

Tsurugi Linux入門 のWriteup

Tsurugi Linuxを利用したForensic技術を学ぶ

Presented by m1llcr3pe

Challenge1-WinRegistry

目的	Windowsのレジストリの解析 Windowsのレジストリファイルよりレジストリキーの内容を解析する
利用 コマンド	Name : reglookup - Windows NT+ registry reader/lookup tool Usage : reglookup [options] <registry-file>
実行結果	<pre>tsurugi@tsurugi-VirtualBox:~\$ sudo reglookup -v /home/tsurugi/Desktop/Tsurugi_Linux_Challenge/Challenge1-WinRegistry grep Planet /Microsoft/DirectDraw/Compatibility/ScorchedPlanet,KEY,,2009-07-14 04:37:08 /Microsoft/DirectDraw/Compatibility/ScorchedPlanet/Name,SZ,SPLANETW.EXE, /Microsoft/DirectDraw/Compatibility/ScorchedPlanet/ID,BINARY,i¥x04L2, /Microsoft/DirectDraw/Compatibility/ScorchedPlanet/Flags,BINARY,¥x02¥x00¥x00¥x00, /Microsoft/Windows/AVTOKYO2018/flag,SZ,HackThePlanet, INFO: Finished printing key tree.</pre>
FLAG	HackThePlanet

Challenge2-BrowsingHistory

目的	Webブラウザの実行履歴の解析 Internet Exploreのキャッシュファイルより実行履歴を解析する
利用 コマンド	Name : pasco - Tool to extract information from MS IE cache files Usage: pasco [options] <filename>
実行結果	<pre>tsurugi@tsurugi-VirtualBox:~\$ sudo pasco /home/tsurugi/Desktop/Tsurugi_Linux_Challenge/Challenge2-BrowsingHistory grep flag URL Visited: IEUser@http://www.avtokyo2018.com/flag/is/20181103 09/04/2018 15:23:13 09/04/2018 15:23:13 URL Visited: IEUser@http://www.avtokyo2018.com/flag/is/20181103 09/04/2018 15:23:13 09/04/2018 15:23:13</pre>
FLAG	20181103

Challenge3-DeletedFile.e01

目的	ディスクイメージの解析 ディスクのイメージファイルより削除されたファイルを解析する
利用 コマンド	Name : fls - List file and directory names in a disk image Usage: fls [-adDFIpruvV] [-m mnt] [-z zone] [-f fstype] [-s seconds] [-i imgtype] [-o imgoffset] [-b dev_sector_size] image [images] [inode]
実行結果	<pre>tsurugi@tsurugi-VirtualBox:~\$ sudo fls -d -r /home/tsurugi/Desktop/Tsurugi_Linux_Challenge/Challenge3-DeletedFile.e01 grep tsurugi r/r * 913139: var/tmp/tsurugiRFIR.swp r/r * 913140: var/tmp/tsurugiRFIS.swp r/r * 3880423: usr/share/man/avtokyo2018/flag/tsurugiDFIR</pre>
FLAG	tsurugiDFIR

Challenge4-WinMemClipBoard (1/2)

目的	メモリーイメージの解析 メモリのdumpファイルよりClipBoardに保存された内容を解析する
利用 コマンド	Name : Volatility Framework - Volatile memory extraction utility framework Usage: volatility -f <memory_image> <option>
実行結果	<pre>tsurugi@tsurugi-VirtualBox:~\$ volatility -f /home/tsurugi/Desktop/Tsurugi_Linux_Challenge/Challenge4-WinMemClipBoard imageinfo Volatility Foundation Volatility Framework 2.6.1 INFO : volatility.debug : Determining profile based on KDBG search... Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86_24000, Win7SP1x86 AS Layer1 : IA32PagedMemoryPae (Kernel AS) AS Layer2 : FileAddressSpace (/home/tsurugi/Desktop/Tsurugi_Linux_Challenge/Challenge4-WinMemClipBoard) PAE type : PAE DTB : 0x185000L KDBG : 0x82b43c30L Number of Processors : 1</pre> <p><中略></p>

Challenge4-WinMemClipboard (2/2)

実行結果

```
tsurugi@tsurugi-VirtualBox:~$ volatility -f /home/tsurugi/Desktop/Tsurugi_Linux_Challenge/Challenge4-WinMemClipboard --
profile=Win7SP1x86_23418 clipboard

Volatility Foundation Volatility Framework 2.6.1

Session  WindowStation Format          Handle Object  Data
-----
1 WinSta0  CF_UNICODETEXT  0x230173 0xfe5fda68 ZmxhZyBpcyB0c3VydWdp
1 WinSta0  0x0L           0x10 -----
1 WinSta0  0x3000L        0x0 -----
1 WinSta0  0x0L           0x3000 -----
1 -----  ----- 0x170067 0xfdfd1268

tsurugi@tsurugi-VirtualBox:~$
tsurugi@tsurugi-VirtualBox:~$ echo -n "ZmxhZyBpcyB0c3VydWdp" | base64 -d
flag is tsurugi
```

FLAG

tsurugi

Challenge5-UnknownExecBinary (1/4)

目的	<p>バイナリファイルの解析</p> <p>バイナリファイル(ELF32)を解析し、バイナリファイルの挙動を確認する</p>
利用 コマンド	<p>Name : radware2 - Advanced command-line hexadecimal editor, disassembler and debugger</p> <p>Usage: r2 [-ACdfLMnNqStuvwzX] [-P patch] [-p prj] [-a arch] [-b bits] [-i file] [-s addr] [-B baddr] [-m maddr] [-c cmd] [-e k=v] file pid - -- =</p>
実行結果	<pre>tsurugi@tsurugi-VirtualBox:~/Desktop/Tsurugi_Linux_Challenge\$ r2 Challenge5-UnknownExecBinary -- Please insert disc 2 and press any key to continue... [0x08048360]> aaaaa [x] Analyze all flags starting with sym. and entry0 (aa) [x] Analyze function calls (aac) [x] Analyze len bytes of instructions for references (aar) [x] Check for objc references [x] Check for vtables [x] Type matching analysis for all functions (aajt) [x] Propagate noreturn information [x] Use -AA or aaaa to perform additional experimental analysis. [x] Finding function preludes [x] Enable constraint types analysis for variables</pre>

Challenge5-UnknownExecBinary (2/4)

実行結果

```
[0x08048360]> afl
0x08048360  1 33      entry0
0x08048350  1 6       sym.imp.__libc_start_main
0x080483a0  4 43      sym.deregister_tm_clones
0x080483d0  4 53      sym.register_tm_clones
0x08048410  3 30      entry.fini0
0x08048430  4 43 -> 40  entry.init0
0x080485a0  1 2       sym.__libc_csu_fini
0x08048390  1 4       sym.__x86.get_pc_thunk.bx
0x080485a4  1 20      sym._fini
0x08048530  4 97      sym.__libc_csu_init
0x0804845b  1 85      sym.weird
0x08048320  1 6       sym.imp.strcpy
0x08048310  1 6       sym.imp.strcmp
0x080484b0  5 127     main
0x080482d4  3 35      sym._init
0x08048340  1 6       loc.imp.__gmon_start
0x08048330  1 6       sym.imp.puts
[0x08048360]>
[0x08048360]> s sym.weird
```


Challenge5-UnknownExecBinary (3/4)

実行結果

```
[0x0804845b]> pdf
; CALL XREF from main @ 0x80484f2
└─ 85: sym.weird (char *src);
|     ; var char *s2 @ ebp-0x90
|     ; var int32_t var_8ch @ ebp-0x8c
|     ; var char *dest @ ebp-0x88
|     ; arg char *src @ ebp+0x8
|     0x0804845b   55          push ebp
|     0x0804845c   89e5          mov ebp, esp
|     0x0804845e   81ec98000000  sub esp, 0x98
|                                     <中略>
|     0x08048479   c78570ffffff. mov dword [s2], 0x4f545641 ; 'AVTO'
|     0x08048483   c78574ffffff. mov dword [var_8ch], 0x4f594b ; 'KYO'
|     0x0804848d   83ec08        sub esp, 8
|                                     <中略>
|     0x080484ae   c9           leave
└─ 0x080484af   c3           ret
[0x0804845b]>
[0x0804845b]> q
```

Challenge5-UnknownExecBinary (4/4)

実行結果

```
tsurugi@tsurugi-VirtualBox:~/Desktop/Tsurugi_Linux_Challenge$ ./Challenge5-UnknownExecBinary AVTOKYO
```

```
::: TSURUGI BINARY CTF 2018 :::
```

```
Crack me! ;)
```

```
Success!
```

FLAG

AVTOKYO

謝辞

最後までご確認いただき、
有難うございました。