Reversing a Simple Shellcode with Radare2

Edd Barrett

@vext01

Twitter: @radareorg

www: http://radare.org

Introducing Radare2

What is Radare2?

Radare2 is an open-source framework to aid reversing and modification of binary files.

Some features

- Multi-architecture and multi-platform
- Hex editor
- Debugger
- Disassembler
- …

Developers

@trufae, @nibble_ds, @earada and handful of testers and contributors.

Shellcodes

Definition

”Shellcode” is a term colloquially used to refer to the payload of an exploit. Typically this would be code injected to start a shell.

- Not to be confused with “Shell Script”.
- See `http://www.projectshellcode.com/` for examples.

An Example – What does this code do?

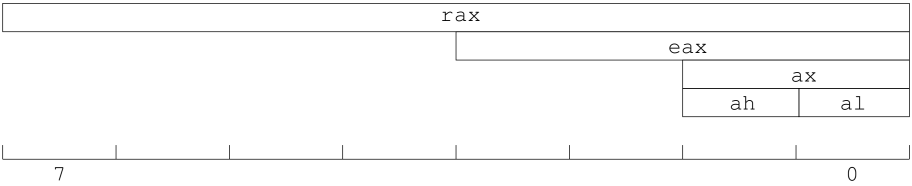
```
char shellcode[] =
        "\xeb\x11\x5e\x31\xc9\xb1\x27\x80"
        "\x6c\x0e\xff\x35\x80\xe9\x01\x75"
        "\xf6\xeb\x05\xe8\xea\xff\xff\xff"
        "\x20\x4a\x66\xf5\xe5\x44\x90\x66"
        "\xfe\x9b\xee\x34\x36\x02\xb5\x66"
        "\xf5\xe5\x36\x66\x10\x02\xb5\x1d"
        "\x1b\x34\x34\x34\x64\x9a\xa9\x98"
        "\x64\xa5\x96\xa8\xa8\xac\x99";

int main(void)
{
        (*(void(*)()) shellcode)();
        return 0;
}
```

1

[1] Thanks to “Gunslinger” for this example

Overlapping Registers in x86/x64



Register Configuration due to Legacy

- In the 16-bit days we had `ax`
  - High and low byte addressable via `ah`, `al`
- In the 32-bit days we also had `eax`
- The newest x64 register has `rax`

Similarly for `bx, cx, dx`.

CALL

From the Intel Manual

Saves procedure linking information on the stack and branches to the called procedure specified using the target operand.

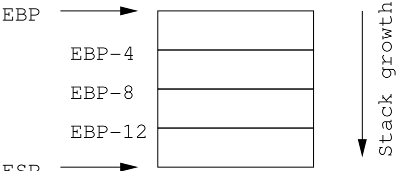
Intel® 64 and IA-32 Architectures Software Developer's Manual
Intel® 64 and IA-32 Architectures Optimization Reference Manual
The SPARC Architecture Manual
SPARC
Version 9

CALL Example

```
0x1c000286    16    e8e1ffffff        call dword 0x1c00026c
0x1c00028b    16    81                ...
```

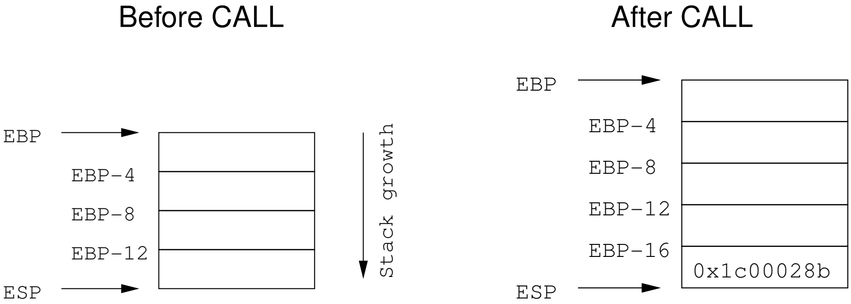
Before CALL

After CALL



CALL Example

```
0x1c000286    16    e8e1ffffff        call  dword  0x1c00026c
0x1c00028b    16    81                ...
```


Before CALL

After CALL

System Calls

Definition

The userland can request services from the kernel by calling special functions known as “system calls”.

How do they work?

- System calls are not called with the `CALL` instr
- Instead an 0x80 interrupt is fired
    - The system call number to execute is in `eax`
    - Arguments should be in { `ebx, ecx, edx, esi, edi, ebp`}

This Exploit Worked Once...

Actually…

- The exploit I have just showed you does not work on modern UNIX/Linux ;)
- NX bit or $W\hat{}X$ prevents such attacks
- Pages in `.data` are writable, therefore not also executable.

Concluding Comments

Thanks for Listening

- Original blog post: `http://canthack.org/2011/07/adventures-with-radare-1-a-simple-shellcode-analysis/`
- Follow radare2 on twitter: @radareorg
- Find radare2 on the web: `http://radare.org`
- Source code for these slides: `https://github.com/vext01/r2-adventures1-talk`