Haow do I sandbox?!?!

Cuckoo Sandbox Internals

Jurriaan Bremer @skier_t

Student (University of Amsterdam), Freelance Security Researcher

June 22, 2013
Introduction - What this talk is NOT about!

Figure: Dragon Sandbox!
Introduction - What this talk is about!

- How we built Cuckoo
- How to evade Cuckoo
  - Left as an exercise for the attendee
  - Who would do such terrible thing though?
Introduction - Today's problems in Malware

- ... Insert long list of problems ...
- In the end, we prefer to blame..
Introduction - Today's problems in Malware

Java Setup - Complete

You have successfully installed Java and Ask Toolbar.

When Java updates are available, you will be prompted to download and install. We recommend you always do so to ensure your system has the latest performance and security improvements.

See the instructions if you want to change this behavior.

Close
Introduction - Today's problems in Malware Analysis

▶ Static Analysis takes a lot of time
  ▶ Obfuscation
  ▶ Packers

▶ Dynamic Analysis also takes a lot of time
  ▶ Multi-threaded malware
  ▶ Anti-debugger, anti-virtual machine, etc.
Introduction - Sandboxing in General (1)

- Enter Sandboxes
  - Automated Malware Analysis - handles all repetitive work
    - Process thousands of samples in a reasonable time
  - Generic methods for bypassing anti’s

- For the Client
  - User friendly - anyone can use it
  - Setup once, use it for eternity
    - For this step, see the manual :p
Introduction - Sandboxing in General (2)

- Existing Solutions
  - Closed Source
    - Not 100% customizable
  - Very expensive
- Enter Cuckoo Sandbox
  - Entirely Open Source
  - Free to use
Introduction - Disadvantages of Sandboxing

- Environment could be detected
  - Anti-sandbox
  - Randomize environment
    - Can only randomize so many things
- Various limitations depending on the implementation
  - We try our best to bypass these
    - E.g., Hook Detection by Malware
- Reports still have to be read by somebody
Cuckoo Sandbox Architecture

- New Job
- 1
- Cuckoo Scheduler
- 4
- Report

- $ python utils/submit.py sample.exe
- Performing the Analysis
  - Resume Virtual Machine
  - Send a zip file with sample & info
  - Run the malware in the VM
  - Suspend Virtual Machine
- Logs reported real-time to Host
- Generate a Report
Demonstration of analyzing a PDF exploit

- Demo showing the entire analysis process
- Quick look through the report
Inside the Virtual Machine - Agent

- Listening Agent
  - Accepts a connection
  - Host connects
  - Host sends zip file

- Agent unpacks zip file
  - Python code
    - Easily upgrade Cuckoo to a new version!
  - Configuration files
  - The sample

- Agent runs the Analyzer
  - Which has been sent through the zip
Inside the Virtual Machine - Analyzer

▶ Analyzer

▶ Initializes Cuckoo stuff
  ▶ Open IPC Channel (Named Pipe)
  ▶ Some handwaving etc

▶ Dumps Configuration for the first Process
  ▶ Name of the Named Pipe
  ▶ IP and Port of the Result Server
  ▶ (Will come back to that later)

▶ Runs the specified Package
Inside the Virtual Machine - Packages

- Package starts an application with commandline parameters
  - Wrappers around `CreateProcess(CREATE_SUSPENDED)`
    - Packages for EXE, DLL, PDF, DOC, etc.
  - Inject Cuckoo Monitor DLL into the process
    - Using APC, `QueueUserAPC(...)`
  - Resume main thread of the process
Inside the Application - Cuckoo Monitor

- When resuming the main thread
  - Cuckoo Monitor is executed first
    - Due to the APC callback
  - Initializes internals & installs API Hooks
  - Notifies the Analyzer
    - Through Named Pipes
  - Real application is started
Outside the Virtual Machine - Result Server

- Cuckoo Monitor logs directly to the Host, over TCP/IP
  - IP and Port retrieved from the Configuration
- More stability than before, when we logged to a local file
  - VM Crashes resulted in no logs
  - Now real-time results
So, what now?

- We’ve covered the basics
- Useful for single-process stuff
- What’s next?
More Advanced Malware (1)

- Some samples run new processes
  - RunPE, for Packers
  - Internet Exploder\(^*\) Explorer for URLs
- Some malware injects into other processes
  - Explorer.exe Injection to evade Firewalls
  - Banking Trojans
Child Process Injection

Before the new Process is executed, we want to inject Cuckoo Monitor.

- Cuckoo Monitor notifies Analyzer
  - Asks to be injected into the target process
  - Analyzer dumps configuration file
  - Injection using APC
Child Injection

Sample
Child Injection

Sample → CreateProcess(...) → Monitor

June 22, 2013
Jurriaan Bremer @skier_t
Haow do I sandbox?!?! Cuckoo Sandbox Internals 23 / 79
Child Injection

Sample

wind0ze

CreateProcess(...)

CREATE_SUSPENDED

Monitor
Child Injection

Sample

wind0ze

CreateProcess(...)

CREATE_SUSPENDED

“Okay, Bro.”

Monitor
Child Injection

CreateProcess(...) → CREATE_SUSPENDED → "Okay, Bro." → Inject Process (APC)

Sample → Monitor

wind0ze → Analyzer
Child Injection

Sample

wind0ze

Analyzer

Sample

CreateProcess(...) → Monitor

CREATE_SUSPENDED

“Okay, Bro.”

Inject Process (APC)

APC

June 22, 2013
Jurriaan Bremer @skier_t
Haow do I sandbox?!?!
Child Injection

Sample

wind0ze

Analyzer

Sample

CreateProcess(...)

CREATE_SUSPENDED

“Okay, Bro.”

Inject Process (APC)

Do it!

APC

Monitor

June 22, 2013
Jurriaan Bremer @skier_t
Haow do I sandbox?!?!
Cuckoo Sandbox Internals
Child Injection

- Sample
  - CreateProcess(...)
  - CREATE_SUSPENDED
  - "Okay, Bro."
  - Inject Process (APC)
  - Do it!
  - APC
  - ResumeThread
- Analyzer
- wind0ze
- Monitor
Child Injection

- Sample
- wind0ze
- Analyzer
- Sample
- Monitor

CreateProcess(...)

CREATE_SUSPENDED

“Okay, Bro.”

Inject Process (APC)

Do it!

APC

ResumeThread

Init Cuckoo Monitor
Process Injection

Before a sample injects and executes code into another process, we also want to inject Cuckoo Monitor.

Process Injection is similar to Child Injection, except for a few steps.

- No APC, but `CreateRemoteThread(...)`
  - Can't guarantee APC finishes in time
- Entirely inject Cuckoo Monitor before resuming execution
  - For Child Processes
Process Injection

Sample

OpenProcess(...) → Monitor

June 22, 2013

Jurriaan Bremer @skier.t

Haow do I sandbox?!?! Cuckoo Sandbox Internals
Process Injection

Sample

wind0ze

OpenProcess(...)

OpenProcess(...)

“Okay, Bro.”

Monitor
Process Injection

Sample

wind0ze

Analyzer

Monitor

OpenProcess(...)

OpenProcess(...)

“Okay, Bro.”

Inject Process (CreateRemoteThread)
Process Injection

Sample

wind0ze

Analyzer

Sample

OpenProcess(...)

OpenProcess(...)

Okay, Bro.

Inject Process (CreateRemoteThread)

CRT

Monitor
Process Injection

Sample

wind0ze

Analyzer

Monitor

OpenProcess(...)

OpenProcess(...)

“Okay, Bro.”

Inject Process (CreateRemoteThread)

Init Cuckoo Monitor

Sample

CRT
Process Injection

Sample

OpenProcess(...)

wind0ze

OpenProcess(...)

“Okay, Bro.”

Analyzer

Inject Process (CreateRemoteThread)

Monit}

Injected CRT

Init Cuckoo Monitor

Sample

Inject

June 22, 2013

Jurriaan Bremer @skier_t Haow do I sandbox?!?! Cuckoo Sandbox Internals 39 / 79
Process Injection

Sample

wind0ze

Analyzer

Inject Process (CreateRemoteThread)

Inject CRT

Init Cuckoo Monitor

OpenProcess(...) → Monitor

“Okay, Bro.”

Do it!

Sample

OpenProcess(...) → Sample

June 22, 2013
Jurriaan Bremer @skier_t
Haow do I sandbox?!?!
Process Injection

Sample

wind0ze

Analyzer

Inject CRT

Init Cuckoo Monitor

OpenProcess(...)

"Okay, Bro."

Inject Process (CreateRemoteThread)

Do it!

Continue

Monitor

Injected

June 22, 2013
Jurriaan Bremer @skier_t
Haow do I sandbox?!?!
Cuckoo Sandbox Internals
That said..

Figure: What the malware thinks it's doing.
That said..

Figure: What Cuckoo Sandbox thinks it’s doing.
That said..

Figure: What really happens.
API Hooking - Overview

- Core functionality of Cuckoo Monitor
- Cuckoo Monitor logs about 170 APIs
  - We’re adding APIs where needed
- Hooks lowest level APIs without losing context
  - Not CreateProcessA
  - Not CreateProcessW
  - Not CreateProcessInternalA
  - But CreateProcessInternalW
- However, we also hook higher-level APIs
  - ShellExecute (supports protocol handlers, URLs, ..)
  - system (can pipe multiple processes)
API Hooking - Trampolines (1)

- Redirect execution using trampolines
  - Create a trampoline
  - Patch the function

http://jbremer.org/x86-api-hooking-demystified/
API Hooking - Trampolines (2)

Figure: Trampolines are really basic.
API Hooking - Trampolines (3)

Figure: A day in the life of... a hooked API.
API Hooking - Avoiding Hook Recursion (1)

Figure: Hello Hook?

```
HOOKDEF(BOOL, WINAPI, WriteFile,
    _In_ HANDLE hFile,
    _In_ LPCVOID lpBuffer,
    _In_ DWORD nNumberOfBytesToWrite,
    _Out_opt_ LPDWORD lpNumberOfBytesWritten,
    _Inout_opt_ LPOVERLAPPED lpOverlapped
) {

    [...]  

    WriteFile(g_log_handle, "Hello Hook", 10, &bytes, NULL);

    [...]  
```

June 22, 2013 Jurriaan Bremer @skier_t Haow do I sandbox?!?! Cuckoo Sandbox Internals
The first hooked API call is interesting, ignore the others.
  Sounds easy enough right?

Around 170 hooks.
  Can’t add code to each hook.
    We’re not coding for our local University.

Solution: Transparently in the hooking mechanism.
API Hooking - Avoiding Hook Recursion (3)

- We need a counter
  - Zero -> execute the hook handler
  - Not Zero -> ignore this API call

- Let's go back to `WriteFile()`
  - `count = 0`
    - Increase counter
    - Execute the Hook Handler
  - `count = 1`
    - Ignore the Hook Handler
We need one counter per thread
- Thread Local Storage it is
Increase it before executing the hook handler
Decrease it after returning from the hook handler
- Oh, we have to run our code after the hook handler returns
- So we have to patch the return address
- Oh, we have to store the original return address temporarily
  - TLS to the rescue?

More on this later.
Thread-specific Error Value, equivalent to errno

Let’s assume CreateProcessInternalW() returns failure
  - However, logging the failure is successful
  - Great!

Last Error is stored in TLS as well

After calling the trampoline function, we copy the Last Error
  - (Right before execution goes back to the hook handler)
API Hooking - Get Last Error (2)

Figure: Example CreateProcessInternalW hook.
API Hooking - Get Last Error (3)

- We have to temporarily backup the Last Error
  - Until the function returns, where we restore it
- TLS anyone?
API Hooking - Special Hooks (1)

- What about our Advanced Persistent Hooks?
- Some hook handlers should always be executed
  - Special CreateProcessInternalW()
    - Somebody has to inject those system()'d processes
  - (The normal CreateProcessInternalW() only logs)
API Hooking - Special Hooks (2)

- Treated as another hook
  - Special hook hooks the target function first
    - Normal hook hooks the Special hooks’ hook (oboy)
  - Special hooks keeps its own data (Last Error, count, ... )
Please enter Brainfart mode now.

The following represents a `system()` hook as if it were the only hook.
void APT() {
    
    
    system(...);
    
    
}
API Hooking - Result

```c
void APT() {
    [...]
    system(...);
    [...]
}
```
API Hooking - Result

```c
void APT() {
    [...] 
    system(...);
    [...] 
}
```

![Diagram](image)

- If count = 0
- If count = 1
void APT() {
    [...]  
    system(...);
    [...]  
}
void APT() {
    [...] 
    system(...);
    [...] 
}

If count == 0
If count == 1

Cuckoo Sandbox Internals 63 / 79
void APT() {
    [...]
    system(...);
    [...]
}
API Hooking - Result

```c
void APT() {
    [...]  
    system(...);
    [...]  
}
```

```
MOV EDI, EDI
PUSH EBP
PUSH ESP
JMP 7635B17C
RUSH EBP
```

**count = 1**

```
JMP 656E017C
SUB ESP, 18
PUSH EDI
PUSH EDI
```

**system**

If count = 0
If count = 1

**jmp**
API Hooking - Result

```c
void APT() {
    [...] system(...);
    [...] count = 1
}
```

- If `count = 0`
- If `count = 1`

```
8BFF 55
8BEC E9 45B0C710
```

```
MOV EDI,EDI
PUSH EBP
MOV EBP,ESP
JMP 7635B17C
PUSH EBP
```

```
MOV EDI,EDI
PUSH EBP
PUSH EDI
JMP 656E017C
```

- `Pre-Backup`
- `TLS.retaddr = retaddr`
- `retaddr = Post-Backup`
- `jmp Hook Handler`
void APT() {
    [...] system(...);
    [...] count = 1
}

If count == 0
    If count == 1
        Pre-Backup
            count = 1
            TLS.repadr = retaddr
            retaddr = Post-Backup
            jmp Hook Handler

June 22, 2013 Jurriaan Bremer @skier_t

Haow do I sandbox?!?! Cuckoo Sandbox Internals
API Hooking - Result

```c
void APT() {
    [...]
    system(...);
    [...] 
}
```

Pre-Trampoline

TLS.lasterraddr = retaddr
retaddr = Post-Trampoline

```
8B FF
55
8B EC
E9 45 B0 C7 10
```

If count = 0
If count = 1

Pre-Backup

```c
count = 1
TLS.retaddr = retaddr
retaddr = Post-Backup
jmp Hook Handler
```

JMP 656E017C
SUB ESP, 18
PUSH ESI
PUSH EDI
JMP 7635B17C
RET
API Hooking - Result

```c
void APT() {
    [...]
    system(...);
    [...]
}
```

JMP 656E017C
SUB ESP, 18
PUSH ESI
PUSH EDI
[...]
RETN
count = 1

Pre-Trampoline

TLS.lastaddr = retaddr
retaddr = Post-Trampoline

Trampoline

If count = 0
If count = 1

Pre-Backup

jmp

count = 1
TLS.retaddr = retaddr
retaddr = Post-Backup
jmp Hook Handler
API Hooking - Result

```c
void APT() {
    [...]
    system(...);
    [...]
}
```

Pre-Trampoline

TLS.lasterraddr = retaddr
retaddr = Post-Trampoline

Trampoline

jmp

Pre-Backup

count = 1
TLS.retaddr = retaddr
retaddr = Post-Backup
jmp Hook Handler

June 22, 2013
Jurriaan Bremer @skier_t
Haow do I sandbox?!?! Cuckoo Sandbox Internals
API Hooking - Result

```c
void APT() {
    [...]
    system(...);
    [...]
}
```

Pre-Trampoline

TLS.lasterraddr = retaddr
retaddr = Post-Trampoline

Trampoline

count = 1

Post-Trampoline

retaddr = TLS.lasterrcnt
backup Last Error

count = 0
retaddr = TLS.retaddr
restore Last Error

Pre-Backup

If count = 0
If count = 1

jmp

Post-Backup

count = 1
TLS.retaddr = retaddr
retaddr = Post-Backup
jmp Hook Handler

June 22, 2013
Jurriaan Bremer @skier_t
Cuckoo Sandbox Internals
API Hooking - Result

```c
void APT() {
    [...] system(...);
    [...] system(...);
}
```

Pre-Trampoline

```c
TLS.lasterraddr = retaddr
retaddr = Post-Trampoline
```

Trampoline

```c
jmp
```

Post-Trampoline

```c
retaddr = TLS.lasterrcnt
backup Last Error
```

Post-Backup

```c
count = 0
retaddr = TLS.retaddr restore Last Error
```

If count = 0
If count = 1

Pre-Backup

```c
count = 1
TLS.retaddr = retaddr
retaddr = Post-Backup
jmp Hook Handler
```
Results

- What kind of logs are we interested in?
  - Process Management
  - Thread Management
  - Registry
  - File Input /Output
  - Sockets
  - ..

- Signatures & Reporting modules
Sometimes APIs are not relevant
  ▶ When injected into another process

Check Return Address in the Stack Trace
  ▶ Nothing interesting?
    ▶ Don’t log it

As usual, sounds easier than it is

Needs Taint Data
  ▶ One process can write to another process
Inter Process Communication required
  - VirtualAllocEx/VirtualFreeEx/.. go through the Analyzer
  - CreateRemoteThread(&LoadLibraryA, "evil.dll")
    - &LoadLibraryA is now interesting
We were testing this code earlier, but got generic Cuckoo errors.

- **Segfaults on NtClose/VirtualFreeEx**
  - Unrelated to this module
  - However, necessary
- Once fixed, should work.
Some malware checks against hooks for common functions.

```c
if(*(uint8_t *) addr == 0xe9) {
    ...
}
```

- **StubDLL doesn’t hook a function**
  - It generates a Shadow DLL in-memory
- **Trampolines for every exported function**
  - Restores context and jumps to original function
- **Prologue is not altered**
Work in Progress - StubDLL (2)

Figure: Simple old versus new system.

June 22, 2013 Jurriaan Bremer @skier_t Haow do I sandbox?!?! Cuckoo Sandbox Internals
Questions?

.. :)

June 22, 2013

Jurriaan Bremer @skier_t  
Haow do I sandbox?!?!  
Cuckoo Sandbox Internals