UNIVERSITY OF TWENTE.

Software Testing and Reverse Engineering
MALWARE ANALYSIS

0xcafebabe
Pham Duy Phuc (s1750550)

March 2016

Outline

1. Malware & Malware analysis
2. Static analysis
3. Dynamic analysis
4. Malware evasive techniques & solutions
5. Protocol RE
6. APK malware behaviors analysis

Malware

- Generally: Any code that “performs evil”
- Executable content with unknown functionality that is resident on a system of investigative interest
  - Viruses
  - Worms
  - Trojans
  - Spyware
  - Rootkits
  - Botnet
- Infection vectors: Exploiting vulnerable services, drive-by download and Social Engineering

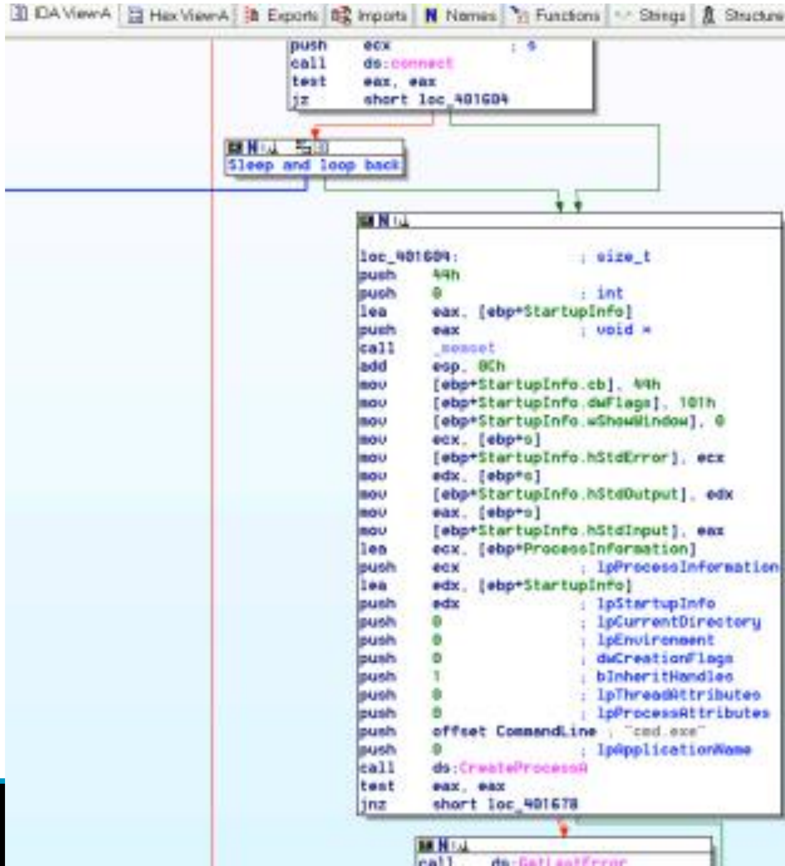
Malware analysis

Static malware analysis	Dynamic malware analysis
techniques that verify the actions the program performs in practice, **without actually executing it** - Disassembler & Decompilers	refers to techniques that execute functions, verify the actions the program performs in practice by **executing** it - Function hooking - Debugger

The Fastest Path to the Best Answers Will Usually Involve a Combination of Both.

Static malware analysis

- Safer
- File fingerprint, strings, metadata, resources
- Disassembly: Automated disassemblers can take machine code and “reverse” it to a slightly higher-level
- Decompiles
- Example [4] Metamorphism analysis paper



```
push    ecx             ; s
call    ds:connect
test    eax, eax
jz      short loc_401604
```

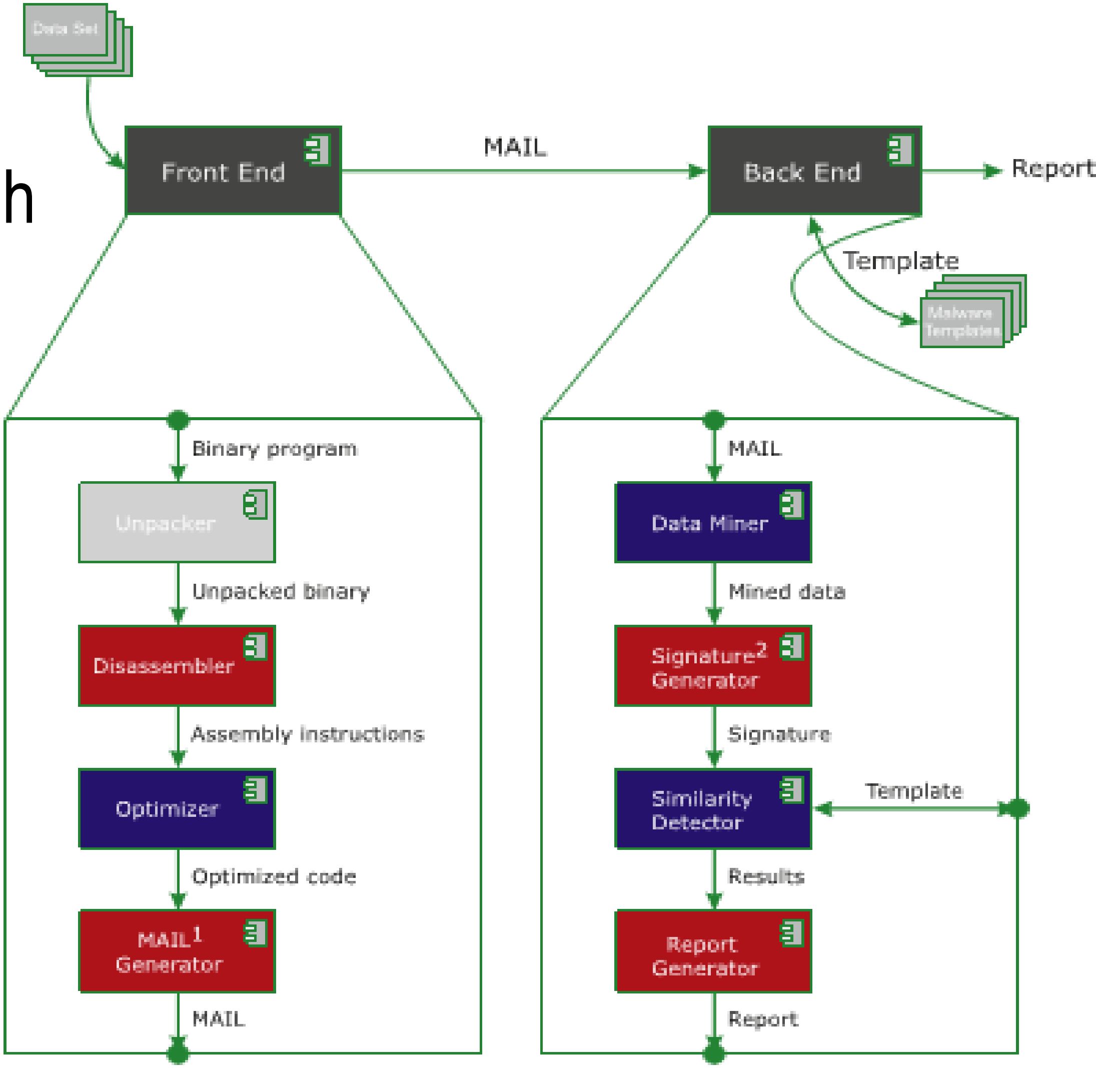
```
Sleep and loop back
```

```
loc_401604:                     ; size_t
push    44h
push    0                       ; int
lea     eax, [ebp+StartupInfo]
push    eax                     ; void *
call    _memset
add     esp, 0Ch
mov     [ebp+StartupInfo.cb], 44h
mov     [ebp+StartupInfo.dwFlags], 101h
mov     [ebp+StartupInfo.wShowWindow], 0
mov     ecx, [ebp+s]
mov     [ebp+StartupInfo.hStdError], ecx
mov     edx, [ebp+s]
mov     [ebp+StartupInfo.hStdOutput], edx
mov     eax, [ebp+s]
mov     [ebp+StartupInfo.hStdInput], eax
lea     ecx, [ebp+ProcessInformation]
push    ecx                     ; lpProcessInformation
lea     edx, [ebp+StartupInfo]
push    edx                     ; lpStartupInfo
push    0                       ; lpCurrentDirectory
push    0                       ; lpEnvironment
push    0                       ; dwCreationFlags
push    1                       ; bInheritHandles
push    0                       ; lpThreadAttributes
push    0                       ; lpProcessAttributes
push    offset CommandLine ; "cmd.exe"
push    0                       ; lpApplicationName
call    ds:CreateProcessA
test    eax, eax
jnz     short loc_401678
```

```
call    ds:GetLastError
```

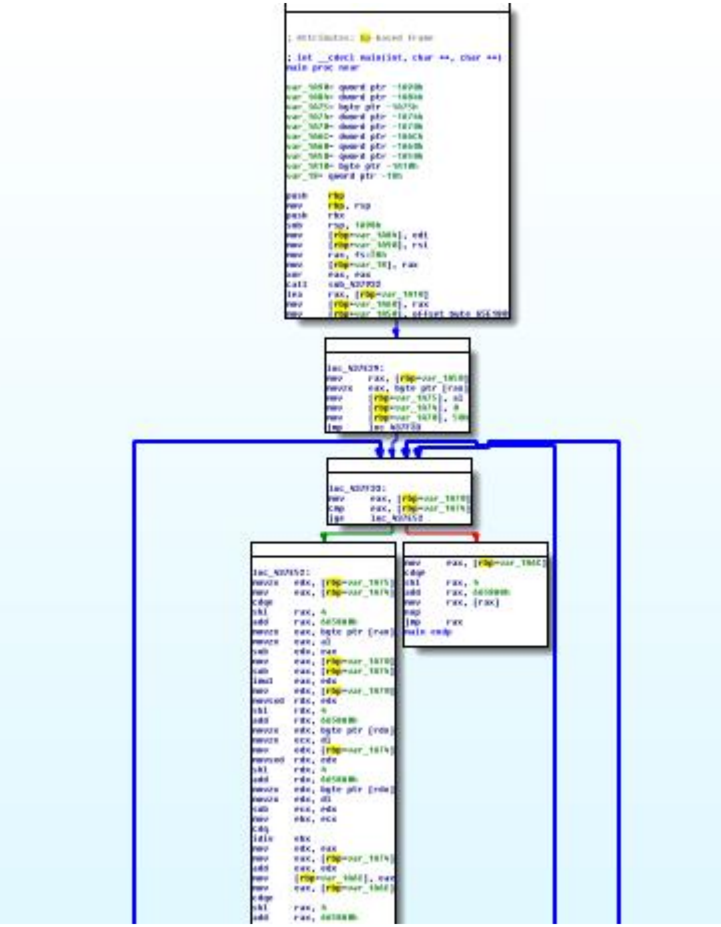
A framework for metamorphic malware analysis and real-time detection

- Annotated Control Flow Graph
- Sliding Window of Difference &Control Flow Weight using MAIL



MAIL = Malware Analysis Intermediate Language
In this version of the Malware Detector there are two types of signature generated:
ACFG (Annotated Control Flow Graph) and
SWOD-CFWeight (Sliding Window of Difference and Control Flow Weight)
The component "Unpacker" is not implemented in this version of the Malware Detector

A framework for metamorphic malware analysis and real-time detection



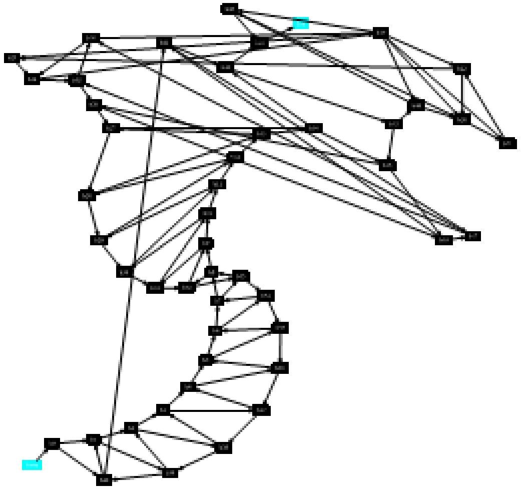
```
; Attributes: bp-based frame

; int __cdecl main(int, char **, char **)
main proc near

var_[illegible]= qword ptr -[illegible]
var_[illegible]= dword ptr -[illegible]
var_[illegible]= byte ptr -[illegible]
var_[illegible]= dword ptr -[illegible]
var_[illegible]= dword ptr -[illegible]
var_[illegible]= dword ptr -[illegible]
var_[illegible]= qword ptr -[illegible]
var_[illegible]= qword ptr -[illegible]
var_[illegible]= byte ptr -[illegible]
var_18= qword ptr -18h

push    rbp
mov     rbp, rsp
push    rbx
sub     rsp, [illegible]
mov     [rbp+var_[illegible]], edi
mov     [rbp+var_[illegible]], rsi
mov     rax, fs:[illegible]
mov     [rbp+var_18], rax
xor     eax, eax
call    sub_[illegible]
lea     rax, [rbp+var_[illegible]]
mov     [rbp+var_[illegible]], rax
mov     [rbp+var_[illegible]], offset byte_[illegible]
```

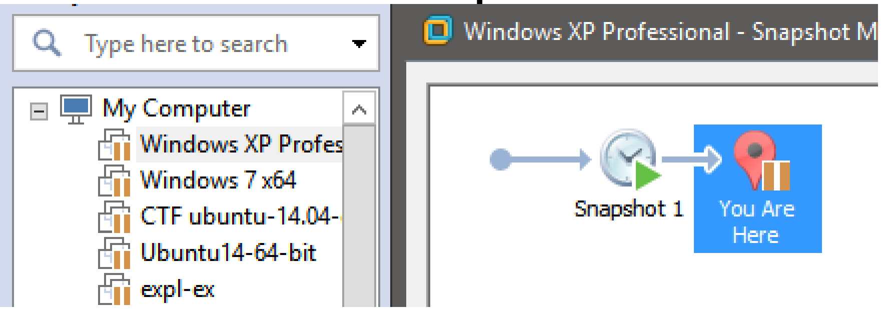
CFG

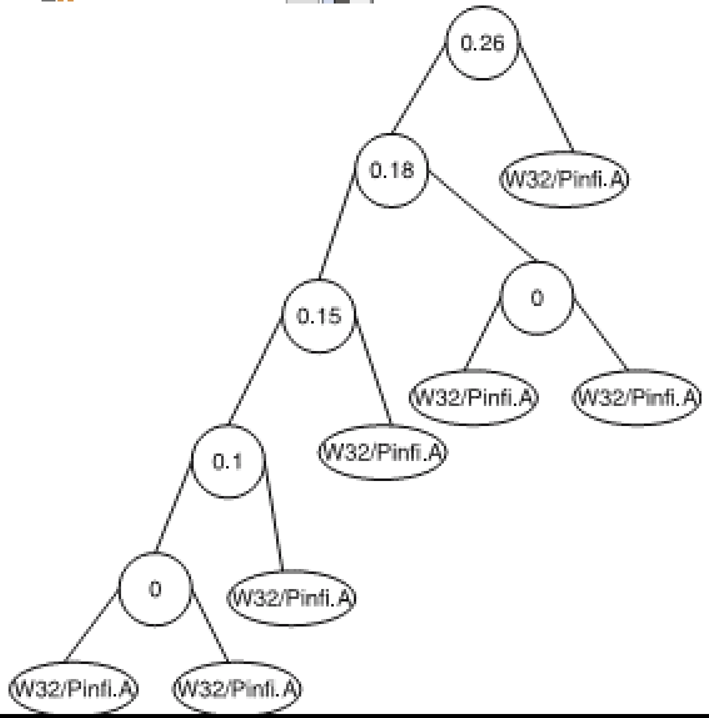


ACFG

Dynamic malware analysis

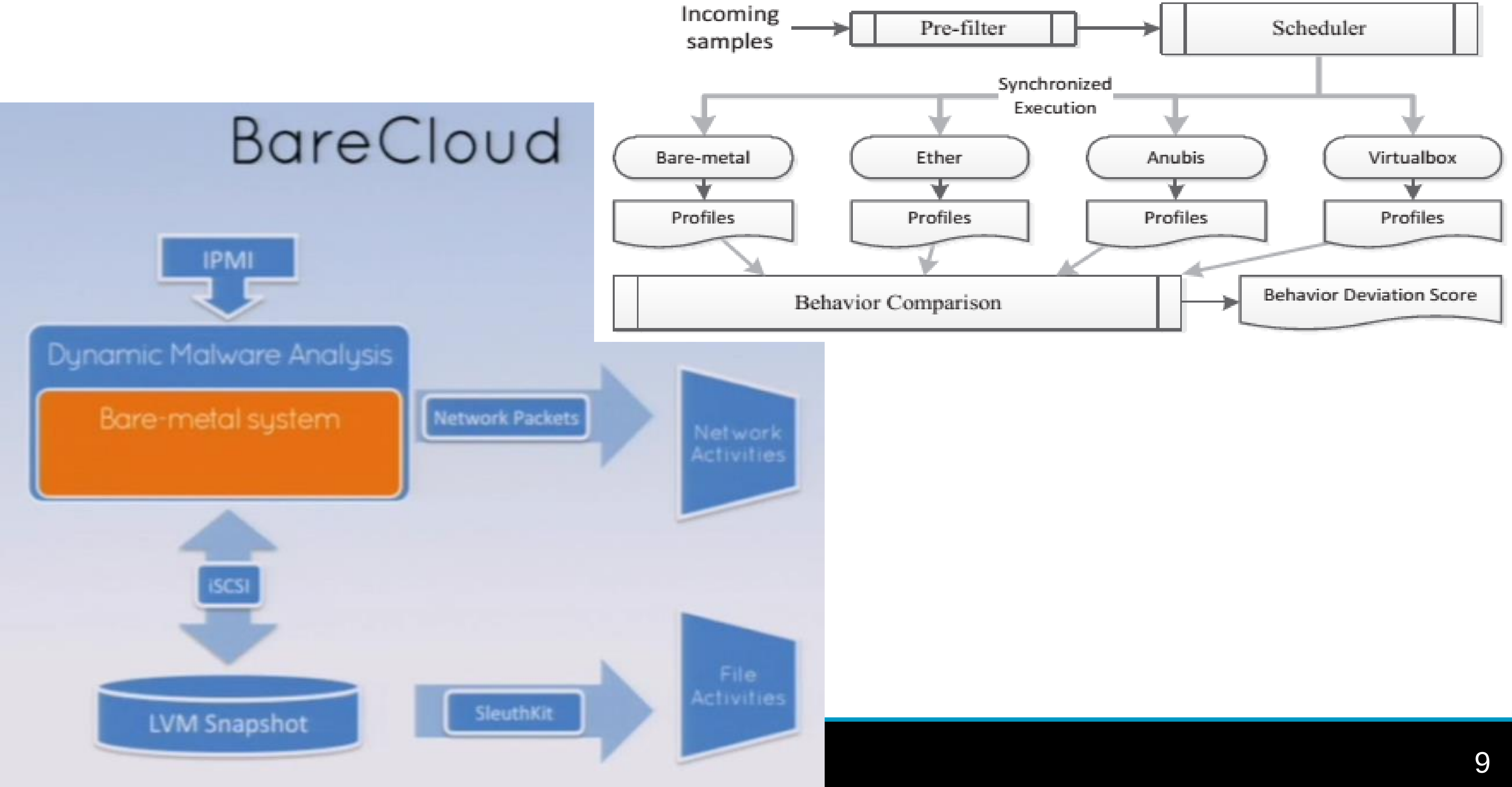
- Static malware analysis limitations
- Safe environment: Do Not Run Malware on Your Computer!
- Network simulation
- [3] Malware behaviour analysis
  - Malware behaviors: function calls
  - Malware behaviors similarity
  - Phylogenetic tree





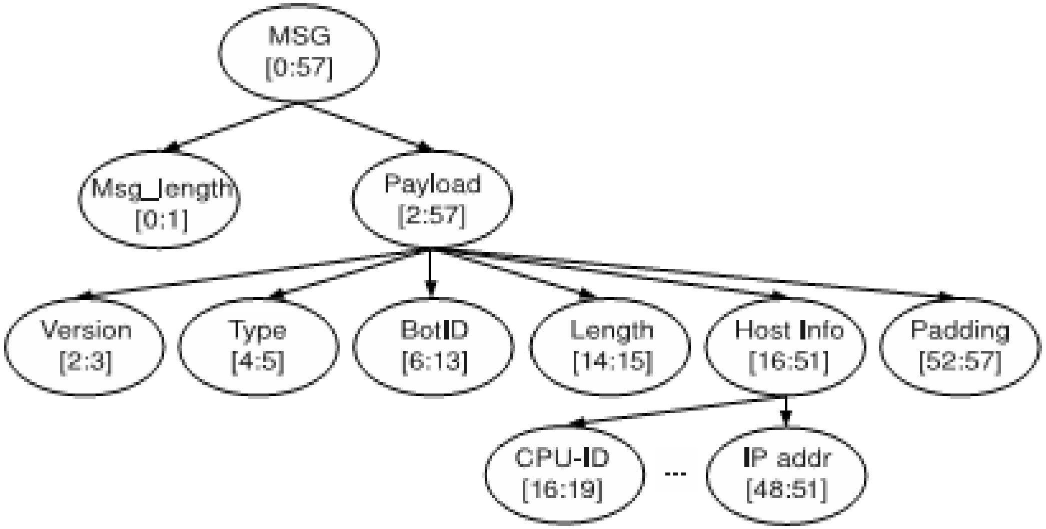
Malware evasive techniques & solutions

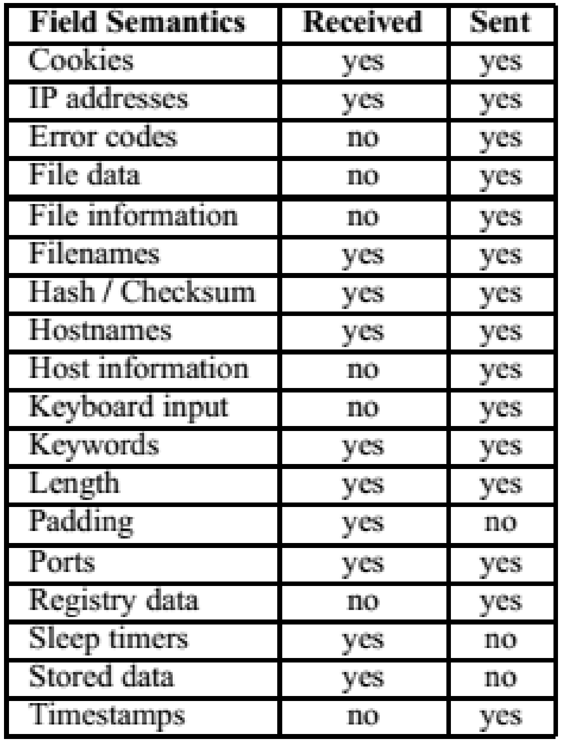
- Self-modifying code & analysis environment detection
- Disk, Bios, keyboard/mouse, UserID, CPU, CVE, timing attack, env vars
- Bare-metal Analysis-based Evasive Malware Detection [5]



Automatic protocol RE

- Dispatcher
- Field semantics inference
- Deconstruct the buffer based on program locations, dependency chains
- Determine the field attributes: keywords, length fields, delimiters, variable-length fields and arrays.

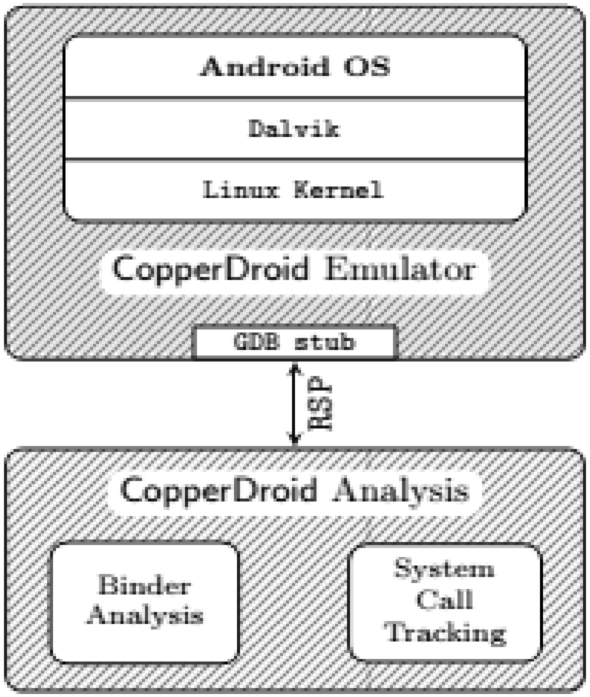




Field Semantics	Received	Sent
Cookies	yes	yes
IP addresses	yes	yes
Error codes	no	yes
File data	no	yes
File information	no	yes
Filenames	yes	yes
Hash / Checksum	yes	yes
Hostnames	yes	yes
Host information	no	yes
Keyboard input	no	yes
Keywords	yes	yes
Length	yes	yes
Padding	yes	no
Ports	yes	yes
Registry data	no	yes
Sleep timers	yes	no
Stored data	yes	no
Timestamps	no	yes

Automatically Reconstruct Android Malware Behaviors

- Tracking System Call Invocations
- Binder Analysis



References

- [1] Egele, Manuel, et al. "A survey on automated dynamic malware-analysis techniques and tools."
- [2] Caballero, Juan, et al. "Dispatcher: Enabling active botnet infiltration using automatic protocol reverse-engineering."
- [3] Wagener, Gérard, Radu State, and Alexandre Dulaunoy. "Malware behaviour analysis."
- [4] Shahid Alam, R.Nigel Horspool, Issa Traore, Ibrahim Sogukpinar. “A framework for metamorphic malware analysis and real-time detection”
- [5] Dhilung Kirat, Giovanni Vigna, and Christopher Kruegel. “BareCloud: Bare-metal Analysis-based Evasive Malware Detection
- [6] Alessandro Reina, Aristide Fattori, Lorenzo Cavallaro. “A System Call-Centric Analysis and Stimulation Technique to Automatically Reconstruct Android Malware Behaviors”. Security (EuroSec).