Radare2 and Cutter:
Primeros pasos con ejemplos

Hugo González

22 de junio de 2018

UNIVERSIDAD POLITÉCNICA DE SAN LUIS POTOSÍ

BSIDES CDMX

Sobre mi

- Hugo González
- Profesor-investigador UPSLP
- The Honeynet Project
    - bitly.com/hugoglez
    - linkedin.com/in/hugoxglez
    - github.com/hugo-glez/
    - @hugo_glez
    - hugo.gonzalez(at)upslp.edu.mx

UNIVERSIDAD
POLITÉCNICA
DE SAN LUIS POTOSÍ

Agenda

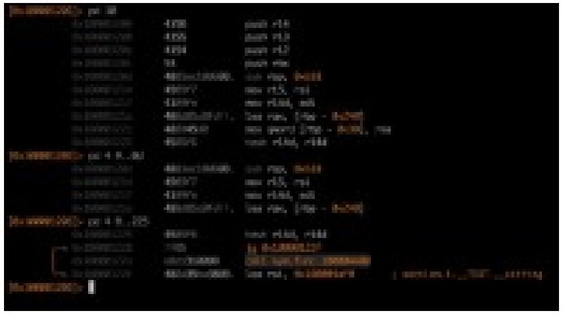
- Breve introducción a radare
- Breve introducción a cutter
  - Instalación
  - Vistazo a cutter
- Crackme 1
- Crackme 2
- Mirai botnet
- Conclusiones finales

UNIVERSIDAD POLITÉCNICA DE SAN LUIS POTOSÍ

Radare2

Radare is a LGPL portable reversing framework that can:

- Disassemble (and assemble for) many different architectures
- Debug natively or use remote targets (gdb, r2pipe, winedbg, windbg)
- Run on Linux, *BSD, Windows, OSX, Android, iOS, Solaris and Haiku
- Perform forensics on filesystems and data carving
- Be scripted in Python, Javascript, Go and more
- Support collaborative analysis using the embedded webserver
- Visualize data structures of several file types
- Patch programs to uncover new features or fix vulnerabilities
- Use powerful analysis capabilities to speed up reversing
- Aid in software exploitation

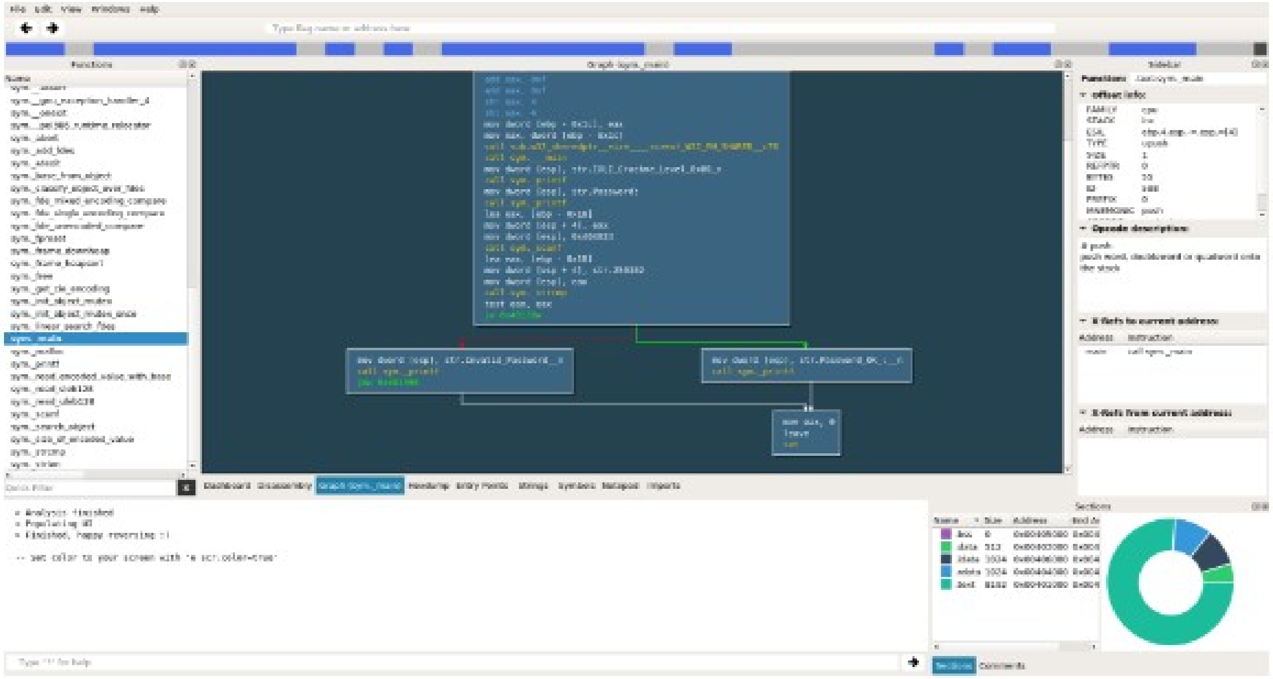


You can start by reading its documentation, check the community, see how it compares to others, get some swag and of course, download it.

Radare2

- Open Source
- Portable
- Versatil
- Activo
- Mas de 10 años de desarrollo
- **Linea de comando**

Cutter



```
mov dword [ebp - 0x1c], eax
mov eax, dword [ebp - 0x1c]
call sym.__main
mov dword [esp], str.IOLI_Crackme_Level_0x00_n
call sym.printf
mov dword [esp], str.Password:
call sym.printf
lea eax, [ebp - 0x18]
mov dword [esp + 4], eax
mov dword [esp], 0x404033
call sym.scanf
lea eax, [ebp - 0x18]
mov dword [esp + 4], str.250382
mov dword [esp], eax
call sym.strcmp
test eax, eax
```

```
mov dword [esp], str.Invalid_Password__n
call sym.printf
```

```
mov dword [esp], str.Password_OK_:)_n
call sym.printf
```

```
mov eax, 0
leave
ret
```

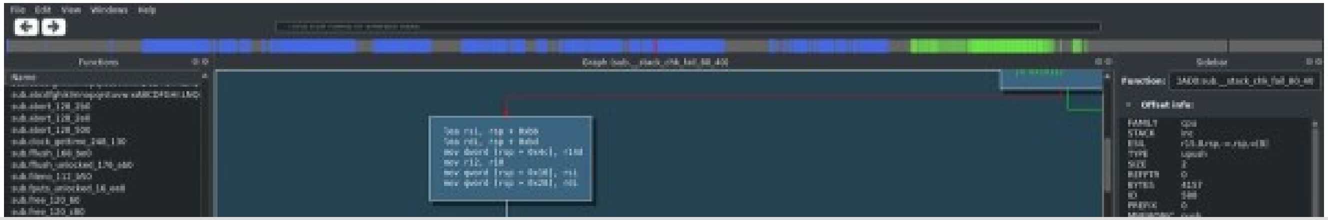
```
- Analysis finished
- Populating UI
- Finished, happy reversing :)

-- Set color to your screen with 'e scr.color=true'
```

Cutter is Qt and C++ GUI for radare2, originally named Iaito. It is not aimed at existing radare2 users, but focuses on those whose aren't fluent yet with the command line, likely because of the steep learning curve.

You can download the latest release here, or build it from source

The best place to obtain help from cutter developers and community is to join the telegram group, or the irc channel.

It even comes with a dark theme:



Cutter

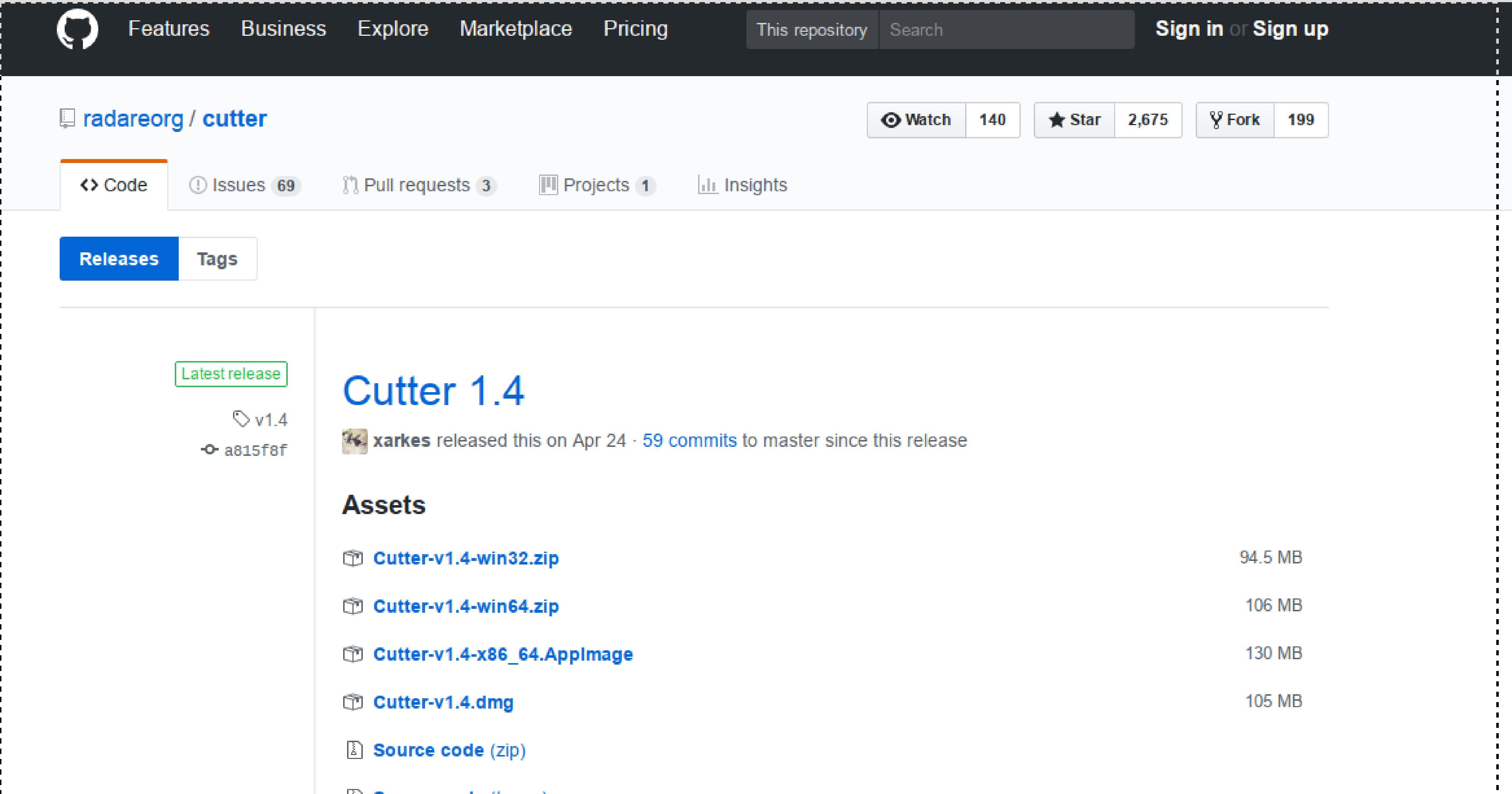
- GUI en QT para radare2 (point and click)
- Reduce el “impacto” para los nuevos usuarios
- Permite realizar scripts en python

Pregunta:

¿ Para que nos va a servir Cutter ?

Instalación

- http://github.com/radareorg/cutter/



radareorg / cutter

Watch 140 · Star 2,675 · Fork 199

Code · Issues 69 · Pull requests 3 · Projects 1 · Insights

Releases · Tags

Latest release

v1.4

a815f8f

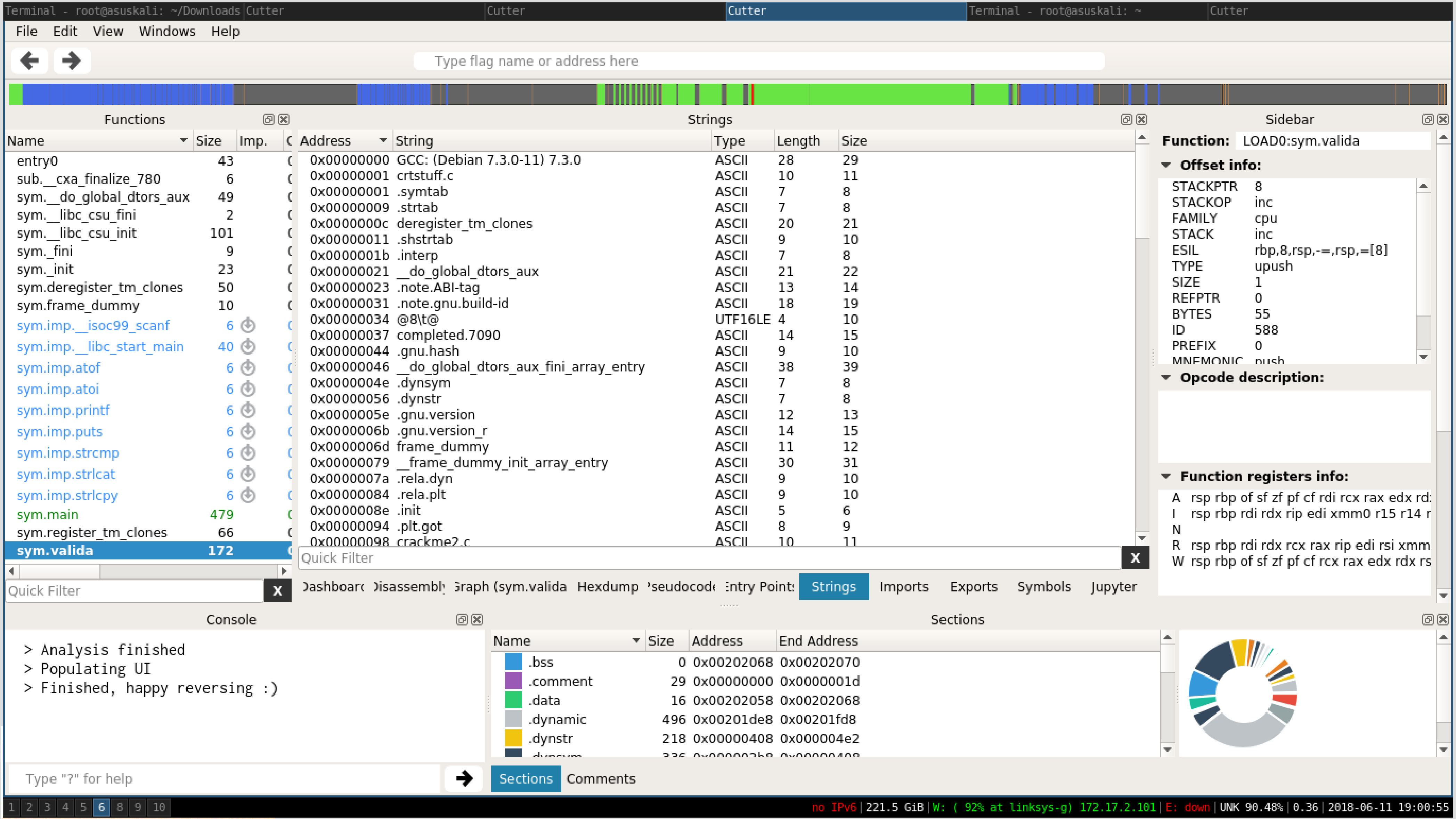
Cutter 1.4

xarkes released this on Apr 24 · 59 commits to master since this release

Assets

Cutter-v1.4-win32.zip	94.5 MB
Cutter-v1.4-win64.zip	106 MB
Cutter-v1.4-x86_64.AppImage	130 MB
Cutter-v1.4.dmg	105 MB
Source code (zip)	

Vista a cutter



Functions

Name	Size	Imp.	C
entry0	43		(
sub.__cxa_finalize_780	6		(
sym.__do_global_dtors_aux	49		(
sym.__libc_csu_fini	2		(
sym.__libc_csu_init	101		(
sym._fini	9		(
sym._init	23		(
sym.deregister_tm_clones	50		(
sym.frame_dummy	10		(
sym.imp.__isoc99_scanf	6	⊕	(
sym.imp.__libc_start_main	40	⊕	(
sym.imp.atof	6	⊕	(
sym.imp.atoi	6	⊕	(
sym.imp.printf	6	⊕	(
sym.imp.puts	6	⊕	(
sym.imp.strcmp	6	⊕	(
sym.imp.strlcat	6	⊕	(
sym.imp.strlcpy	6	⊕	(
sym.main	479		(
sym.register_tm_clones	66		(
sym.valida	172		(

Strings

Address	String	Type	Length	Size
0x00000000	GCC: (Debian 7.3.0-11) 7.3.0	ASCII	28	29
0x00000001	crtstuff.c	ASCII	10	11
0x00000001	.symtab	ASCII	7	8
0x00000009	.strtab	ASCII	7	8
0x0000000c	deregister_tm_clones	ASCII	20	21
0x00000011	.shstrtab	ASCII	9	10
0x0000001b	.interp	ASCII	7	8
0x00000021	__do_global_dtors_aux	ASCII	21	22
0x00000023	.note.ABI-tag	ASCII	13	14
0x00000031	.note.gnu.build-id	ASCII	18	19
0x00000034	@8\t@	UTF16LE	4	10
0x00000037	completed.7090	ASCII	14	15
0x00000044	.gnu.hash	ASCII	9	10
0x00000046	__do_global_dtors_aux_fini_array_entry	ASCII	38	39
0x0000004e	.dynsym	ASCII	7	8
0x00000056	.dynstr	ASCII	7	8
0x0000005e	.gnu.version	ASCII	12	13
0x0000006b	.gnu.version_r	ASCII	14	15
0x0000006d	frame_dummy	ASCII	11	12
0x00000079	__frame_dummy_init_array_entry	ASCII	30	31
0x0000007a	.rela.dyn	ASCII	9	10
0x00000084	.rela.plt	ASCII	9	10
0x0000008e	.init	ASCII	5	6
0x00000094	.plt.got	ASCII	8	9
0x00000098	crackme2.c	ASCII	10	11

Sidebar

Function: LOAD0:sym.valida

Offset info:

STACKPTR	8
STACKOP	inc
FAMILY	cpu
STACK	inc
ESIL	rbp,8,rsp,-=,rsp,=[8]
TYPE	upush
SIZE	1
REFPTR	0
BYTES	55
ID	588
PREFIX	0
MNEMONIC	push

Opcode description:

Function registers info:

A	rsp rbp of sf zf pf cf rdi rcx rax edx rd:
I	rsp rbp rdi rdx rip edi xmm0 r15 r14 r
N	
R	rsp rbp rdi rdx rcx rax rip edi rsi xmm
W	rsp rbp of sf zf pf cf rcx rax edx rdx rs

Console

```
> Analysis finished
> Populating UI
> Finished, happy reversing :)
```

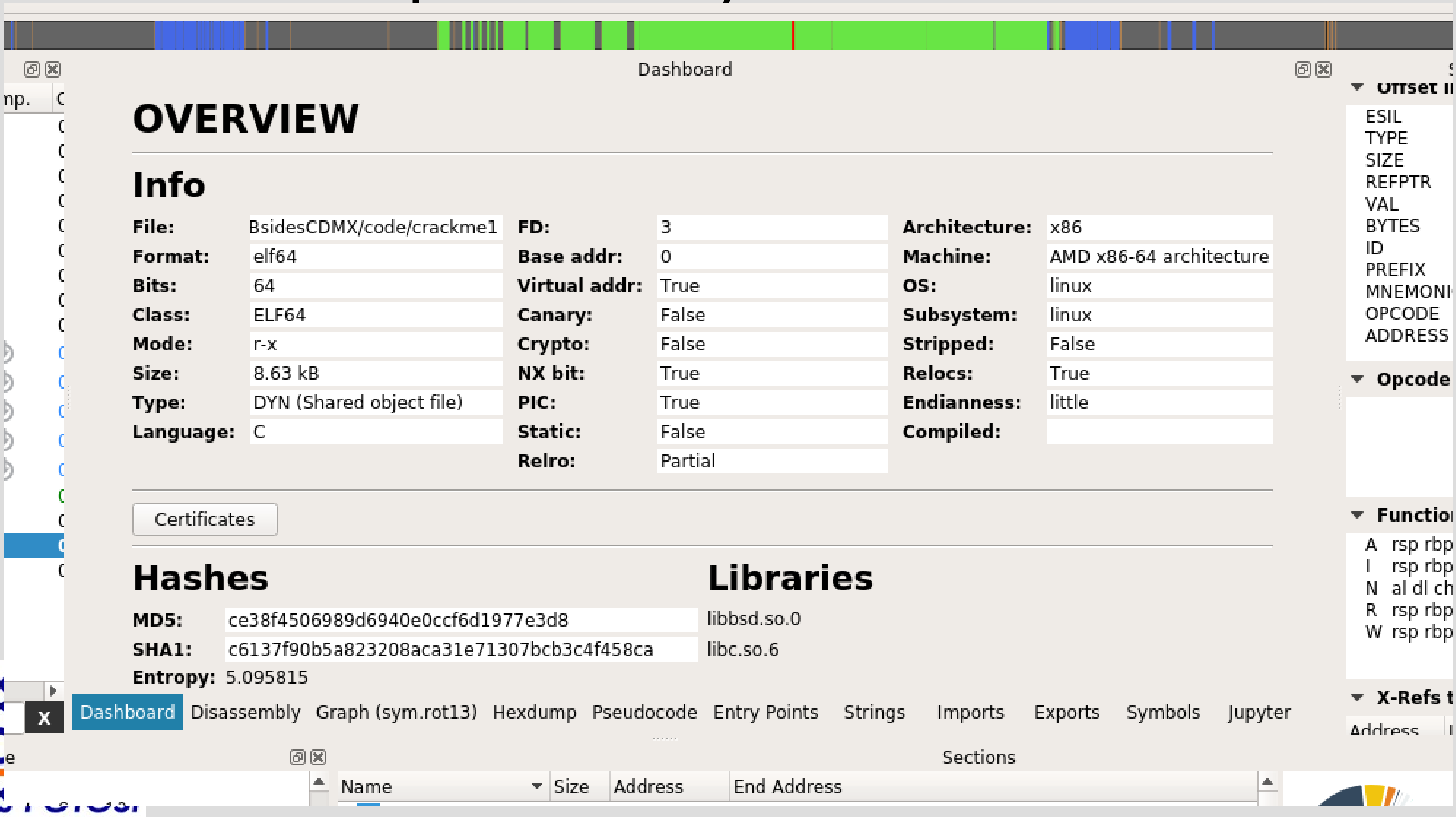
Sections

	Name	Size	Address	End Address
🟦	.bss	0	0x00202068	0x00202070
🟪	.comment	29	0x00000000	0x0000001d
🟩	.data	16	0x00202058	0x00202068
⬜	.dynamic	496	0x00201de8	0x00201fd8
🟨	.dynstr	218	0x00000408	0x000004e2
⬛	.dynsym	336	0x000002b8	0x00000408

no IPv6 | 221.5 GiB | W: (92% at linksys-g) 172.17.2.101 | E: down | UNK 90.48% | 0.36 | 2018-06-11 19:00:55

Crackme 1

- https://github.com/hugo-glez/BSidesCDMX/
- Es un Binario para linux, 64 bits



Dashboard

OVERVIEW

Info

File:	BsidesCDMX/code/crackme1	**FD:**	3	**Architecture:**	x86
Format:	elf64	**Base addr:**	0	**Machine:**	AMD x86-64 architecture
Bits:	64	**Virtual addr:**	True	**OS:**	linux
Class:	ELF64	**Canary:**	False	**Subsystem:**	linux
Mode:	r-x	**Crypto:**	False	**Stripped:**	False
Size:	8.63 kB	**NX bit:**	True	**Relocs:**	True
Type:	DYN (Shared object file)	**PIC:**	True	**Endianness:**	little
Language:	C	**Static:**	False	**Compiled:**	
		Relro:	Partial		

Certificates

Hashes

MD5: ce38f4506989d6940e0ccf6d1977e3d8

SHA1: c6137f90b5a823208aca31e71307bcb3c4f458ca

Entropy: 5.095815

Libraries

libbsd.so.0

libc.so.6

UNIVERS POLITÉC DE SAN LUIS

Demo

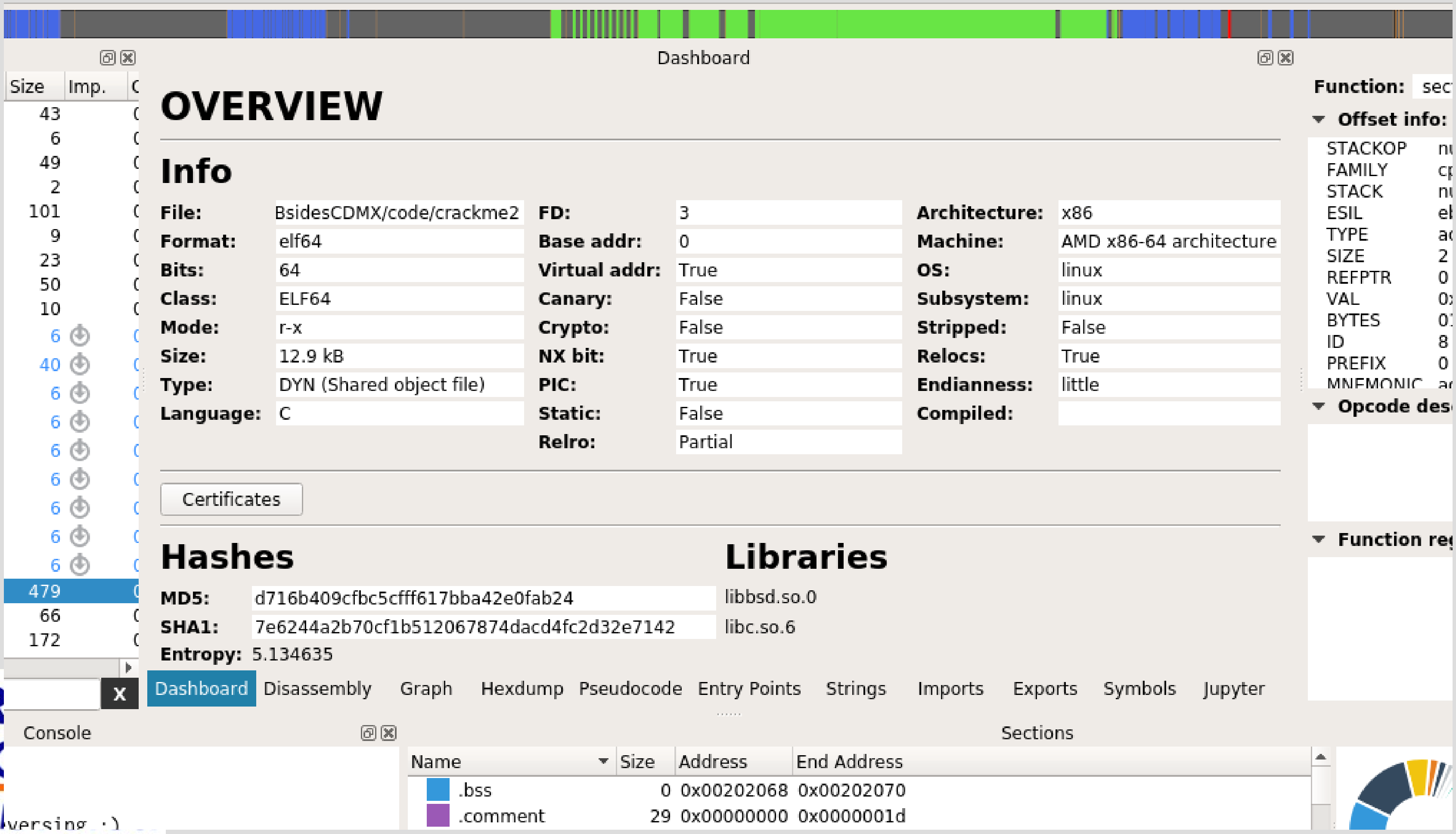
UNIVERSIDAD POLITÉCNICA DE SAN LUIS POTOSÍ

Solución

- Buscar la funcion “valida”
- Ver pseudocódigo para encontrar la cadena
- Aplicar rot13
  - !!rahash2 -E rot -S s:13 -s 'xxxxxxxxxxxx'
- FTW!

Crackme 2

- https://github.com/hugo-glez/BSidesCDMX/
- Este pide tres valores para felicitarte.



OVERVIEW

Info

File:	BsidesCDMX/code/crackme2	**FD:**	3	**Architecture:**	x86
Format:	elf64	**Base addr:**	0	**Machine:**	AMD x86-64 architecture
Bits:	64	**Virtual addr:**	True	**OS:**	linux
Class:	ELF64	**Canary:**	False	**Subsystem:**	linux
Mode:	r-x	**Crypto:**	False	**Stripped:**	False
Size:	12.9 kB	**NX bit:**	True	**Relocs:**	True
Type:	DYN (Shared object file)	**PIC:**	True	**Endianness:**	little
Language:	C	**Static:**	False	**Compiled:**	
		Relro:	Partial		

Certificates

Hashes

MD5: d716b409cfbc5cfff617bba42e0fab24

SHA1: 7e6244a2b70cf1b512067874dacd4fc2d32e7142

Entropy: 5.134635

Libraries

libbsd.so.0

libc.so.6

Name	Size	Address	End Address
.bss	0	0x00202068	0x00202070
.comment	29	0x00000000	0x0000001d

Demo

Solución

- El último password es 3.14

Ejemplo con Mirai botnet

- Objetivo, desencriptar la configuración del ejemplo

Demo

Conclusiones

- Cutter nos permite disminuir la curva de aprendizaje de radare2
- Tiene soporte de Jupyter, entonces podemos automatizar algo con python y r2pipe
- Para análisis manuales es muy útil.
- Aprendan radare2 desde la linea de comandos, o utilicen r2pipe para automatizar.

Preguntas