SeBROP: Blind ROP Attacks without Returns

Tianning ZHANG, Miao CAI, Diming ZHANG, Hao HUANG

Frontiers of Computer Science,DOI: 10.1007/s11704-021-0342-8

Problems & Ideas

- Problems of conducting code reuse attack under strict environment.
  - Many defense techniques, such as ASLR and XOM, are proposed to defend against ROP attacks.
  - In victim programs, most syscall instructions lack following ret instructions.
- Ideas: SeBROP attack
  - collect all gadgets by blind execution to bypass fine-grained ASLR and XOM
  - exploits the current vulnerable signal checking mechanism to realize the execution flow control even when ret instructions are absent.

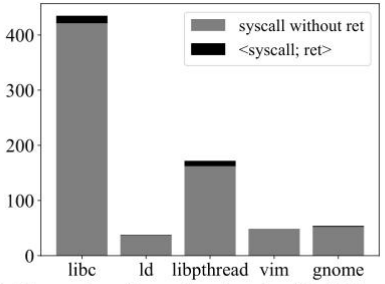

Fig. 1: The number of syscall gadgets in x86_64 libraries and binaries

Table 1: Comparison of Four ROP Attacks. Syscall gadget means the gadget in the form of < `syscall; ret` > or < `int 0x80; ret` >. × means the attack can bypass the protection. √ means that the protection can defend the attack.

Attack	Precondition		Environment			Protections			
	Dump Code	Syscall Gadget	32-bit	64-bit	JIT Compiler	DEP	Fine-grained ASLR	XOM	Readactor
BROP [3]	Yes	No	No	Yes	No	×	×	√	√
SROP [6]	Yes	Yes	No	Yes	No	×	√	√	√
JIT-ROP [11]	Yes	Yes	No	No	Yes	×	×	√	√
SeBROP	**No**	**No**	**Yes**	**Yes**	**No**	×	×	×	×

Main Contributions

SeBROP can defeat almost all state-of-the-art defense techniques.

The SeBROP attack is compatible with both modern 64-bit and 32-bit systems.

SeBROP attack can successfully spawn a remote shell on Nginx, ProFTPD, and Apache with less than 8500/4300/2100 requests, respectively.

Cumulative Number of Requests per Attack Phase

Attack Phase	Nginx	ProFTPD	Apache
Stack Reading	710	0	0
Find basic gadgets	2280	1320	990
Find shared memory	2449	1490	1027
Find all *ret* instruction	5859	3087	1844
Fingerprint all gadgets	8390	4250	2066
Launch a shell	8395	4256	2068

Gadgets found in Apache, ProFTPD and Nginx