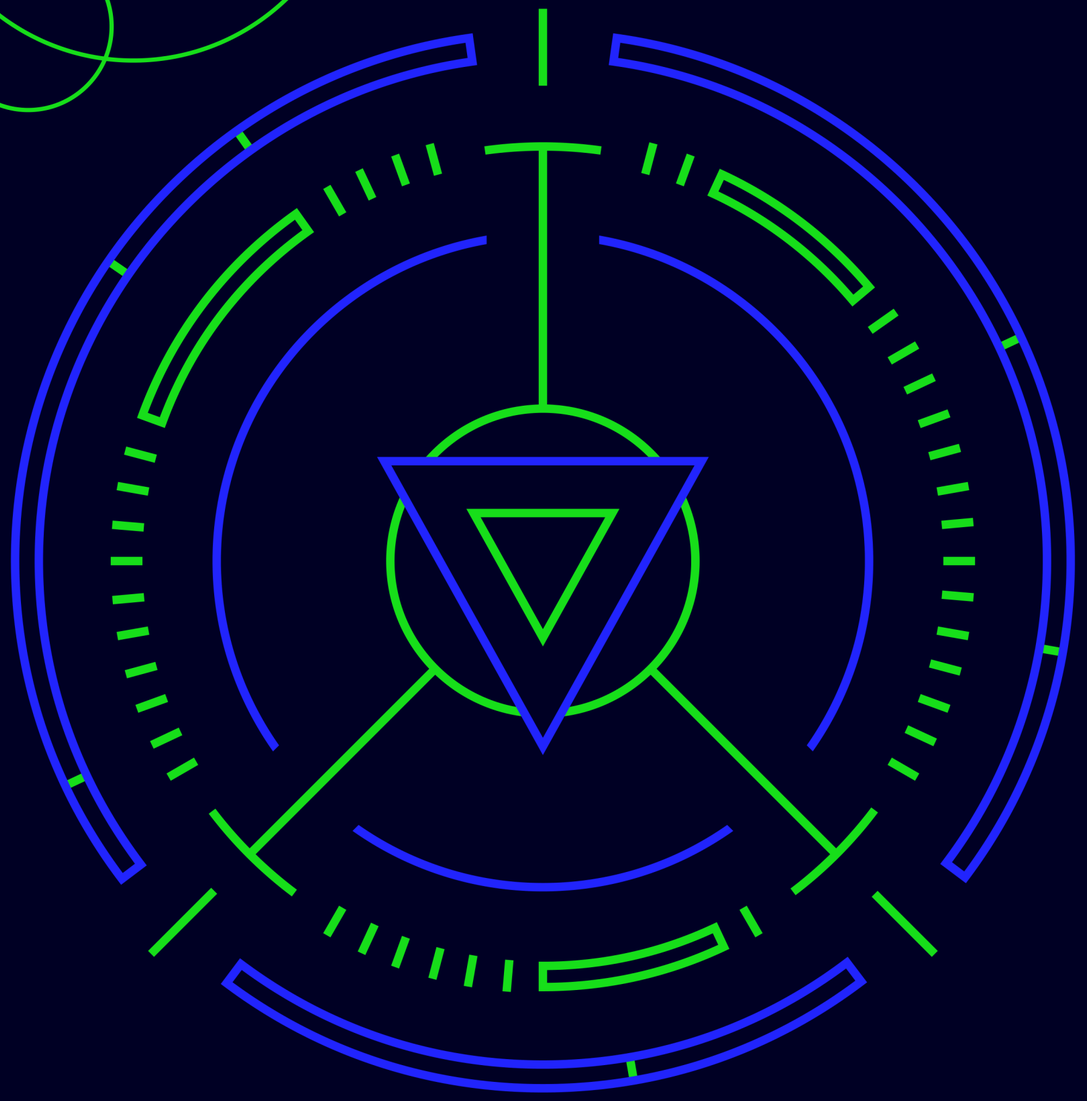

BINARY EXPLOITATION

ADVANCED STACK BASED BINARY EXPLOITATION FOR CTFS

ABOUT ME

PUGAL SELVAN V

PWN & REV

- Currently stuyding EEE III year.
- CTF Player at TAMILCTF.
- Electronics enthusiastic.

PUGALIST_09#5908

Table Of Content

Prerequisites

Knowledges in C language

Knowledges in ASM.

PWN tools & GDB.

Our previous sessions.

ROP

What is Return oriented programming ?

01 Chain gadgets to execute malicious code.

02 A gadget is a suite of instructions which end by the branch instruction ret (Intel) or the equivalent on ARM.

03 Find your gadgets. Store your gadgets addresses on the stack. You must to overwrite the saved eip with the address of your first gadget .

ROP

Why return oriented programming?

01 It can bypass the ASLR .

02 It can bypass the NX bit.

03 A gadget can contain other gadgets.(Not on RISC architectures like ARM, MIPS, SPARC...)

f7c7070000000f9545c3 → test edi, 0x7 ; setnz byte ptr [rbp-0x3d] ;

c7070000000f9545c3 → mov dword ptr [rdi], 0xf000000 ; xchg ebp, eax ; ret

ROP

Tools for ROP

01 Rp++

02 Ropper

03 ROPgadget

OTHER TOOLS :

Ropeme, Ropc and Nrop.

ADDITIONAL INFO:

One Gadget

Ret2CSU

ret2csu is a technique for populating registers when there is a lack of gadgets.When an application is dynamically compiled (compiled with libc linked to it), there is a selection of functions it contains to allow the linking. These functions contain within them a selection of gadgets that we can use to populate registers we lack gadgets.

THEY CONTAIN TWO USEFUL GADGETS, THEY ARE:

01 GADGET 1

02 GADGET 2

Ret2SCU

GADGET 1

0x004011a2	5b	pop rbx
0x004011a3	5d	pop rbp
0x004011a4	415c	pop r12
0x004011a6	415d	pop r13
0x004011a8	415e	pop r14
0x004011aa	415f	pop r15
0x004011ac	c3	ret

Ret2CSU

GADGET 2

```
0x00401188    4c89f2        mov rdx, r14
0x0040118b    4c89ee        mov rsi, r13
0x0040118e    4489e7        mov edi, r12d
0x00401191    41ff14df      call qword [r15 + rbx*8]
```

SROP

Signal Return Oriented Programming

01 It's a varaint of ROP.

02 Uses the SIGRETURN Linux signal to load values from the stack to the registers

03 Store the values on the stack then raise the SIGRETURN syscall .Your registers will be initialized with the stack values .

SROP

ABOUT SIGRETURN SYSCALL

Return from signal handler and cleanup stack frame.

RETURN VALUE

sigreturn() never returns.

TRIGERRING/CALLING SIGRETURN

Syscall number of SIGRETURN is 15(0xf) , So we need to put 15 into accumulator and trigger the syscall instruction.

Stack Pivoting

Stack Pivoting is a technique we use when we lack space on the stack. And also we can move the stack pointer to our desired memory region/segment.

Few ways to do this are:

```
pop rsp
```

```
xchg <reg>, rsp
```

```
leave; ret
```

DOUBTS &TIPS

THANK YOU