



SAPIENZA  
UNIVERSITÀ DI ROMA

# BlueTracer: a Robust API Tracer for Evasive Malware

**Simone Nicchi**

*Thesis Advisor: Prof. Camil Demetrescu*

*Thesis Co-Advisors: Dr. Daniele Cono D'Elia, Dr. Emilio Coppa*

**Master of Science in Engineering in Computer Science**

July 20, 2018

# Function call monitoring

Functions can abstract implementation details providing a semantically richer representation of some functionality.

Example:

`[2, 4, 1, 3, 5] → sort() → [1, 2, 3, 4, 5]`

The abstractions embodied by **system calls** and **library calls** can be used to grasp the visible behavior of a malicious sample

# Implementation of function call monitoring

## API Hooking

The interception of function calls provided by dynamically linked libraries (DLLs)

Three broad categories:

- Binary Rewriting
  - Call Redirection
  - Function Rewriting
- Virtual Machine Introspection (VMI)
- **Dynamic Binary Instrumentation (DBI)**



# Dynamic Binary Instrumentation (DBI)

A dynamic binary analysis technique in which the behaviour of an application is inspected at run-time via the injection of analysis code.

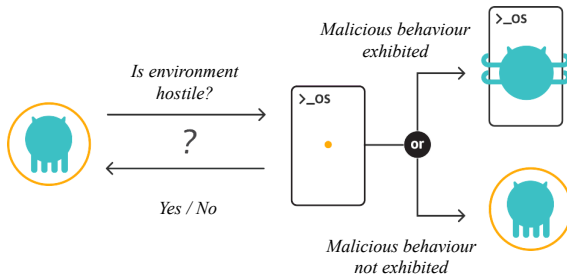
```
record_before(libcall_name, arg1)  
retval = libcall(arg1, &arg2)  
record_after(retval, *arg2)
```

**Problem 1:** existing products have limited logging capabilities

# The threat posed by evasive malware

## Evasive malware

Malware that conceals its harmful behaviour when detecting a hostile environment, such as a well-known sandbox solution



**Problem 2:** API hooking techniques in literature are not coupled with mechanisms to hide their presence from evasive malware

# Our solution: BlueTracer

**BlueTracer** is a robust library and system call tracer for Windows programs specialized in evasive malware

Implementation details:

- Based on the **Intel Pin** DBI framework
- Integrated with the **BluePill** stealthy execution framework
- Combines reliable external sources of prototypes information

Key features:

- Undetected tracing of input parameters, output buffers and return values of over 17 000 system calls and library calls
- Logging of asynchronous events
- Resolution of named constants

# Why Intel Pin ?

Characteristics:

- User-friendliness
- Portability
- Transparency
- Efficiency



**Analysis routines:** embody the code to be inserted during the application's execution

**Instrumentation routines:** determine where the analysis code has to be placed

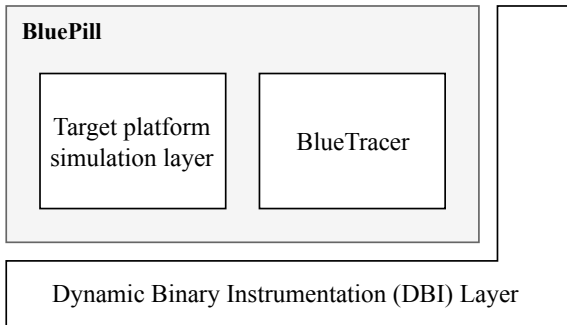
Different analysis and instrumentation granularities

- Instruction, trace, routine and image

# Integration with BluePill

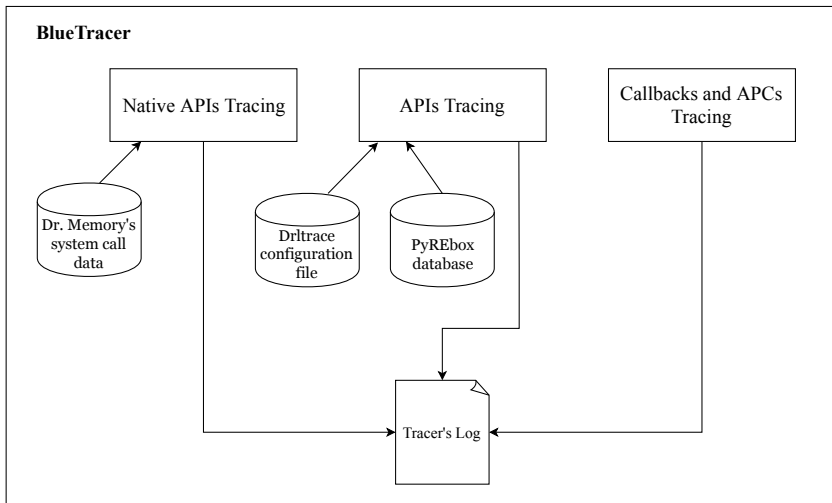
**BluePill** is a software toolkit which:

- Allows the simulation of a real production environment a specific malware sample was intended for
- Conceals any virtualization artifacts and software setup which might set off evasion





# BlueTracer's architecture



# Evaluation with AI-Khaser

**AI-Khaser** is an open-source application which performs common checks employed by malware families to determine if they are being executed in an analysis environment.

Checks divided in categories:

- **Anti-Debugging**
- **Timing-based**
- **Human Interaction Detection**
- **Anti-Virtualization**
- **Anti-Analysis**



BlueTracer was undetected with respect to all the checks!

# Example of tracked evasion check

File system artifacts can be checked in order to uncover the presence of a virtualized environment.

---

```
~~3160~~ 24980 KERNELBASE.dll!GetFileAttributesW
24980  arg 0: C:\Windows\system32\drivers\VBoxMouse.sys
        (name=lpFileName, type=wchar_t*, size=0x2)
24980      executed KERNELBASE.dll!GetFileAttributesW =>
24980  retval: 0xffffffff (name=Return value, type=DWORD, size=0x4)
```

---

# Evaluation with evasive malware samples

Five highly evasive samples collected by Joe Security:

ID	MD5	Name
1	0af4ef5069f47a371a0caf22ae2006a6	<i>trojan/banker</i>
2	9437eabf2fe5d32101e3fbf9f6027880	<i>dropper</i>
3	cbdda646a20d95f078393506ecdc0796	<i>trojan</i>
4	cfdd16225e67471f5ef54cab9b3a5558	Olympic
5	ef694b89ad7addb9a16bb6f26f1efaf7	CCleaner

The logs collected by BlueTracer reveal behaviors consistent with the analysis reports authored by Joe Security

# Conclusions

## Contribution:

Design and implementation of **BlueTracer**, a robust library and system call tracer for Windows programs specialized in evasive malware.

## Future Developments:

- Test with a larger set of highly evasive malware samples
- Improve logging capabilities
- Adopt log filtering techniques

**Thank you for your attention!**