



Practical DEP + ASLR



DEP

Automatic ROP Chain Building

Ropper – Automated RopChain Building

- `./Ropper.py --file /bin/ls --chain "execve cmd=/bin/sh" --badbytes 000a0d`
 - <https://anee.me/hackover-ctf-2016-ping-gnop-writeup-246f68b083aa>
 - <https://advancedpersistentjest.com/2017/07/31/writeups-rev75-simplephp-pwn100-bugs-bunny-ctf/>

Angrop - Automated Write/Read/Syscall

- <https://github.com/salls/angrop>
 - <https://bannsecurity.com/index.php/home/10-ctf-writeups/34-openctf-2016-tyro-rop2>

```
>>> import angr, angrop
>>> p = angr.Project("/bin/lx")
>>> rop = p.analyses.ROP()
>>> rop.find_gadgets()
>>> chain = rop.set_regs(rax=0x1337, rbx=0x56565656)
>>> chain.payload_str()
'\xb32@\x00\x00\x00\x00\x007\x13\x00\x00\x00\x00\x00\x00\xa1\x18@\x00\x00\x00\x00\x00VVV\x00\x00\x00\x00'
>>> chain.print_payload_code()
chain = ""
chain += p64(0x410b23) # pop rax; ret
chain += p64(0x1337)
chain += p64(0x404dc0) # pop rbx; ret
chain += p64(0x56565656)
```



ASLR

ASLR

- Theory:
 - 32 bit – No leak needed
 - ASLR bruteforce (Small space)
 - Buffer overflow/Format String/Arbitrary write
 - 64 bit – Leak needed
 - Information Leak (Separate from write)
 - Buffer overflow/Format String/Arbitrary write

No awesome tools for this yet

- Read the write-ups!
 - ASLR 32bit Bruteforce
 - <http://taishi8117.github.io/2015/11/11/stack-bof-2/>
 - ASLR 64bit Bypass through .got/.plt
 - <https://www.trustwave.com/Resources/SpiderLabs-Blog/Baby-s-first-NX-ASLR-bypass/>
 - <https://sploitfun.wordpress.com/2015/05/08/bypassing-aslr-part-iii/>
 - ASLR 32bit/64bit Bypass through ret2libc
 - <https://sploitfun.wordpress.com/2015/05/08/bypassing-aslr-part-i/>
 - <https://sploitfun.wordpress.com/2015/05/08/bypassing-aslr-part-ii/>