Zero Wine Tryouts

An Open Source Malware Analysis Tool

Chae Jong Bin

About me

- Chae Jong Bin, born in South Korea
- Security researcher
- Malware analyst
- Software developer
- Studying computer science at the Kwangwoon University, South Korea

Project members

- Chae Jong Bin
  - Project maintainer, Developer
- Frank Poz
  - Developer

What is it?

Zero Wine Tryouts is an open source malware analysis tool.

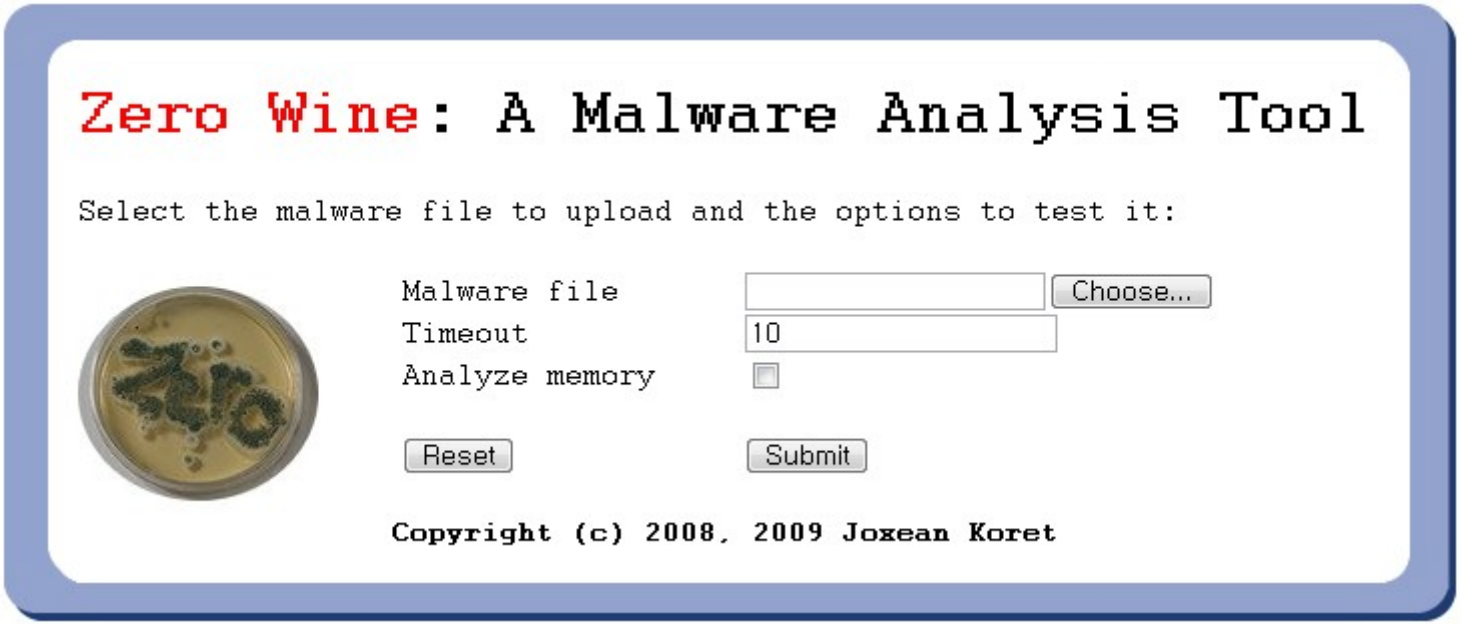
Just upload your suspicious file (e.g., Windows executable file, PDF file) through the web interface and let it analyze.

Zero Wine + X = Zero Wine Tryouts

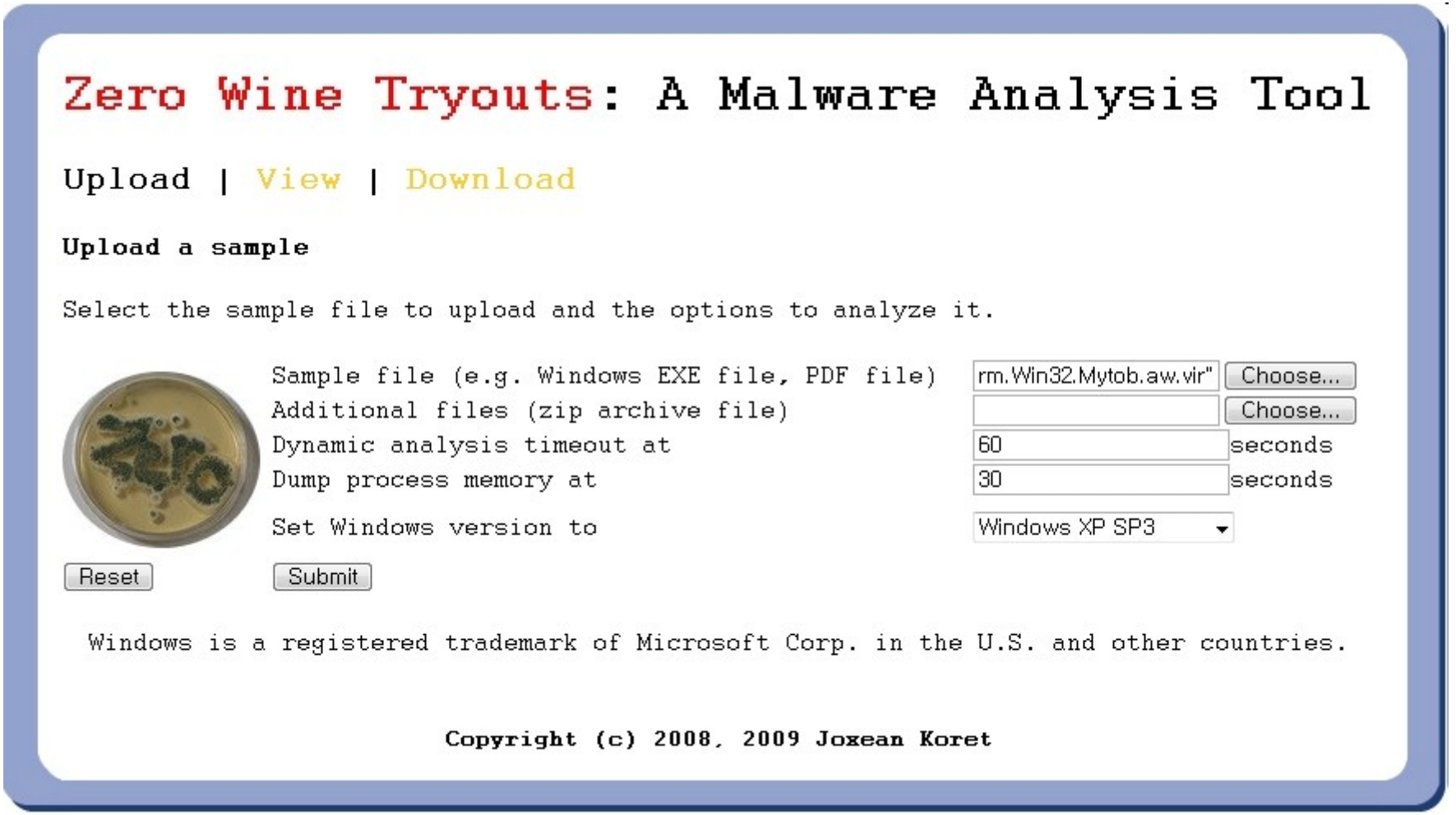
The Zero Wine Tryouts project is a fork of the original Zero Wine project.

The last modification to the source code of the original project was done back in Jan 2009. (Version 0.0.2.1)

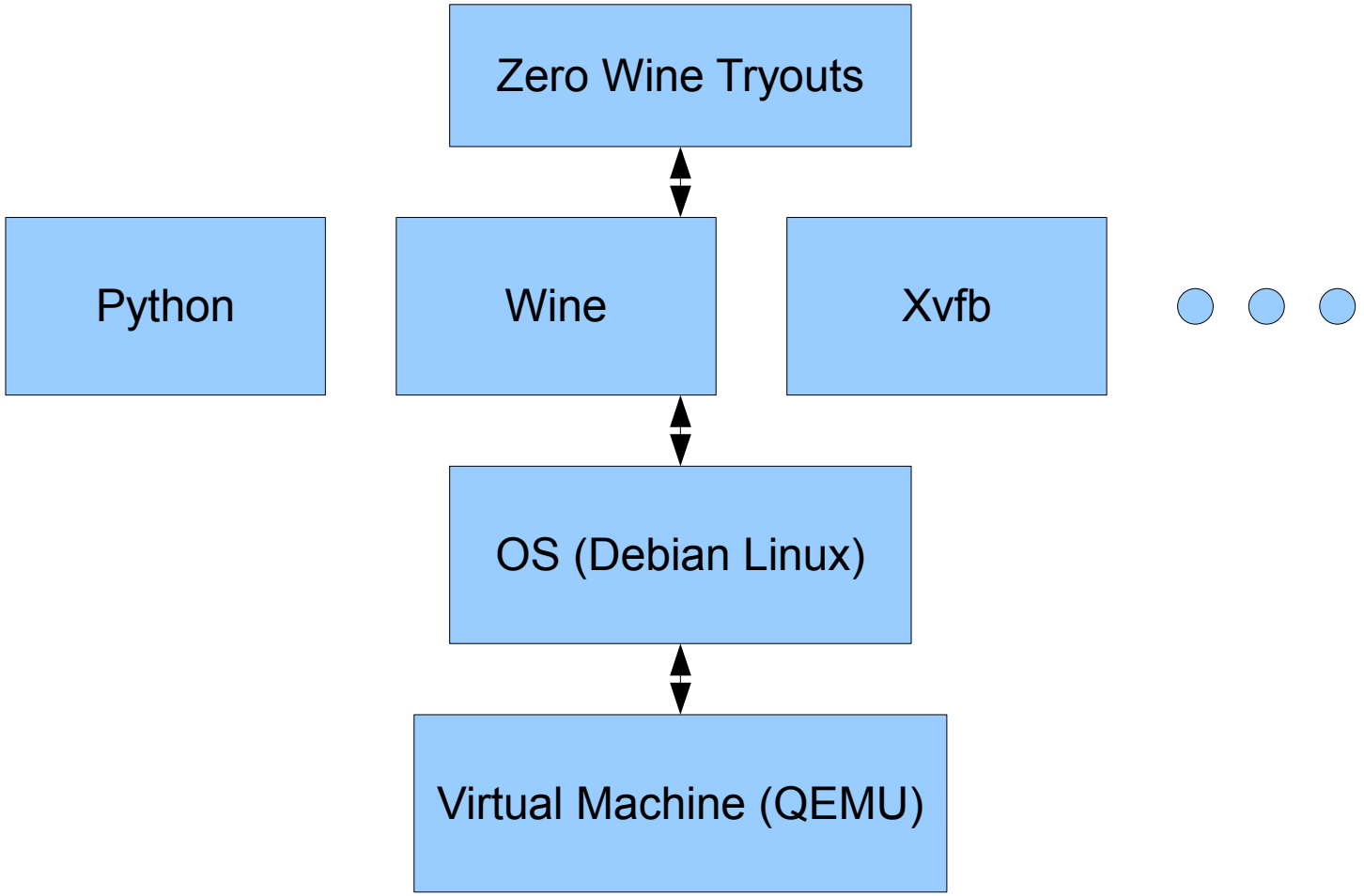
Zero Wine
(By Joxean Koret)



Zero Wine Tryouts



Architecture



Features

Static analysis

+

Dynamic analysis

Static analysis

- All files
  - Generate hash values (e.g., MD5, SHA-1)
  - Identify file types (via TrID)
  - Extract strings (via Strings)
  - Anti-VM tricks detection

Static analysis (Cont'd)

- Windows executable files
  - Packer detection with PEiD's signatures (via pefile)
  - Header Inspection (via pefile)
  - Sections analysis (via pefile)
  - Unpack (via UPX)
- Adobe PDF files
  - Extract JavaScript
  - Analysis (via pdftk, pdfid.py and pdf-parser.py)
  - Uncompress (via pdftk)

DEMO

Dynamic analysis

- Windows executable files
  - API trace (via Wine's WINEDEBUG)
  - Process dump (via python-ptrace)
  - File/Registry differences (via diff)
  - Network packet capture (via TCPDUMP)
  - Some tricks detection (e.g., Anti-Debugger, Anti-AV)

DEMO

Known problems

- Security
  - Wine is not sandbox nor secure
    - Can be escaped
      - int 0x80, sysenter
        - Possible solution: System call interception (Patch Linux kernel)
    - Can be detected in many ways
      - Registry
      - Files
      - Etc
        - Possible solution: Patch Wine

Known problems (Cont'd)

- Compatibility
  - Wine is not Windows
- Slow speed
  - More disk I/O than original Zero Wine
  - Web browser timeout
  - Visual Basic executable

Known problems (Cont'd)

- Dirty clean up
  - Sometimes unable to kill Wine processes
    - Need to kill Wine processes manually
- PDF uncompressing
  - Filter problem
    - Limitation of pdftk
- Etc

Todo

- Function
  - Dynamic Analysis
    - Real-time monitoring (Suggested by Keivan Komeilipour)
    - Network dump analysis (Suggested by Curt Wilson)
    - Better timeout handling
      - If process crashed, return immediately
    - Better process memory dump
    - Better Windows version setting
    - Multi-user support
    - Screenshot support
    - Reboot support

Todo (Cont'd)

- Static Analysis
  - Better PDF uncompressing
    - pdf-parser.py (Suggested by Paul Melson)
    - jsunpack-n pdf.py
    - MuPDF pdfclean
  - JavaScript
    - Analysis support (Suggested by Adnan bin Mohd Shukor)
    - Deobfuscator support
  - Disassembler support
  - Binary Analysis Tool support (Suggested by Keivan Komeilipour)

Todo (Cont'd)

- Both (Static and dynamic analysis)
  - Verbosity level tweak (Suggested by Curt Wilson)
  - Generate low-level & medium-level report (Suggested by Curt Wilson)
    - MAEC (Malware Attribute Enumeration and Characterization)
      - Conficker Worm Characterization
  - Export result to other format
    - XML
    - HTML
    - CSV
  - Add more useful functions
  - Database support

Todo (Cont'd)

- Documentation
  - Better documentation
- Security
  - Secure Wine
    - Sandboxing
    - Anti-Anti-Debug
      - Anti-Anti-Wine

Todo (Cont'd)

- Compatibility
  - Microsoft .net (MSIL) application support
- User interface
  - Better HTML output
- Etc
  - Code refactoring

Project final goal

“Upload-and-forget”

Prebuilt QEMU image and source code available at

http://zerowine-tryout.sourceforge.net

Thank you!

Have any questions?

Twitter: @2gg