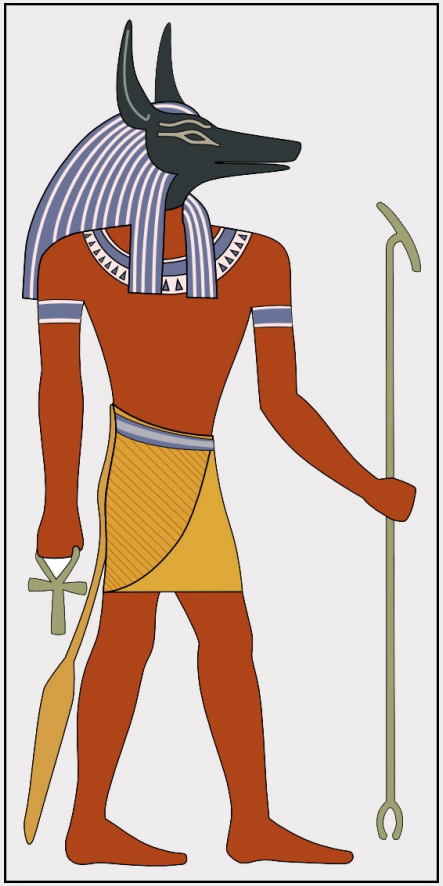


ANUBIS

A platform for the analysis of malicious code

Ulrich Bayer
ulli@seclab.tuwien.ac.at
Secure Systems Lab - TU Vienna

Agenda

1. Introduction
   Who is behind Anubis?, Project goals

1. Malware Analysis With ANUBIS
   The need for automated malware analysis, static vs. dynamic, Anubis Core functionality

1. The Online Anubis platform
   Submission Statistics, Architectural Overview

1. Advanced Anubis Features
   Data Tainting, Clustering (find malware families)

Agenda (cont.)

1. Anubis Reference Projects
   SGNET, WOMBAT

1. Anubis Analysis Issues
   Detection of Anubis/QEmu, Triggers

1. Conclusion and Current Developments

About myself

- Ulrich Bayer, born in Austria
- Studied computer science at the TU Vienna
- Since 2006, PhD student at the TU Vienna
- Currently visiting scientist at Eurecom, France

- Master's thesis: "TTAnalyze: A Tool For Analyzing Malware"
  - Carried out at the Seclab TU Vienna
  - In cooperation with Ikarus Software
  - Predecessor of ANUBIS

Who's behind ANUBIS (1)

- **International Secure Systems Lab**
  - Research group
  - Online: http://www.iseclab.org
  - Founded in 2005 at the TU Vienna, Austria by

    Engin Kirda, PhD, Assistant Professor at Eurecom, France

    Christopher Kruegel, PhD, Assistant Professor at UCSB, US
  - Research on system security, > 10 PhD students

    e.g., Web-Security, Spam, Malware/Spyware Analysis
  - Now geographically distributed over three locations (Vienna, Eurecom, UCSB)
  - Hosting public ANUBIS website (http://anubis.iseclab.org)

Who's behind ANUBIS (2)

- **IKARUS Security Software**
  - Austrian A/V company (based in Vienna)
  - Commercial partner and distributor for ANUBIS
  - Already funded TTAnalyze, the predecessor of Anubis
  - Distribute a commercial version of Anubis

    Trial version is available too.

    More details: anubis@ikarus.at

Anubis Team

- **Main developers**
  - Ulrich Bayer (Anubis, Database, Webserver, Admin, Clustering)
  - Florian Nentwich (Ikarus)
- **Developers**
  - Paolo Milani Comparetti (Post-Doc, Clustering)
  - Clemens Hlauschek (Clustering)
  - Valentin Habsburg
  - Sylvester Keil
  - Florian Lukavsky
  - Matthias Neugschwandtner
  - Michael Weissbacher
- **Scientific Advisors**
  - Engin Kirda
  - Christopher Kruegel

Project Goals

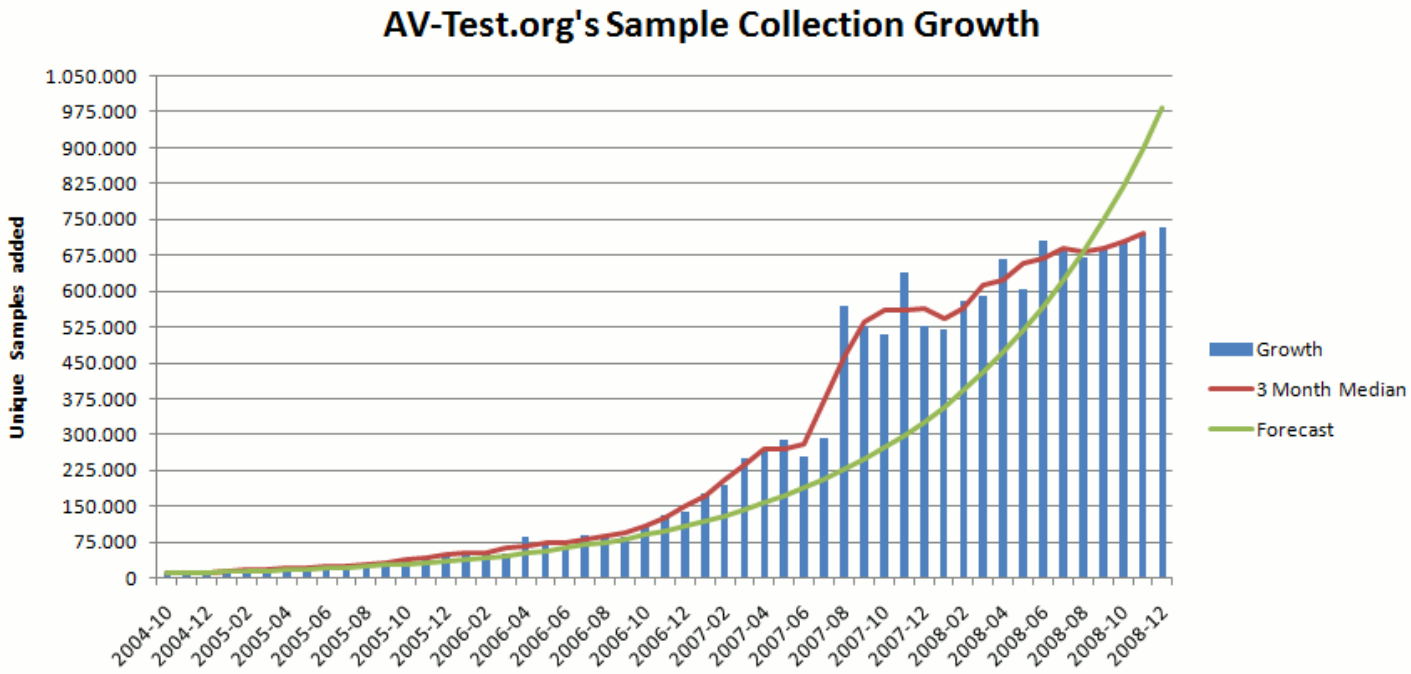
- **Seclab**: Research Prototype
  - Access to virus samples
  - Allows us to see current malware behavior
  - Real world operation: Opens new research problems
  - Provides the infrastructure for several other research projects (multiple execution paths, botnet monitoring/detection/analysis, clustering…)
  - Great source of topics for student internships/master thesis
- **Ikarus**: Internal Tool
  - Internal Tool designed to help in the presorting of malware
  - Build in-house high-technological assets
  - Technology Transfer University -> Company

Chapter 2

Malware Analysis With Anubis

Automated Malware Analysis: Why?

- **Too much new malware samples/day**
  - Really nobody can handle this!
- Automated malware collection (honeypots etc.)



Anubis: Core Functionality

- We **run** the binary
  - Dynamic analysis
- in an **emulated** environment
  - Emulation of a complete PC (CPU, hardware devices)
  - Qemu used as emulation environment
  - We've installed an out of the box Windows XP SP2
  - Completely transparent to sample
- and we **monitor** its actions
  - System Calls, Windows API calls

Static analysis versus dynamic analysis

- **Static** analysis
  - code is not executed
  - all possible branches can be examined (in theory)
  - quite fast
- Problems of static analysis
  - undecidable in general case, approximations necessary
  - disassembly difficult (particularly for Intel x86 architecture)
  - 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*
  - analysis environment possibly not *comprehensive*
  - scalability issues

Anubis Analysis-Report

- File Activities
  - Read, write, create,…
- Registry Activities
  - Create, change, delete a registry key/value
- Process Activities
  - create, terminate, inter-process communication
- Windows Service Activities
  - Start or Stop Windows Services
- Network Activities
  - DNS, HTTP/FTP Downloads, SMTP/IRC conversations, …

- **Let's look at an example Anubis report [1]**

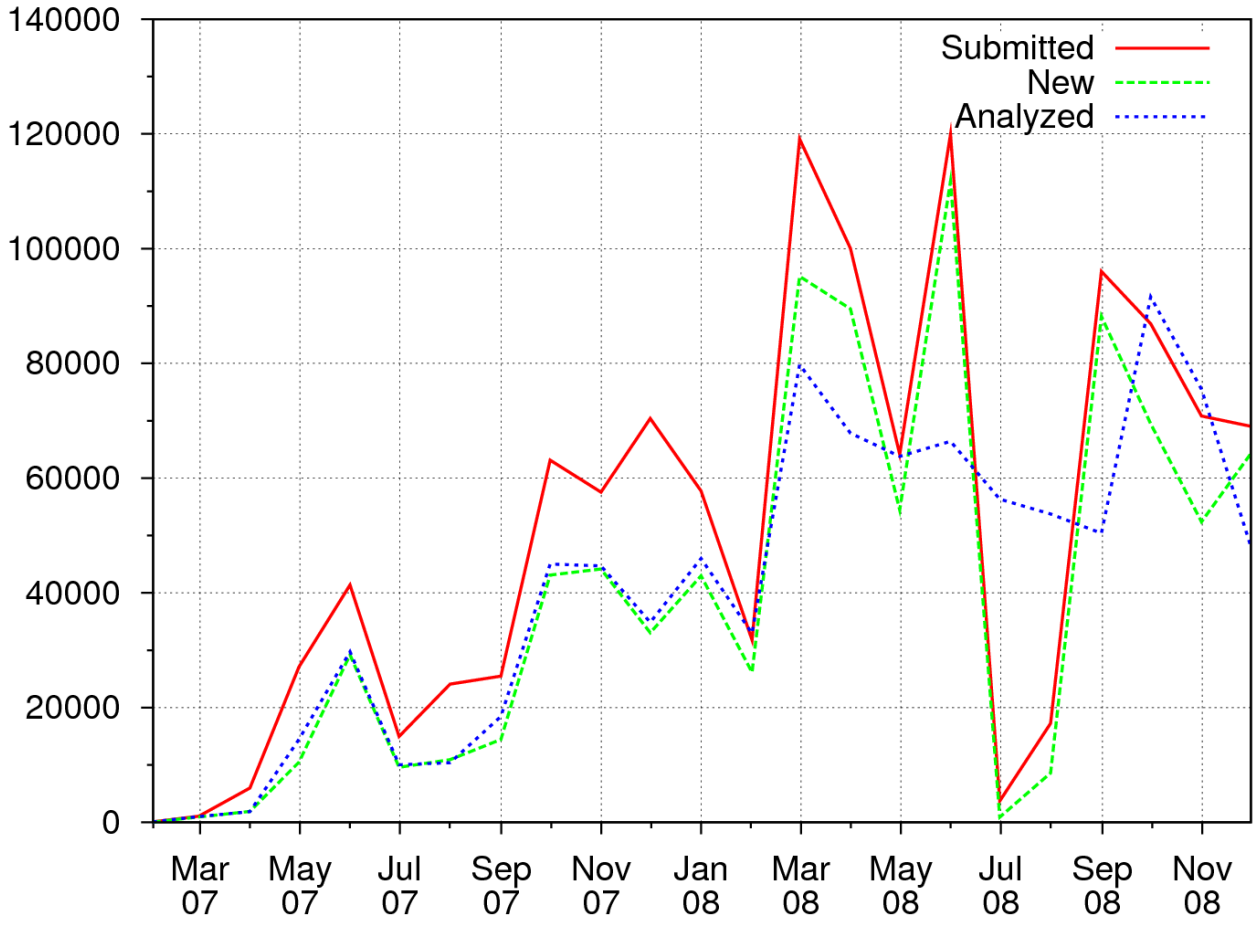
Benefits of ANUBIS

- **Detailed reports after 4 min.**
  - Manual in-depth analysis > 72h (no code obfuscation!)
- **ANUBIS uses sandbox technology**
  - Non-intrusive inspection from "outside" leads to better results
  - Classic VM detection doesn't always work (VMware, Virtual PC)
  - Though ANUBIS detection is possible (more on that later…)
- **But ANUBIS still requires experts for operations**
  - Management summary on top of the report gives quick overview
  - Interpretation of detailed reports still needs expert know-how

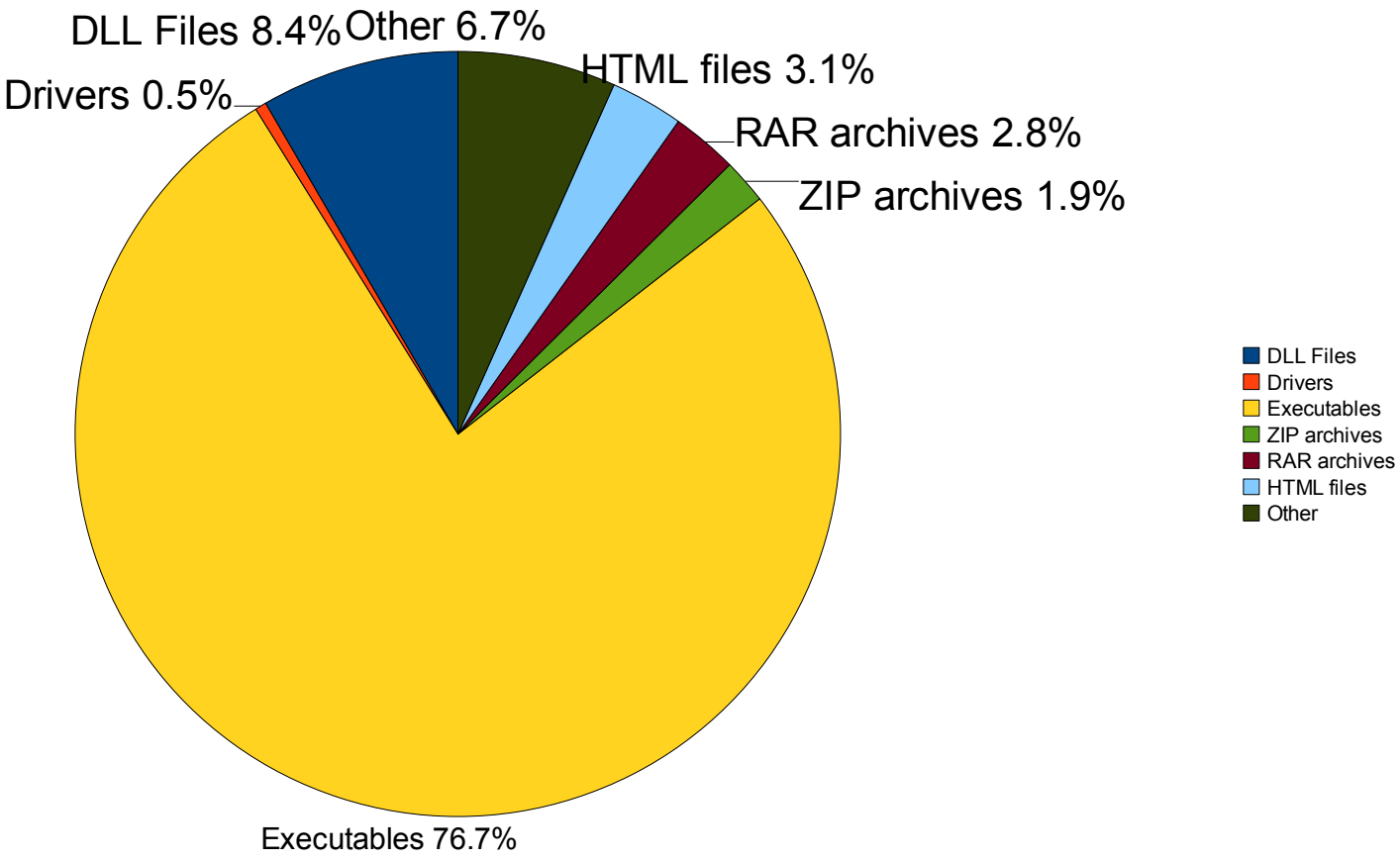
Chapter 3

The online ANUBIS platform http://anubis.iseclab.org

Anubis Submission Statistics

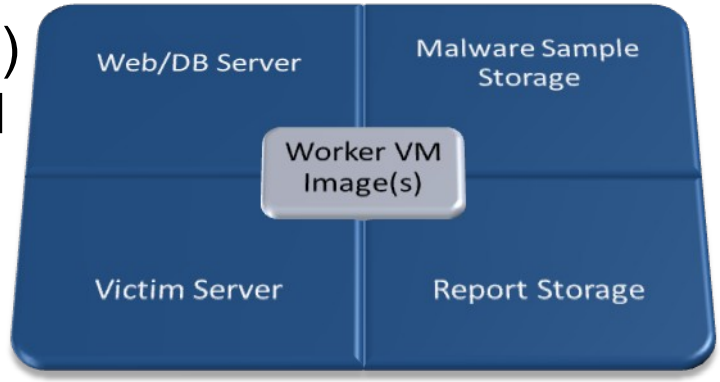


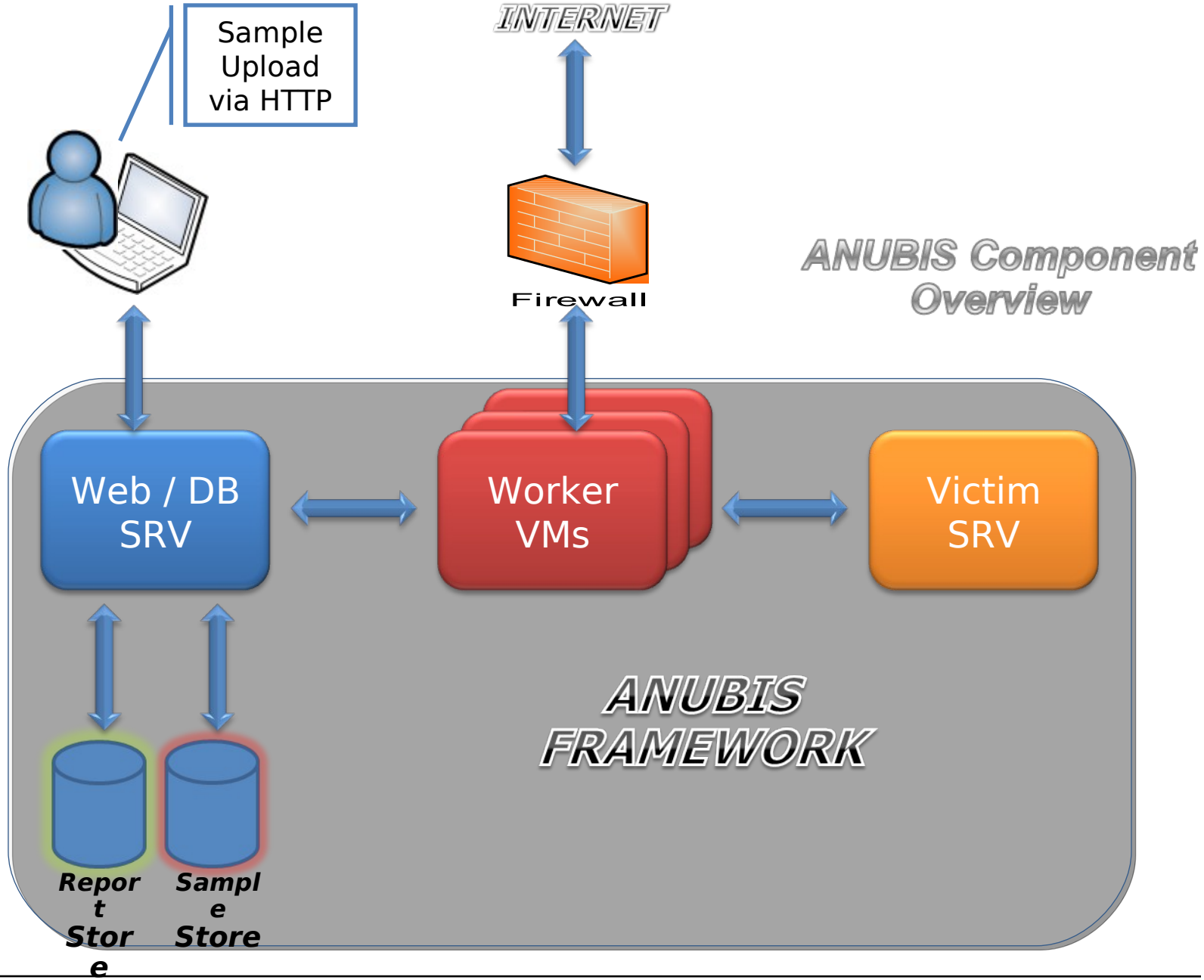
Submitted File Types



Architecture and Capabilities

- **ANUBIS has 5 primary building blocks**
  - Web/DB Server
    HTTP(s) frontend (upload/admin)
    Relational DB stores reports and references to samples
  - Malware Sample Storage
    Archives uploaded and already analyzed samples
  - Report Storage
    Archives report/result files (traffic dumps, downloaded files...)
  - Victim Server
    Acts as local honeypot for certain services
  - Worker (VM) Images
    Does all the analysis work!




Sample Upload via HTTP
INTERNET
Firewall
ANUBIS Component Overview
Web / DB SRV
Worker VMs
Victim SRV
ANUBIS FRAMEWORK
Report Store
Sample Store

Chapter 4

Advanced ANUBIS features

Advanced Features

- Records and analyzes network traffic
  - HTTP, FTP, SMTP, IRC, …
- Storage of analysis reports in relational DB
  - What Servers have been contacted, what files created, …
- Several Report Formats
  - XML, HTML, MHT, PDF, TXT
- URL Analysis
- Tracking of data flows (more info later)
- Clustering (more info later)
- …

Memory Tainting Overview

- Powerful technique for tracing data flows of a program
  - E.g., how network data is processed by a program

- How does tainting work?
  - performed on hardware level, using a system emulator
  - bytes in (emulated) physical memory are labeled, using a shadow memory
  - taint sources: each data element of interest is labeled (tainted)
  - taint propagation

    When memory values are copied => copy taint labels

Memory Tainting Example

Consider the following code fragment

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

Creates Random Filename

Enhanced with tainting information

```
ticks = GetTickCount()
```

ticks → <GetTickCount>

Tainting Label

```
filename = "c:\\" + ticks + ".exe"
```

filename → <GetTickCount>

```
file = CreateFile(filename, ...)
```

=> CreateFile is called with a random filename

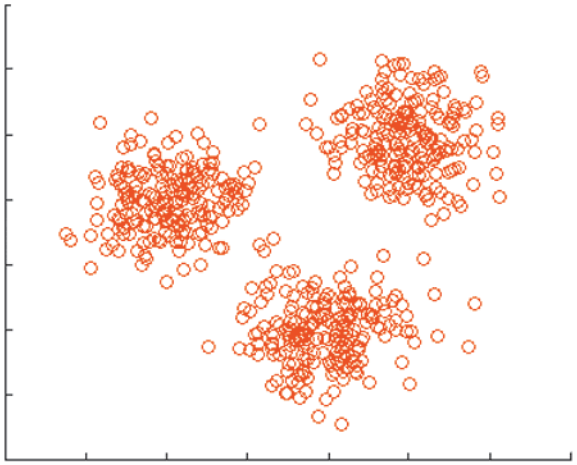
Clustering: Motivation

- 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:
  - discard reports of samples that have been seen before
  - 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

Scalable, Behavior-Based Malware Clustering

Secure Systems Lab
Technical University Vienna

- **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”)



- **Behavior-Based:** A malware sample is represented by its actions performed at run-time
- **Scalable:** It has to work for large sets of malware samples

Clustering

- Clustering is online since February 2009

- Last Clustering Run (June 7th 2009):
  - http://anubis.iseclab.org/?action=browse_clusters&task=259
  - Runtime: 5h38m
  - Number of clustered samples: 683,791
  - Number of clusters: 74,526
  - Among the biggest clusters there are several Allaple clusters

Chapter 5

ANUBIS Reference Projects

Leurré.com v2.0, SGNET

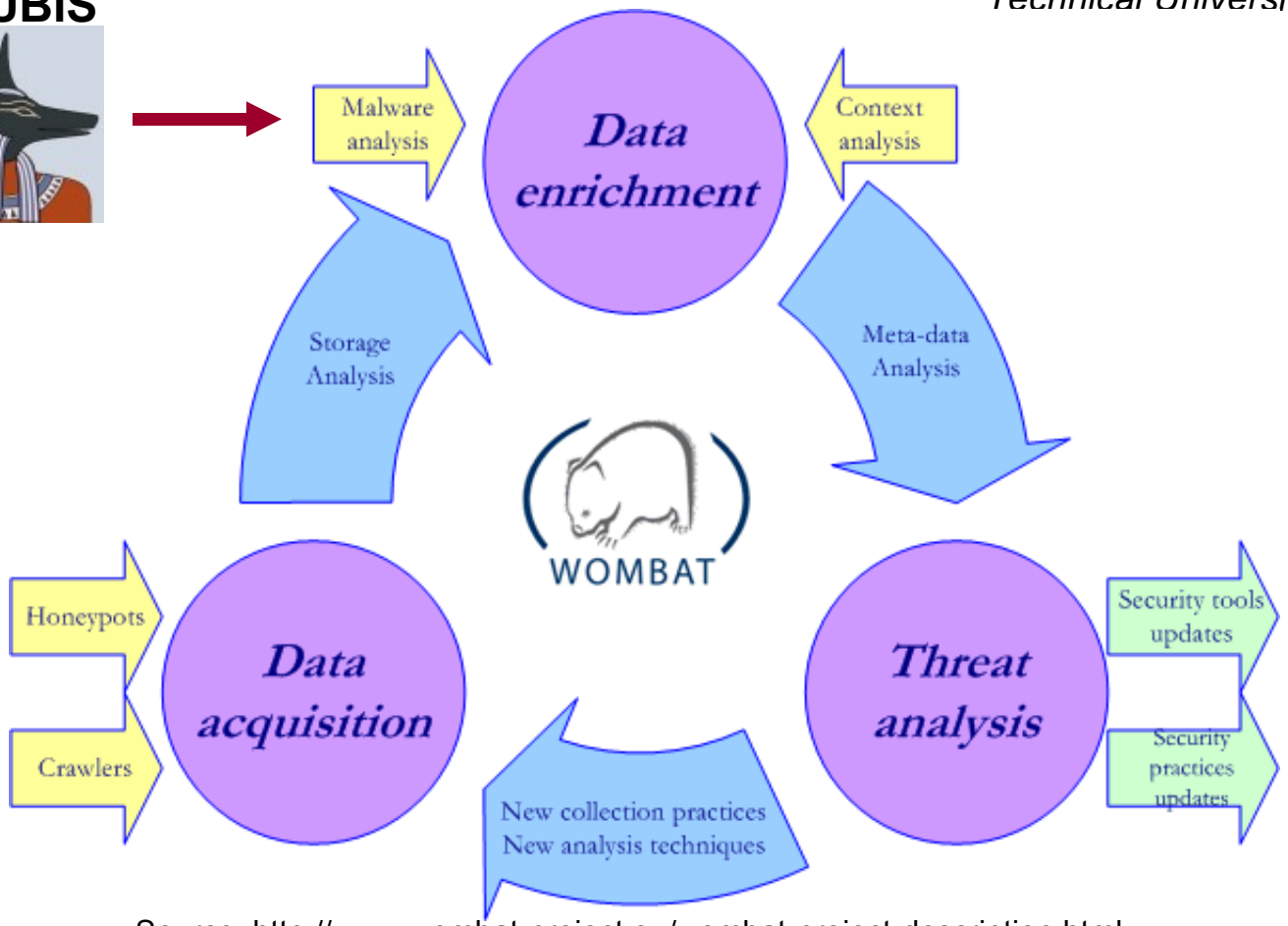
- **Based on Fabien Pouget's Honeynet Project (v1.0)**
- **SGNET - a distributed infrastructure to handle zero-day exploits**
- **Academic People involved**
  - Corrado Leita, Marc Dacier (Director of Research @Symantec)
- **SGNET =**
  - *Scriptgen* (Eurecom) + *Argos* (VU Amsterdam) + *Nepenthes* (TU Mannheim) + *ANUBIS* (TU Vienna) + *Virustotal* (Hispasec)
  - Continue honeypot conversation with the attacker up to the point, where malware is downloaded (resp. uploaded)
  - Sensors feed potential malware automatically into ANUBIS and Virustotal for further analysis. Results are archived in DB

WOMBAT

- **EU project**
- **Worldwide Observatory of Malicious Behaviours and Attack Threats**
  - Started 01/08
  - http://www.wombat-project.eu/wombat-project-description.html
- **Objectives of WOMBAT**
  - new means to understand existing and emerging Internet threats
  - Implements automated analysis using ANUBIS
- **Major Partners**
  - VU Amsterdam, Eurecom, FORTH, PoliMilano, TU Vienna

Role of ANUBIS in WOMBAT

ANUBIS



Source: http://www.wombat-project.eu/wombat-project-description.html

Chapter 6

ANUBIS Analysis Issues

Anubis Analysis Issues

- Evasion
  - attacks against Qemu
  - specific attacks against Anubis sandbox
  - blacklisting of our IP addresses and DNS names
- Timeout
  - 4 minutes (real-time) per analysis
- Single execution path only
  - may miss trigger behavior
  - some malware disables itself after some deadline

Timeout - Problem

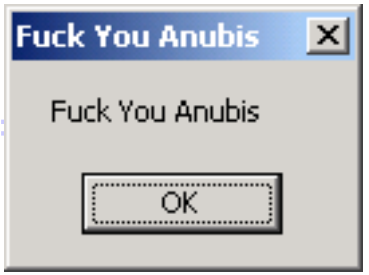
- **General to all sandboxed solutions**
  - Timeouts, how long shall the analysis run?
  - Automatic analysis has to quit at some point (when?)
- **Most recent timeout problems**
  - Analysis of Mebroot malware resulted in empty ANUBIS logs
  - Mebroot waits about 20 min. before infecting the system
  - 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

Known Ways to detect ANUBIS

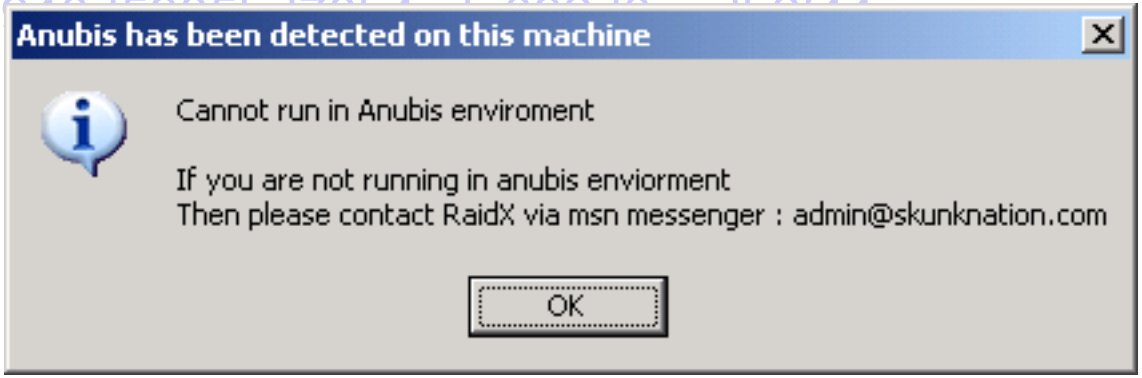
- **Malware Scene's Response (defeating ANUBIS)**
  - Check whether current Windows username equals “andy” or “user”
  - Check Windows Product ID
  - Check whether the file *C:\exec.exe* exists
  - Check whether the executable name equals *C:\sample.exe*
  - Check whether the computer name

ANUBIS-aware Malware

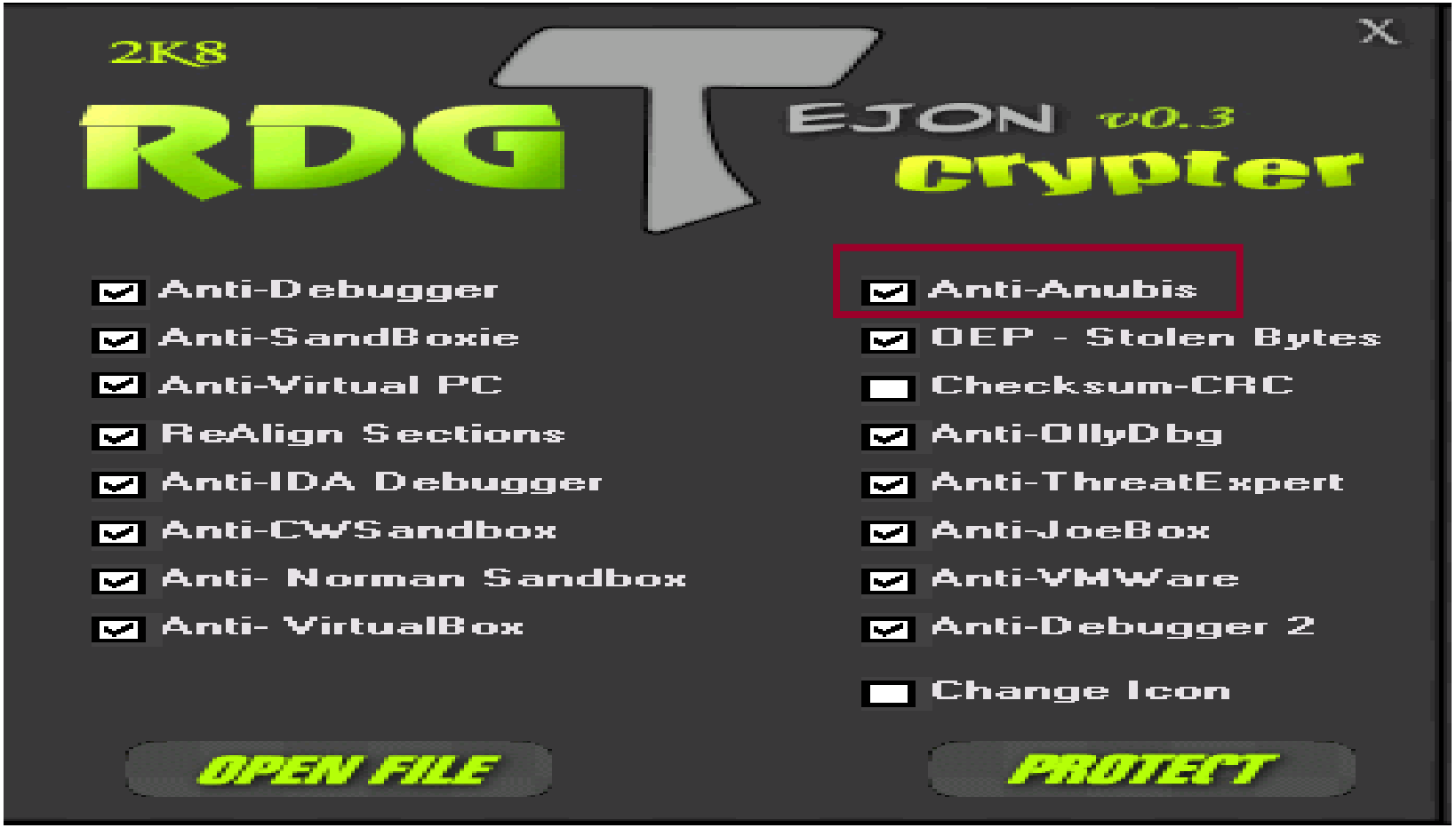
- **ANUBIS aware Malware**
  - https://anubis.iseclab.org/index.php?action=result&task_id=68f521af923abac4319a3ce6d3a85678



- **Detection of ANUBIS terminates Malware Process**
  - https://anubis.iseclab.org/index.php?action=result&task_id=07



Packer with Anti-Anubis Features



Chapter 7

Conclusions and Current Developments

Current Developments

- Anti Anubis-Detection
- Improved Network Analysis
  - Recognition of Exploits in Network Traffic, Bugfixes,…
- Better Statistics
- Adaptive Analysis End
- Incremental Clustering

Conclusion

- **Anubis Project**
  - Partners and Goals
- **Automatic, Dynamic Analysis with ANUBIS**
  - Analysis is a fully automated task with extreme time saving
  - helps quickly identifying potential threats
- **Advanced ANUBIS Features**
  - Tracking information flows via tainting
  - Clustering
- **Anubis Analysis Issues**
  - Detection of Anubis/Qemu
  - Single execution path

Questions?

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