rubypwn Documentation

Release 0.0.5

atdog

December 24, 2015

Contents

pwntools for ruby developer

CHAPTER 1

Getting Started

1.1 About rubypwn

A simple library for CTF pwning challegnges.

Like Python's pwntools, it's used to help you write exploit quickly.

1.2 Getting Started

1.2.1 Installation

```
gem install rubypwn
```

1.2.2 Basic Usage

class Exec

```
2.2.2 :002 > e = Exec.new "ls"
 => #<Exec:0x007f96742814e8>
2.2.2 :003 > e.gets
Makefile
 => "Makefile\n"
2.2.2 :004 > e = Exec.new "ls", debug: false, color: false
 => #<Exec:0x007f96742814e8>
```

class Netcat

```
2.2.2 :004 > e = Netcat.new "www.google.com", 80
 => #<Netcat:0x007f9671a00cf8>
2.2.2 :005 > e.puts "GET / HTTP/1.1"
GET / HTTP/1.1
 => #<IO:fd 12>
2.2.2 :006 > e.puts
 => #<IO:fd 12>
2.2.2 :008 > e.gets
HTTP/1.1 302 Found
 => "HTTP/1.1 302 Found\r\n"
```

1.3 Format String

1.3.1 Basic Usage

String Ext

```
2.2.3 :001 > require 'rubypwn'
 => true
2.2.3 :002 > str = "prefix_str_"
 => "prefix_str_"
2.2.3 :003 > str.fmtstr
str.fmtstr
2.2.3 :003 > str.fmtstr i32(0x601808), 10, bytes: 2
 => "prefix_str_%6152c%10$hn%59480c%11$hn"
2.2.3 :004 > str.fmtstr "system", 12, bytes: 2
 => "prefix_str_%6152c%10$hn%59480c%11$hn%30995c%12$hn%64256c%13$hn%63730c%14$hn"
2.2.3 :005 > "test".fmtstr i64(0x3121), 1, fmt_size: 100
 => "test%189c%1$hhn%16c%2$hhn%207c%3$hhn%4$hhn%5$hhn%6$hhn%7$hhn%8$hhn"
```

CHAPTER 2

Module Index

All documented module in rubypwn.

2.1 Useful Function

- **def i64()**

```
2.2.2 :004 > a = 0x1234567890abcdef
 => 1311768467294899695
2.2.2 :005 > i64 a
 => "\xEF\xCD\xAB\x90xV4\x12"
```

- **def i32()**
- **def i16()**
- **def s64()**
- **def s32()**

```
2.2.2 :004 > a = "\x30\x00\x00\x00"
 => "0\u0000\u0000\u0000"
2.2.2 :005 > s32 a
 => 48
```

- **def s16()**
- **def c()**

```
2.2.2 :004 > a = 3
 => 3
2.2.2 :005 > c a
 => "\x03"
```

- **def hex()**

```
2.2.2 :002 > a = "test"
 => "test"
2.2.2 :003 > hex a
 => "74657374"
```

- **def nop()**

```
2.2.2 :002 > nop
 => "\x90"
```

2.2 class Asm

Used to compile assembly code

```
2.2.3 :004 > Asm.compile "push eax"
 => "50"
2.2.3 :005 > Asm.compile "push rax", arch: "amd64"
 => "50"
2.2.3 :012 > Asm.compile "mov r15, r14", arch: "arm", format: "c"
 => "\\x0e\\xf0\\xa0\\xe1"
```

2.3 class Elf

Used to get some constant value from the binary:

```
2.2.2 :001 > require 'pp'
2.2.2 :002 > require 'rubypwn'
 => true
2.2.2 :003 > e = Elf.new "traveller"
2.2.2 :004 > pp e
#<Elf:0x007fb498862550
 @arch="x86",
 @bits=32,
 @dynamic=
  {"strtab"=>134513496,
   "symtab"=>134513160,
   "rel_type"=>"REL",
   "jmprel"=>134513844},
 @global={"__gmon_start__"=>{"offset"=>134521192, "value"=>0}},
 @got=
  {"__errno_location"=>134521208,
   "sigemptyset"=>134521212,
   "getpid"=>134521216,
   "__gmon_start__"=>134521220,
   "__isoc99_sscanf"=>134521224,
   "fgets"=>134521228,
   "__libc_start_main"=>134521232,
   "sigaltstack"=>134521236,
   "siglongjmp"=>134521240,
   "sigaction"=>134521244,
   "strlen"=>134521248,
   "printf"=>134521252,
   "setvbuf"=>134521256,
   "puts"=>134521260,
   "kill"=>134521264,
   "__sigsetjmp"=>134521268,
   "exit"=>134521272},
 @sections=
  {""=>{"addr"=>0, "offset"=>0, "size"=>0, "flag"=>"r--"},
   ".interp"=>{"addr"=>134512948, "offset"=>308, "size"=>19, "flag"=>"r--"},
   ".note.ABI-tag"=>
    {"addr"=>134512968, "offset"=>328, "size"=>32, "flag"=>"r--"},
   ".hash"=>{"addr"=>134513000, "offset"=>360, "size"=>160, "flag"=>"r--"},
   ".dynsym"=>{"addr"=>134513160, "offset"=>520, "size"=>336, "flag"=>"r--"},
   ".dynstr"=>{"addr"=>134513496, "offset"=>856, "size"=>232, "flag"=>"r--"},
   ".gnu.version"=>
```

```
 {"addr"=>134513728, "offset"=>1088, "size"=>42, "flag"=>"r--"},
".gnu.version_r"=>
 {"addr"=>134513772, "offset"=>1132, "size"=>48, "flag"=>"r--"},
".rel.dyn"=>{"addr"=>134513820, "offset"=>1180, "size"=>24, "flag"=>"r--"},
".rel.plt"=>{"addr"=>134513844, "offset"=>1204, "size"=>136, "flag"=>"r--"},
".init"=>{"addr"=>134513980, "offset"=>1340, "size"=>48, "flag"=>"r-x"},
".plt"=>{"addr"=>134514028, "offset"=>1388, "size"=>288, "flag"=>"r-x"},
".text"=>{"addr"=>134514320, "offset"=>1680, "size"=>1612, "flag"=>"r-x"},
".fini"=>{"addr"=>134515932, "offset"=>3292, "size"=>28, "flag"=>"r-x"},
".rodata"=>{"addr"=>134515960, "offset"=>3320, "size"=>445, "flag"=>"r--"},
".eh_frame_hdr"=>
 {"addr"=>134516408, "offset"=>3768, "size"=>100, "flag"=>"r--"},
".eh_frame"=>
 {"addr"=>134516508, "offset"=>3868, "size"=>368, "flag"=>"r--"},
".ctors"=>{"addr"=>134520972, "offset"=>4236, "size"=>8, "flag"=>"rw-"},
".dtors"=>{"addr"=>134520980, "offset"=>4244, "size"=>8, "flag"=>"rw-"},
".jcr"=>{"addr"=>134520988, "offset"=>4252, "size"=>4, "flag"=>"rw-"},
".dynamic"=>{"addr"=>134520992, "offset"=>4256, "size"=>200, "flag"=>"rw-"},
".got"=>{"addr"=>134521192, "offset"=>4456, "size"=>4, "flag"=>"rw-"},
".got.plt"=>{"addr"=>134521196, "offset"=>4460, "size"=>80, "flag"=>"rw-"},
".data"=>{"addr"=>134521276, "offset"=>4540, "size"=>8, "flag"=>"rw-"},
".bss"=>{"addr"=>134521312, "offset"=>4548, "size"=>16812, "flag"=>"rw-"},
".comment"=>{"addr"=>0, "offset"=>4548, "size"=>61, "flag"=>"r--"},
".shstrtab"=>{"addr"=>0, "offset"=>4609, "size"=>213, "flag"=>"r--"}}>
```

CHAPTER 3

Executable Index

All documented executable binary in rubypwn

3.1 patch_alarm

Patch alarm() to isnan().

How to use?

```
$ cat test.c && make test
#include <unistd.h>
main() {
    alarm(0);
}
cc     test.c   -o test

$ patch_alarm ./test
Done.

$ patch_alarm ./test.patch
No "alarm" found.

$ ltrace ./test.patch
__libc_start_main(0x40052d, 1, 0x7ffe3ca9a1b8, 0x400540 <unfinished ...>
isnan(0, 0x7ffe3ca9a1b8, 0x7ffe3ca9a1c8, 0)                                    = 0
+++ exited (status 0) +++
```

CHAPTER 4

Indices and tables

- genindex
- modindex
- search