Centro de
Investigación en
Computación

Dynamic Analysis

Análisis Dinámico

Técnicas y herramientas

Course

Análisis y Detección de Malware

Instructor

Acosta Bermejo Raúl

Lecture notes

Instituto
Politécnico
Nacional

Table of contents (outline)

Tabla de contenido

Dynamic Analysis

Definición

- **Dynamic program analysis** is the analysis of computer software that is performed by **executing programs** on a real or virtual processor.
- For dynamic program analysis to be effective, the target program must be executed with sufficient test inputs to produce interesting behavior.
- Use of *software testing measures* such as code coverage helps ensure that an adequate slice of the program's set of possible behaviors has been observed.
- Also, care must be taken to minimize the effect that instrumentation has on the execution (including temporal properties) of the target program.

Link

- https://en.wikipedia.org/wiki/Dynamic_program_analysis

Recolección de datos

Tiempo real

Técnicas y herramientas

Recolección de datos

Introducción

Consumo de recursos

1. CPU
   Tiempo o porcentaje de uso.
2. Memoria
   Cantidad de memoria.
3. Archivos
   Ubicación de los archivos y sus operaciones.
   El uso de datos estructurados: BD o registros del sistema.
4. E/S
   Puertos usados.
5. Comunicaciones
   Sockets: UNIX/Inet, Datagrams (UDP)/Streams(TCP)

Recolección de datos

System Call / Invocaciones al SO

Syscall

1. En el binario
   Es difícil saber de manera estática cuales son los syscalls invocados.
   Pbms: depende del SO y sus versiones.
2. Comando
   $ strace /bin/ls
   access("/etc/ld.so.preload", R_OK) = -1 ENOENT
   syscall parametros valor de retorno
   $ strace –p PID
   $ strace –c /bin/ls

   Lista un resumen de los syscalls: cantidad, porcentaje, tiempo.

Recolección de datos

System Call / Invocaciones al SO

% time	seconds	usecs/call	calls	errors	syscall
0.00	0.000000	0	7		read
0.00	0.000000	0	2		write
0.00	0.000000	0	11		close
0.00	0.000000	0	10		fstat
0.00	0.000000	0	27		mmap
0.00	0.000000	0	8		mprotect
0.00	0.000000	0	1		munmap
0.00	0.000000	0	3		brk
0.00	0.000000	0	2		rt_sigaction
0.00	0.000000	0	1		rt_sigprocmask
0.00	0.000000	0	2		ioctl
0.00	0.000000	0	8		pread64
0.00	0.000000	0	2	2	access
0.00	0.000000	0	1		execve
0.00	0.000000	0	2	2	statfs
0.00	0.000000	0	2	1	arch_prctl
0.00	0.000000	0	2		getdents64
0.00	0.000000	0	1		set_tid_address
0.00	0.000000	0	9		openat
0.00	0.000000	0	1		set_robust_list
0.00	0.000000	0	1		prlimit64
100.00	0.000000		103	5	total

Recolección de datos

System Call / Invocaciones al SO

El análisis se centra en:

1. **Categorizar** las syscalls
   Por ejemplo: Time, filesystem, process memory, communication, etc.

   Son necesarias todas las syscalls?

2. **Simplificar** su procesamiento
   Cada syscall recibe parámetros mediante el uso de la pila: push, pop.
   Aquí se tiene bytes de información posiblemente importantes.

   Son necesarios los parámetros?

Recolección de datos

System Call / Invocaciones al SO

Memoria

1. Por proceso

   $ ps -aux –q PID

   ```
   USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
   ronin       PID  0.0  0.0  12024  3400 pts/2    Ss   13:28   0:00 bash -c cd
   /home/ronin/Develop/coach/webapp; bash --login
   ```

2. Comando especialziados

   $ pmap PID

   308124: bash -c cd /home/ronin/Develop/coach/webapp; bash --login

   ```
   000056344e118000    180K.   r----    bash
   000056344e23a000     40K    rw---  [ anon ]
   00007f8832e21000    8100K   r----    locale-archive
   00007f883360d000     136K   r----    libc-2.31.so
   00007f8833805000     56K    r----    libtinfo.so.6.2
   00007f8833848000     28K    r--s-     gconv-modules.cache
   00007f883384f000     4K     r----    ld-2.31.so
   00007ffd6a428000     132K   rw---  [ stack ]
   total          12028K
   ```

The end

Contacto

Raúl Acosta Bermejo

http:www.cic.ipn.mx
http://www.ciseg.cic.ipn.mx/

racostab@ipn.mx
racosta@cic.ipn.mx

57-29-60-00
Ext. 56652