INTERNALS OF EXPLOIT WRITING X86-64

JUAN SACCO

EXPLOIT WRITER AND REVERSE ENGINEER, WORKED AT CORE SECURITY, NOD32, HOMELAND SECURITY (ARG) AND OTHERS SECURITY RELATED ORGANIZATIONS.

PRESENT: ING Bank
PAST: Core Security, Nod 32

TWITTER: @JUANSACCO
EMAIL: JSACCO@SDF.ORG

WHY TALKING ABOUT X86-64

MOST EXPLOITS STILL USES X86

BECAUSE EXPLOIT WRITING HAS BECOME MORE AND MORE DIFFICULT SINCE 2004, PROTECTIONS, NO-EXECUTION, RANDOMIZATION AND OTHER KINDS OF DEFENSES ARE APPLIED NOW.

X86-64 BECAME A REALITY 2-3 YEARS AGO AND NOWADAYS EVEN MOBILE DEVICE ARE USING IT.

BOUNTY HUNT? QUICK RELEASE? AVOID SOME STEPS? YEAP. EVEN MS WITH HIS 8.1 AND IE IS DOING IT.

SAY HELLO0 SYSCALLS

EVERYTHING START WITH HELLO WORLD

A QUICK COMPARISON OF 32 AND 64 BIT SYSTEM CALLS.

WITH NO EXCEPTION AN INTEGER VALUE REPRESENTING THE SYSTEM_WRITE CALL IS PLACED IN THE FIRST REGISTER, FOLLOWED BY ITS ARGUMENTS, AND WHEN EVERYTHING IS THEIR PROPER REGISTER S, THE SYSTEM IS CALLED AND THE MESSAGE DISPLAYED.

ASSEMBLER DIFFERENCES

OPCODES, QUAD WORDS AND RELATIVE ADDRESSING

NEW REGISTERS FROM R8 TO R15 ENCODED USING REX PREFIX.

TO WRITE 64BIT INSTRUCTIONS, USE 'Q' AS A SUFFIX (Q FOR 'QUAD-WORD'):

```
  MOVL $1,  %EAX     # 32-BIT INSTRUCTION
  MOVQ $1,  %RAX     # 64-BIT INSTRUCTION
```

WAKE UP NOW COMES 64 BITS!

STACK – BASED BUFFER OVERFLOW

LET'S COMPARE HOW A SIMPLE VULNERABLE PROGRAM WRITTEN IN C IS TRANSLATED TO 32 AND 64.

AFTER THAT WE WILL RUN IT INSIDE A DEBUGGER AND CONTROL EIP BY DOING A TYPICAL OVERFLOW.

DEMO!

FINAL THOUGHTS

EXPLOIT WRITING IN X86-64

Windows and Linux exploits are generally harder to code than in 32 bits platforms, including the facts that some heap sprays are off the table. On top of that Linux ASLR is better than Windows ASLR.

Unfortunately it is quite possible there won't be such a thing as a simple exploit on x64.

THANKS!

CONTACT DETAILS:

EMAIL: JSACCO@SDF.ORG
SKYPE: JUANSACCO