Malware Analysis, Software Obfuscation and the Ethics of Large Scale Malware Evaluation

SBA Research & Vienna University of Technology

Edgar R. Weippl
Sources

• PhD thesis and slides by Ulrich Bayer
• Master thesis and slides by Martina Lindorfer
• Presentation at CCC by Sebastian Schrittwieser

• Most content can be found at https://www.iseclab.org/
Agenda

• Introduction
  – Why large-scale dynamic malware analysis, static vs. dynamic
• Malware Analysis With ANUBIS
  – Anubis Core functionality, Submission Statistics
• A View on Current Malware Behaviors
  – Graphs, Trends
• Scalable, Behavior-based Malware Clustering
  – Finding malware families
• Improving the Efficiency of Dynamic Malware Analysis
• Conclusions
Automated Malware Analysis: Why?

• Too many new malware samples/day
  – Today’s number of malware samples make manual analysis impossible

• Automated malware collection (honeypots etc.)
Static analysis versus dynamic analysis

- **Static analysis**
  - code is not executed
  - all possible branches can be examined (in theory)

- **Problems of static analysis**
  - undecidable in general case, approximations necessary
  - disassembly difficult
    - obfuscated code, packed code
  - self-modifying code
Static analysis versus dynamic analysis

• Dynamic analysis
  – code is executed
  – sees instructions that are actually executed

• Problems of dynamic analysis
  – in general, single path (execution trace) is examined
  – analysis environment possibly not invisible
  – scalability issues
Large-Scale Malware Analysis

• Large-Scale Dynamic Malware Analysis refers to the problems that arise when dynamic malware analysis techniques are leveraged to examine a
  – big amount of malicious files
  – over a longer period of time
Large-Scale Malware Analysis - Challenges

• Large number of analysis results
  – We compile statistics and overviews
• Need ways to structure the information
  – Clustering
• Requires considerable computing resources
  – Avoid multiple analysis runs of polymorphic files
Anubis: Analyzing Unknown Binaries

- Anubis: A platform for dynamic malware analysis
  - Runs the binary in an emulated environment
    - Monitors Windows API and system service calls
    - Records network traffic
    - Tracks data flows

- Anubis receives samples through
  - A public web-interface (http://anubis.iseclab.org)
  - A number of feeds from security organizations
  - Originated from Uli Bayer’s master’s thesis
  - Is available online and analyzing (potentially) malicious samples since the beginning of 2007
Analysis Report for nepenthes-65c242c013045c678974e3be0796188d-index.html

Summary:

**Description**

<table>
<thead>
<tr>
<th>Description</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creates files in the <strong>Windows system directory</strong>: Malware often keeps copies of itself in the Windows directory to stay undetected by users.</td>
<td><img src="red" alt="Risk" /></td>
</tr>
<tr>
<td>Performs <strong>Address Scan</strong>: The executable scans a range of IP Addresses. In most cases these scans identify more potential vulnerable targets.</td>
<td><img src="red" alt="Risk" /></td>
</tr>
<tr>
<td>Performs <strong>File Modification and Destruction</strong>: The executable modifies and destructs files which are not temporary.</td>
<td><img src="red" alt="Risk" /></td>
</tr>
<tr>
<td><strong>Spawns Processes</strong>: The executable produces processes during the execution.</td>
<td><img src="yellow" alt="Risk" /></td>
</tr>
<tr>
<td>Performs <strong>Registry Activities</strong>: The executable reads and modifies registry values. It may also create and monitor registry keys.</td>
<td><img src="yellow" alt="Risk" /></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>b616dcf0c05e539b317edd9d279a267a6fad0c01e</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

### - Services Created:

<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:

<table>
<thead>
<tr>
<th>MSWindows</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSWindows</td>
</tr>
</tbody>
</table>
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......@
      0000 0000 0000 0000 0000 0000 8806

- 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
ANUBIS has 5 primary building blocks

- **Web/DB Server**
  - HTTP(s) frontend (upload/admin)
  - Relational DB stores reports
  - references to samples
- **Malware Sample Storage**
  - Files uploaded for analysis
- **Report Storage**
  - Archives report/result files (traffic dumps, downloaded files...)
- **Victim Server**
  - Acts as local honey pot for certain services
- **Worker (VM) Images**
  - Does all the analysis work!
ANUBIS Component Overview

Sample Upload via HTTP

INTERNET

Firewall

ANUBIS FRAMEWORK

Web / DB SRV

Worker VMs

Victim SRV

Report Store

Sample Store
Motivation

• Abstract from the thousands of individual behavioral descriptions
• Compile statistics on observed malicious behavior
  – File system activity
  – Registry activity
  – Network activity
  – GUI windows
  – Botnet activity
  – Sandbox Detection
• Uli’s Contributions
  – Shed light on common malicious (host) behavior
  – Confirm/deny folk wisdom
Registry activity

• 64.71% of all samples create registry keys
• 74.59% of all samples modify a registry value
• Some interesting activities:
  – 33.7% disable the windows firewall
  – 8.97% tamper with security settings (MSWindows\Security)
  – 35.8% of all samples modify the registry to get launched at startup
• Most common Autostart locations:
  – HKLM\System\Currentcontrolset\Services\%\ImagePath (17.53% of all samples)
  – HKLM\Software\Microsoft\Windows\CurrentVersion\Run% (16.00% of all samples)
Botnet Activity

- **Goal:** determine prevalence of HTTP, IRC or P2P Botnet clients
- **Typical bot behavior:**
  - Bots perform DDoS attacks
  - Send out spam e-mails
  - Download malicious executables
- **Our model:**
  - HTTP, IRC or P2P (BitTorrent, DC, Edonkey,..) traffic
  - Address scan, Port scan, DNS MX queries, high number of SMTP connections
  - Heuristics such as known IRC or URL strings
  - Create a blacklist of identified C&C servers
- **We identified 5.8% samples as bots**
Clustering: Motivation

• Thousands of new malware samples appear each day
• Automatic analysis systems allow us to create thousands of analysis reports
• Now a way to group the reports is needed. We would like to cluster them into sets of malware reports that exhibit similar behavior.
  – we require automated clustering techniques
• Clustering allows us to:
  – guide an analyst in the selection of those samples that require most attention
  – derive generalized signatures, implement removal procedures that work for a whole class of samples
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”)

- Blaster
- Phishing
- 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
System Overview

Dynamic Analysis of the Sample

Result

Execution Trace augmented with taint-information and network analysis results

Result

Input

Extraction of the Behavioral Profile

Behavioral Profile

Input

Clustering
Extraction of the Behavioral Profile

• In this step, we process the execution trace provided by the ‘dynamic analysis’ step

• Goal: abstract from the system call trace
  – system calls can vary significantly, even between programs that exhibit the same behavior
  – remove execution-specific artifacts from the trace

• A behavioral profile is an abstraction of the program's execution trace that accurately captures the behavior of the binary
Scalable Clustering

- Most clustering algorithms require to compute the distances between all pairs of points $\Rightarrow O(n^2)$
- We use LSH (locality sensitive hashing), a technique introduced by Indyk and Motwani, to compute an approximate clustering that requires less than $n^2$ distance computations
- Our clustering algorithm takes as input a set of malware samples where each malware sample is represented as a set of features
  - we have to transform each behavioral profile into a feature set first
- Our similarity measure: Jaccard Index defined as

$$J(a, b) = \frac{|a \cap b|}{|a \cup b|}$$
Performance Evaluation

• Input: 75,692 malware samples

• Previous work by Bailey et al
  – extrapolated from their results of 500 samples
  – Number of distance calculations: 2,864,639,432
  – Runtime: 995 h (~ 6 weeks)

• Our results:
  – Number of distance calculations: 66,528,049
  – Runtime: 2h 18min
Conclusions

• Novel approach for clustering large collections of malware samples
  – dynamic analysis
  – extraction of behavioral profiles
  – clustering algorithm that requires less than a quadratic amount of distance calculations

• Experiments on real-world datasets
  – demonstrate that our techniques can accurately recognize malicious code that behaves in a similar fashion
Detecting Identical Behavior: Motivation

• During the last three years the number of malware programs appearing each day has increased by a factor of ten

• Keeping pace with this development causes considerable hardware costs

• Is it possible to optimize the analysis process i.e., to analyze more samples in the same time while maintaining the quality of our results?

• Our insight: the huge number of new malicious files is due to mutations of only a few malware programs
Pre-empting behaviorally identical samples

Execution of program **a**:  
- Checkpoint-time
- Behaviorally identical
- Analysis-End

Execution of program **b**:  
- Checkpoint-time
- Analysis-End
Prototype Implementation

• We represent a program’s behavior in the form of a “behavioral profile”
  – i.e., after 60 seconds we create a behavioral profile

• We consider two programs behaviorally identical at time $t$ if $J(a,b) < d$

• We employ LSH to allow for efficient and scalable nearest neighbor searching
  – instead of having to perform $n-1$ comparisons
Overview of approach

Analyze a binary $b$ for time $T_c$

Create a behavioral profile $bp(b)$

Find behavioral profile $p$ with $\text{dist}(p, bp(b)) < d$

Continue analysis

Stop Analysis

Return analysis report of $p$

Found no near profile

Store LS hashes

Found $p$
Overview of approach

Analyze a binary $b$ for time $T_c$
Create a behavioral profile $bp(b)$

Find behavioral profile $p$ with $\text{dist}(p, bp(b)) < d$

Return analysis report of $p$
Found $p$
Store LS hashes
Continue analysis
Found no near profile
Overview of approach

Analyze a binary \( b \) for time \( T_c \)
Create a behavioral profile \( bp(b) \)
Stop Analysis
Find behavioral profile \( p \) with \( \text{dist}(p, bp(b)) < d \)
Return analysis report of \( p \)
Found no near profile
Store LS hashes
Continue analysis
Found \( p \)
Stop Analysis
Return analysis report of \( p \)
Evaluation

• Real-world experiment
  – We operated our prototype inside the Anubis platform for several days
  – We analyzed 10,922 unique samples in this period

• Results:

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Pre-empted files</th>
<th>Time saved/pre-emption</th>
<th>Total time saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>60s, d = 0.12</td>
<td>2,747 (25.15%)</td>
<td>250s</td>
<td>190.8 hours</td>
</tr>
</tbody>
</table>
Conclusion

• Operation of Anubis, an automated dynamic analysis system, since February 2007.
  – Allows us to provide statistics and insights into common malicious behaviors

• A novel and scalable clustering technique
  – produces more precise results than previous approaches
  – based on an abstract representation of a program’s behavior

• A novel and practical approach for improving the efficiency of dynamic malware analysis systems
  – We suggest a technique that avoids performing a full analysis of the same polymorphic file multiple times
References

- The previous slides were mainly Uli Bayer’s PhD thesis and publications.
- The following slides are Martina Lindorfer’s work. She at NII right now:
Detecting Environment-Sensitive Malware
Evasion Techniques

• “Environment-sensitive” malware checks for
  – Characteristics of the analysis environment
  – Characteristics of the Windows environment
• Emulation/Virtualization detection
• Timing
• Unique identifiers
• Running processes
• Restricted network access
• Public IP addresses
Evasion Detection

- Execute malware in multiple environments
- Detect deviations in behavior and identify root cause
- Modify analysis sandboxes to thwart evasion techniques
DISARM

“DetectIng Sandbox-AwaRe Malware”

• Agnostic to root cause of divergence in behavior
• Agnostic to employed monitoring technologies

• Automatically screen samples for evasive behavior
• Collect execution traces in different environments
• Eliminate spurious differences in behavior caused by different environments
• Compare normalized behavior and detect deviations
• Use findings to make sandbox resistant against evasion
Disarm
Disarm

• Execution monitoring
  - Execute malware in multiple sandboxes (Qemu, Anubis, ...)
  - Different monitoring technologies & Windows installations

• Behavior comparison
  - Normalize behavior from different environments
  - Measure distance of behavior and calculate evasionscore
Disarm - Evasion score

• Intra-sandbox distance (diameter) between executions in the same sandbox
• Inter-sandbox distance (distance) between executions in different sandboxes
References


• Martina Lindorfer, Clemens Kolbitsch, and Paolo Milani Comparetti, Detecting Environment-Sensitive Malware, International Symposium on Recent Advances in Intrusion Detection (RAID 2011), Menlo Park, CA, September 2011

References


• Christopher Kruegel and Engin Kirda and Ulrich Bayer and Davide Balzarotti and Imam Habibi, "Insights Into Current Malware Behavior," in 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), Boston, 2009.

ETHICS
Ethics

• “Once the rockets are up, who cares where they come down? / That's not my department”

• Tom Lehrer - Wernher von Braun (1965)
Tuskegee Syphilis Experiment
Tuskegee Syphilis Experiment

- Patients were not informed about available treatments
- No precautions were taken that patients did not infect others
- They were also actively given false information regarding treatment
Your Botnet is My Botnet: Analysis -

Brett Stone-Gross, Maija Ollila
Richard Kemmerer
Department of Computer Science
University of Wisconsin-Madison
bstone@cs.wisc.edu

Abstract
Botnets, networks of malware-infected
by an adversary, are the root cause of
problems on the Internet. A particu-
larly large type of bot is "Torpig," a malware
harvest sensitive information (such as
data) from its victims. In this paper, we
control of the "Torpig" botnet and as-
we ten days. During this period, we observed
infections and recorded almost 70
"Torpig" botnets have been "Tor-

Shining Light in Dark Places:
Understanding the Tor Network

- McColley

Is the Internet for Porn?
An Insight Into the Online Adult Industry

- Holz, Plutzer, Kruegel

PharmaLeaks: Understanding the Business of
Online Pharmaceutical Affiliate Programs

- McColley, Jordan, Weaver

- McColley, Savage, Liben

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weaver

- McColley, Liben, Weav
Quotes

- Spamalytics
  - “passive actors”
  - “ensuring neutral actions”
  - “users should never be worse off due to our activities”

- Your botnet is my botnet
  - “The sinkholed botnet should be operated so that any harm and/or damage to victims and targets of attacks would be minimized”
  - “The sinkholed botnet should collect enough information to enable notification and remediation of affected parties”
PharmaLeaks

“[...] ethics of using data that was, in all likelihood, gathered via illegal means. [...] We justify our own choice [...] by reasoning about harm.

“some [...] contents have already been widely and publicly documented. Consequently, we cannot create any new harm simply through association with these entities or repeating these findings”

Is the Internet for Porn

“Clearly, one question that arises is if it is ethically acceptable [...] to participate in adult traffic trading. [...] we believe that realistic experiments are the only way to reliably estimate success rates of attacks in the real-world”

“we did not withdraw any funds but forfeited our traffic trading accounts at the end of the experiments”
Papers


Papers
