>
<th>Top 3 A/V Labels</th>
<th>next&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>6824791</td>
<td>116995</td>
<td>53% -unlabeled-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2% trojan-dropper.agent</td>
<td>2% virus.win32.virut</td>
</tr>
<tr>
<td>6825011</td>
<td>92488</td>
<td>74% net-worm.win32.allape</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>25% -unlabeled-</td>
<td>0% virus.win32.cheburgen</td>
</tr>
<tr>
<td>6823539</td>
<td>49148</td>
<td>61% -unlabeled-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>14% trojan.generic</td>
<td>9% not-a-virus:.webtoolbar</td>
</tr>
<tr>
<td>6789759</td>
<td>31984</td>
<td>50% -unlabeled-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10% trojan-clicker.html iframe</td>
<td>8% trojan-clicker.js.agent</td>
</tr>
<tr>
<td>6798627</td>
<td>25494</td>
<td>47% -unlabeled-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9% trojan-dropper.agent</td>
<td>7% virus.worm.win32.socks</td>
</tr>
<tr>
<td>6830115</td>
<td>24639</td>
<td>82% net-worm.win32.allape</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>17% -unlabeled-</td>
<td>0% backdoor.rbot</td>
</tr>
<tr>
<td>6830127</td>
<td>17440</td>
<td>98% -unlabeled-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0% trojan-downloader.win32.autoit</td>
<td>0% packed.win32.klone</td>
</tr>
<tr>
<td>6797651</td>
<td>16042</td>
<td>58% backdoor.win32</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>40% -unlabeled-</td>
<td></td>
</tr>
<tr>
<td>6618183</td>
<td>15483</td>
<td>69% -unlabeled-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>25% trojan-downloader.win32</td>
<td></td>
</tr>
<tr>
<td>6788535</td>
<td>14495</td>
<td>91% net-worm.win32.allape</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9% -unlabeled-</td>
<td>0% vlrus.wln32.cheburgen</td>
</tr>
</tbody>
</table>
Clustering Workflow

1) Periodic (e.g., weekly) full cluster runs:

- Blaster
- Fishing
- Slammer

2) Nearest neighbor search for each new sample:

- Nearest neighbor search
- Fishing
- Handyman
Lessons learned from 2 Years ANUBIS

Bot Analysis

- Bot analysis and IP address blacklisting become a problem
- Bot herders know IP range of public version of ANUBIS

ANUBIS Detection and Evasion

- Currently we’ve seen about 0.03% samples ITW with ANUBIS detection capabilities
- ANUBIS is capable of detecting if malware tries to evade ANUBIS
Some general Sandbox Problems

Timeout issues (general to automated sandbox analysis)

- Timeouts, how long shall the analysis run?
- Automatic analysis has to quit at some point (when?)

Most recent timeout problems

- Analysis of Mebroot malware resulted in empty ANUBIS logs
- Mebroot waits several minutes before infecting the system
- Watch out for empty logs!
- Timeout can not be altered in public online version (but in the in-house version this value is customizable)

Malware waiting for some user interaction

- Mouse movement/clicks, keystrokes, certain URL to be loaded
Packer with Anti-ANUBIS Features
Conclusion

ANUBIS offers technology to speed up malware analysis

- Automatic processing of incoming samples saves valuable time
- ANUBIS improves traditional analysis process flow with its features
- Clustering feature is unique to ANUBIS (AFAIK)
- Can offer additional functionality for “in the cloud” services (already used in academic research projects like WOMBAT/SGNET)
  See paper for more info on that.

Public version vs. commercial version

- Commercial version available on request
- Offers more features and keeps your samples in-house
- Offers customization (language, VM OS, 3rd party apps, ...)
- Offers integration into you existing pre-sorting process flow
Questions

Thank you for your attention!
We'd be happy to answer all of your questions!

Please send your questions to: anubis@ikarus.at, anubis@iseclab.org or thomas.mandl@mandl-itc.at