PoMMaDe: Pushdown Model-checking for Malware Detection

Fu Song and Tayssir Touili

http://sist.shanghaitech.edu.cn/faculty/songfu/Projects/PoMMaDe/

Motivation

- Malware can produce serious damage.
- The number of malware in 2010 is more than 1.5 billion.
- Existing antivirus techniques based on signature-matching and dynamic analysis are easy to get around.

$\Longrightarrow$ It is urgent to have efficient malware detectors.

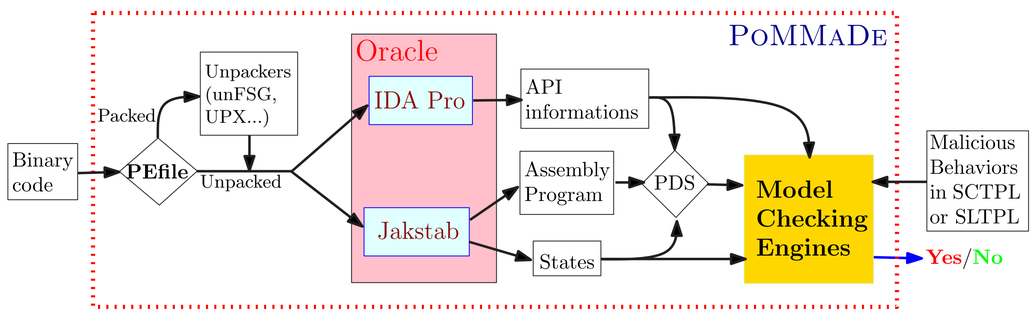
- Static-based techniques (model-checking) are efficient techniques for malware detection.
- However, existing static-based techniques cannot track the stack behavior nor specify behaviors over the stack content (needed for malware detection).
- Our solution: use PushDown Systems (PDS) to model binary codes, and use SCTPL/SLTPL to express malicious behaviors.

Our Tool: PoMMaDe

Given a binary code and a SCTPL or SLTPL property expressing a malicious behavior, our tool PoMMaDe can:

- output a PDS modeling the binary code for future analysis.
- check whether the binary code satisfies the property. If this is the case, PoMMaDe returns Yes, meaning that the binary code may be a malware. Otherwise, it returns No, i.e., the program is benign.

Implementation



- PEfile: gives the name of the packer if a binary code is packed
- Unpackers(unFSG, UPX, etc): unpack the binary code
- IDA Pro: outputs API functions' information and assembly programs
- Jakstab: performs static analysis, outputs assembly programs and values of registers at each control point [Kinder-Veith, 2008]

Approach

Binary Code $\models$ Malicious Behavior ?

$\downarrow$ $\downarrow$

Pushdown System $\models$ SCTPL/SLTPL ?

SCTPL=CTL+variables+$\{\forall,\exists\}$+predicates over the stack
SLTPL=LTL+variables+$\{\forall,\exists\}$+predicates over the stack

Model-Checking Engines

Pushdown System $\models$ SCTPL/SLTPL

$\downarrow$

Emptiness problem of (Symbolic) Alternating Büchi Pushdown System (SABPDS/SBPDS)

$\downarrow$

Finite Automaton (FA)

recognizing potential infinite set of configurations from which the SABPDS/SBPDS has an accepting run

Pushdown System $\models$ SCTPL/SLTPL $\Longleftrightarrow$ The FA is nonempty

[Song-Touili, TACAS'12,FM'12]

Example: the E-mail worm NetSky

l_1: push a
l_2: push 0
l_3: call GetModuleFileNameA
l_4: push a
l_5: call CopyFileA

Fragment of NetSky

Behavior: the worm copies itself to other locations. To do this, it calls the API function GetModuleFileNameA with 0 and an address a as parameters. After this, the file name of its own executable will be stored in the address a. Then, the API function CopyFileA is called with a as parameter (i.e., its own file name). This copies its file into other locations. We can specify this behavior in SCTPL as follows:

$$\mathbf{EF}\exists a(call(GetModuleFileNameA) \wedge 0a\Gamma^* \wedge \mathbf{EF}(call(CopyFileA) \wedge a\Gamma^*))$$

$0a\Gamma^*$ (resp. $a\Gamma^*$) is a predicate stating that the top of the stack are 0 and a (resp. a). The above formula states that there exists a path in which GetModuleFileNameA is called with 0 and some address a as parameter (i.e., 0 and a are on the top of the stack), later CopyFileA is called with a as parameter.

Experiments

Detection of Real Malwares

- Several hundreds of real malwares and 27 benign programs taken from Microsoft Windows XP system.
- Our tool PoMMaDe can detect all these malwares and prove that these benign programs are benign.
- Some results of malware detection are shown in the following table.

	Example	#LOC	SLTPL			SCTPL		
			Time	Memory	Result	Time	Memory	Result
Virus	Akez	264	13.78	59.02	Yes	14.75	15.59	Yes
	Alcaul.b	904	9.79	37.40	Yes	26.25	1.08	Yes
	Alcaul.c	347	2.05	9.40	Yes	26.52	2.45	Yes
	Alcaul.d	837	0.24	0.17	Yes	23.52	20.39	Yes
	Alcaul.e	907	2.20	2.76	Yes	39.26	0.94	Yes
	Alcaul.f	84	0.98	4.35	Yes	18.57	0.98	Yes
Email-worm	Ardurk.d	1497	3.22	10.54	Yes	51.50	7.67	Yes
	Atak.b	4480	21.17	24.92	Yes	42.28	15.50	Yes
	Atak.l	1902	1.75	5.99	Yes	21.70	3.75	Yes
	Predec.f	2813	8.44	47.21	Yes	51.59	10.40	Yes
	Predec.h	2645	9.68	56.00	Yes	54.62	12.26	Yes
	Predec.j	2818	9.81	57.71	Yes	54.17	12.83	Yes
	Netsky.gen	5496	10.37	14.98	Yes	56.10	11.67	Yes
	Netsky.k	6117	35.28	58.84	Yes	68.32	90.13	Yes
	Netsky.p	6004	35.88	46.37	Yes	68.18	79.96	Yes
	Kirbster	1261	948.52	1383.02	Yes		o.o.m.	
	Krynos.b	18357	987.22	947.92	Yes		o.o.m.	
	Newapt.B	11703	1120.21	1042.74	Yes		o.o.m.	
	Newapt.F	11771	1045.17	908.35	Yes		o.o.m.	
	Newapt.E	11717	1059.45	970.27	Yes		o.o.m.	
	Mydoom.j	22335	89.66	40.15	Yes	200.41	48.17	Yes
	Mydoom.v	5960	10.78	19.03	Yes	66.34	16.49	Yes
	Mydoom.y	26902	66.77	36.60	Yes	90.00	43.19	Yes
	Klez.e	15008	48.87	47.07	Yes	60.87	59.47	Yes
	Klez.i	15357	50.52	48.37	Yes	69.36	62.38	Yes
	Klez.j	15006	48.40	47.07	Yes	60.86	59.47	Yes
Trojan	LdPinch.aar	1245	32.03	198.88	Yes	1.66	8.47	Yes
	LdPinch.aoq	7688	46.29	234.86	Yes	7.33	10.13	Yes
	LdPinch.mj	5952	39.07	199.28	Yes	5.74	8.90	Yes
	LdPinch.ld	6609	8.37	13.36	Yes	5.41	4.24	Yes
Benign	Cmd.exe	35887	109.81	20.00	No		o.o.m.	
	Find.exe	936	13.44	201.22	No	14.42	601.58	No
	Notepad.exe	6943	670.04	451.61	No		o.o.m.	
	Ping.exe	1842	8.53	31.67	No	77.98	245.77	No
	Print.exe	862	6.75	20.98	No		o.o.m.	
	Shutdown.exe	2524	31.69	62.93	No		o.o.m.	
	Regedt.exe	60	0.02	0.02	No	10.62	0.03	Yes
	Java.exe	21868	184.58	27.96	No	78.64	238.77	Yes

Comparison with existing anti-viruses.

- 200 new malwares generated by NGVCK and VCL32, respectively. NGVCK and VCL32 are the best malware generators.
- Our tool PoMMaDe can detect all these new malwares.
- Several well-known and widely used anti-viruses were not able to detect several of them.

Generator	No. of Variants	PoMMaDe	Avira	Kaspersky	Avast	Qihoo 360	McAfee	AVG	BitDefender	Eset Nod32	F-Secure	Norton	Panda	Trend Micro
NGVCK	100	100%	0%	23%	18%	68%	100%	11%	97%	81%	0%	46%	0%	0%
VCL32	100	100%	0%	2%	100%	99%	0%	100%	100%	76%	0%	30%	0%	0%