Malware Analysis Using Visualized Image Matrices

Tzu-Ming Huang

CISC850 Cyber Analytics



Overview

- malware visual analysis method
 - convert binary files into images
 - Reduce computation major block
 - similarity calculation method between these images



Method Overview

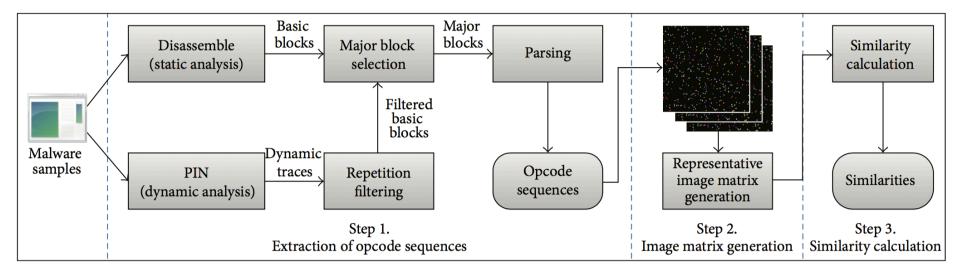


FIGURE 1: Overview of the proposed method.



Extract opcode sequences from binary

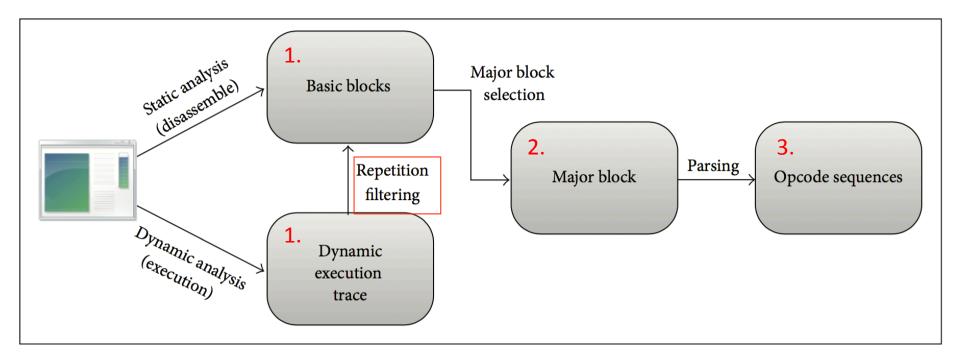


FIGURE 2: Opcode sequence extraction procedure.



Repetition Filtering

0042A68B	Main	JBE	SHORT Exploit0042A69C
0042A68D	Main	- MOV	AL, BYTE PTR DS: [EDX]
0042A68F	Main	INC	EDX
0042A690	Main	MOV	BYTE PTR DS: [EDI], AL
0042A692	Main	INC	EDI
0042A693	Main	DEC	ECX
0042A694	Main	JNZ	SHORT Exploit0042A68D
0042A68D	Main		AL, BYTE PTR DS: [EDX]
0042A68F	Main	INC	EDX
0042A690	Main	MOV	BYTE PTR DS:[EDI],AL
0042A692	Main	INC	EDI
0042A693	Main	DEC	ECX
0042A694	Main	\subset_{JNZ}	SHORT Exploit0042A68D
0042A68D	Main	⊂ MOV	AL, BYTE PTR DS: [EDX]
0042A68F	Main	INC	EDX
0042A690	Main	MOV	BYTE PTR DS: [EDI], AL
0042A692	Main	INC	EDI
0042A693	Main	DEC	ECX
0042A694	Main	\subset_{JNZ}	SHORT Exploit0042A68D



Extract opcode sequences from binary

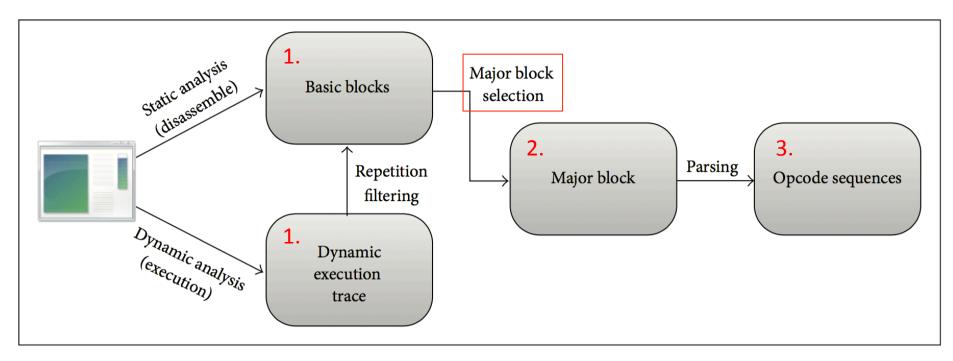


FIGURE 2: Opcode sequence extraction procedure.



Major Block Selection

- Not all of the basic blocks (file header, meaning less blocks)
- Target suspicious behavior
- Blocks include "CALL" instruction



Major Block Selection

MOV	Basic blocks [ebp + var_8], eax		
MOV MOV LEA MOV MOV	$[ebp + var_{-8}], eax$ $eax, [ebp + arg_{-8}]$ $[ebp + var_{-4}], eax$ $eax, [ebp + var_{-8}]$ $[ebx-4], eax$ $esi, [ebx + 0Ch]$ $edi, [ebx+8]$	PUS LEA PUS PUS	A ebp, [ebx + 10 h] SH 0FFFFFFFh SH ebx
PUSH LEA PUSH PUSH CALL ADD POP MOV	ebp ebp, [ebx + 10 h] 0FFFFFFFh ebx sub_4019EA esp, 8 ebp eax, 1	CAI ADI POF MO PUS LEA CAI	D esp, 8 P ebp V eax, 1 SH esi SH ebp A ebp, [ebx + 10h] LL dword ptr [edi + ecx $*$ 4 + 4]
PUSH PUSH LEA CALL POP POP MOV OR JZ	esi ebp ebp, [ebx + 10 h] dword ptr [edi + ecx * 4 + 4] ebp esi ebx, [ebp + arg_4] eax, eax short loc_40192F	POF POF MO OR JZ	P esi V ebx, [ebp + arg_4]
	÷		



Extract opcode sequences from binary

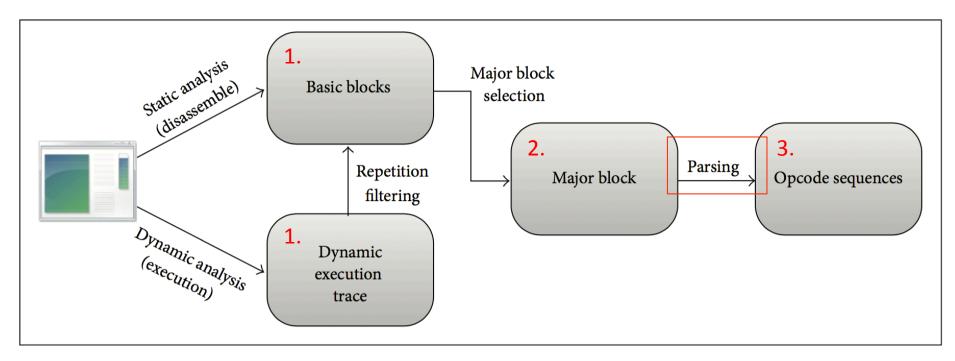


FIGURE 2: Opcode sequence extraction procedure.

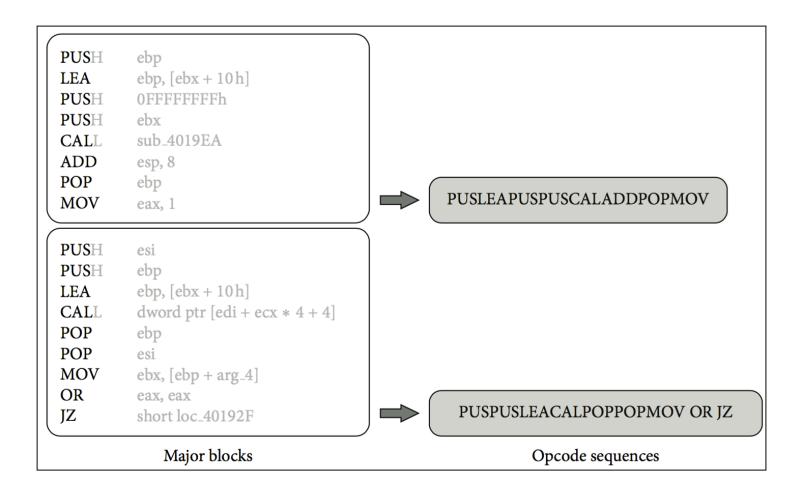


Parsing Opcode Sequence

- First three characters of opcode
 - 41.4% of opcodes have3 characters
 - Meaning is maintained
 - Eg. PUSH -> PUS; CALL -> CAL; OR?
- These three-character opcodes are concatenated together



Parsing Opcode Sequence



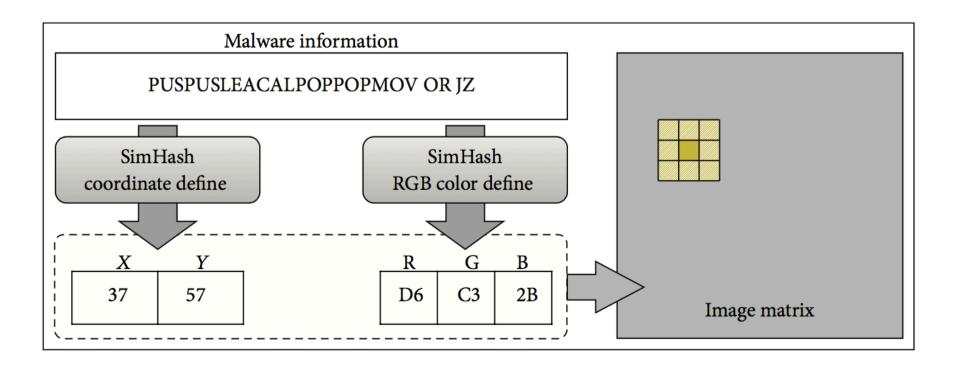


Generate Image Matrix

- Use hash function (*SimHash*) to decide X-Y coordinate and RGB colors of the pixels
- Length and width of matrix are 2ⁿ (8)
- If hash in same X-Y coordinate, simply sum the RGB colors value



Generate Image Matrix





Choose Representative Image Matrix

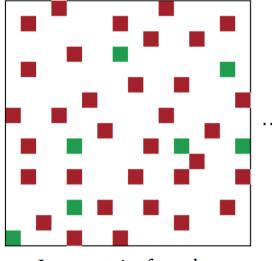


Image matrix of sample.a

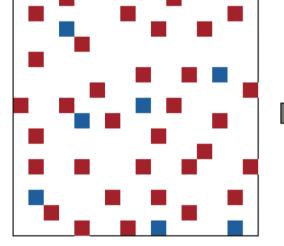
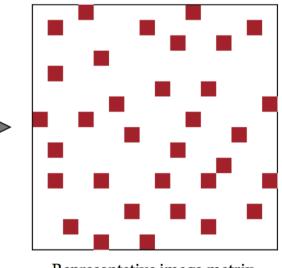
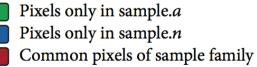


Image matrix of sample.*n*



Representative image matrix of sample family



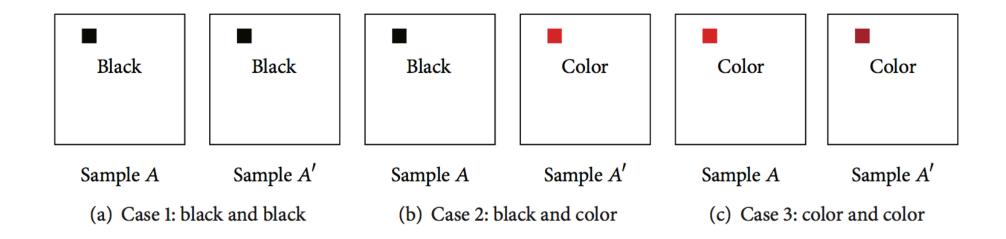


Similarity Calculation Using Image Matrix

- Faster performance than opcode string comparison
- Finding pairs in string: O(n²)
- Simhash and calculate similarity in image: O(n)



Similarity Calculation Using Image Matrix





Similarity Calculation Using Image Matrix

- vector angular-based distance measurement algorithm
 - Pixels are viewed as 3D vector

$$\delta\left(x_{i}, x_{j}\right) = \left[1 - \frac{2}{\pi}\cos^{-1}\left(\frac{x_{i} \cdot x_{j}}{\left|x_{i}\right| \left|x_{j}\right|}\right)\right] \left[1 - \frac{\left|x_{i} - x_{j}\right|}{\sqrt{3 \cdot 255^{2}}}\right]$$

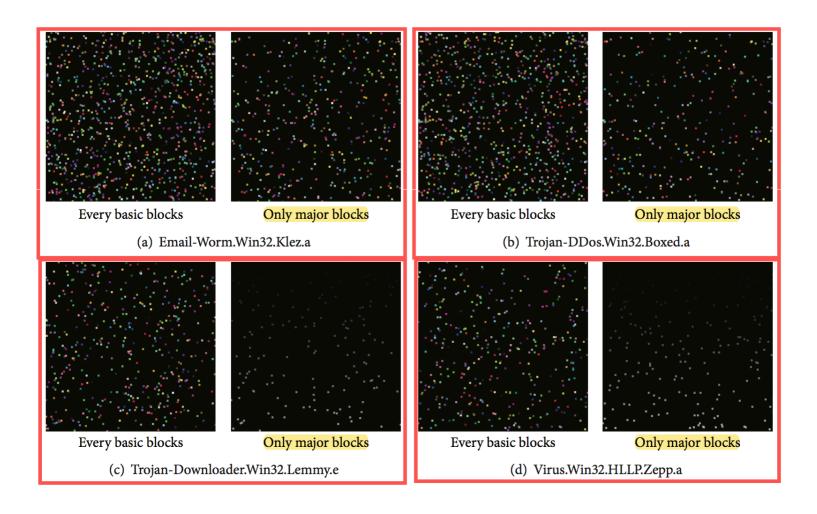


Similarity Calculation Using Image Matrix

 $Sim(A, B) = \frac{sum of pixel similarity values in case 3}{\# of pixels in case 2 and case 3}$

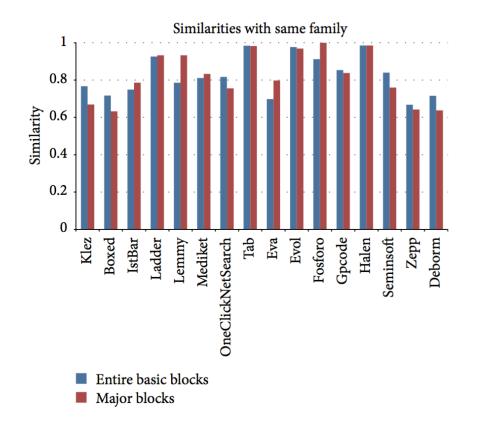


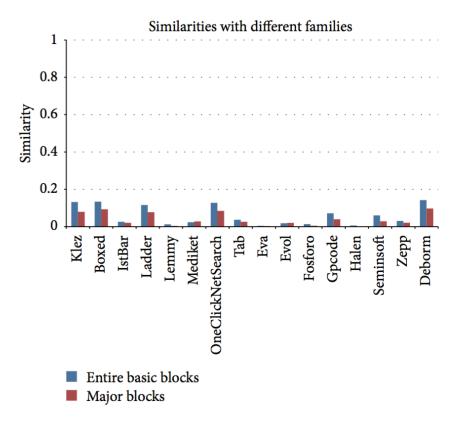
Experiment: Major Blocks Selection?





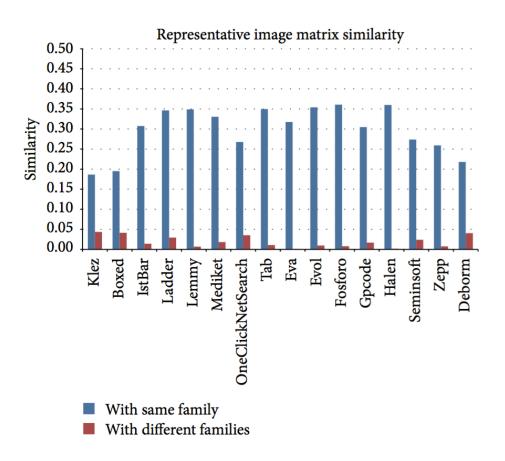
Experiment: Major Blocks Selection?







Experiment: Feasibility





Experiment: Feasibility

- Similarity of sample malwares from same family: 0.19 ~ 0.36
- Similarity of sample malwares from different family: < 0.05
- Classification accuracy = 0.9896