Code Vulnerabilities & Attacks: Classic Buffer Overflow

Marcus Botacin

UFPR

2021

Topics

Topics

ABIs and Calling Conventions (CCs)

- How parameters are passed.
- The order they are passed.
- Where they are passed.

Calling Conventions (CCs)

- **cstd**: Arguments on the `stack`, on the reverse order.
- **x64**: Arguments on `registers`.
- **syscalls**: Syscall number on `eax`.

Concept Definitions

Application Binary Interface (ABI)

"Uma definição da interface entre módulos de *software*, tais como entre códigos e o sistema operacional. Uma ABI especifica uma *calling convention*."

Calling Convention (CC)

"Uma definição de como trechos de código são chamados, incluindo a ordem, localização, e a restauração dos parâmetros."

Stack Frame

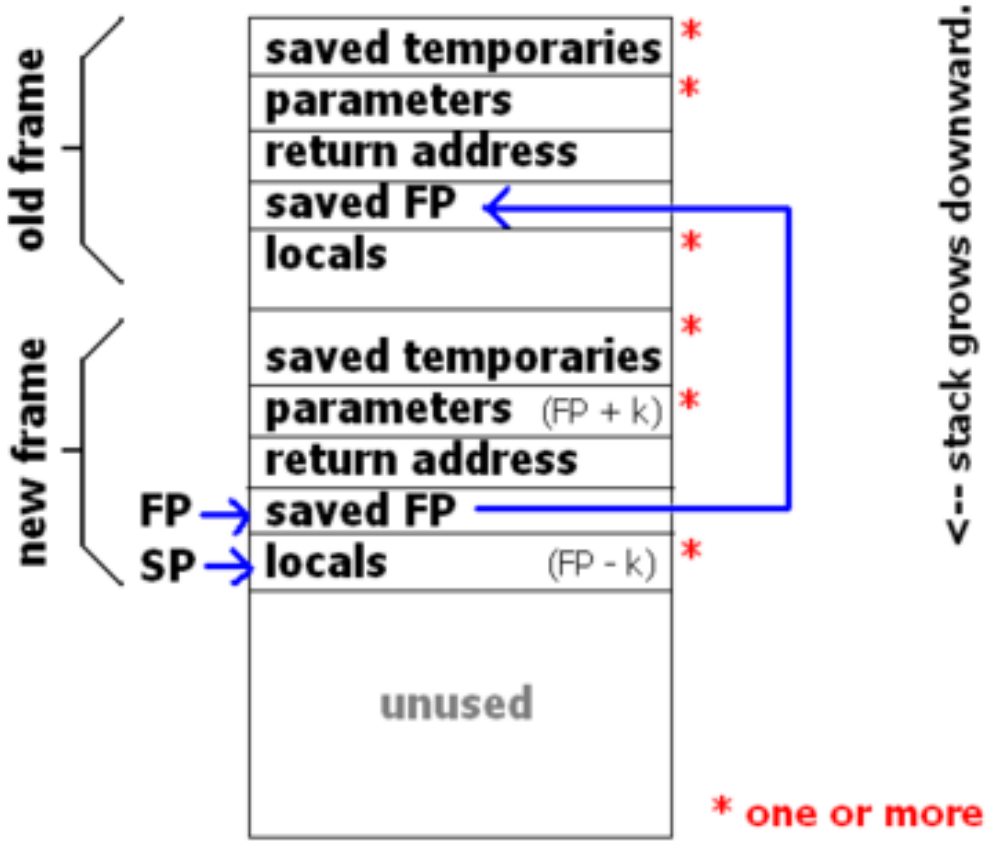

Figure: Stack Frame

[0] `https://tinyurl.com/hhq934r`

Stack Frame

Definition

"Uma definição de contexto de função de forma temporária."

Topics

A basic overflow example

```
1  int main(int argc, char *argv[])
2  {
3     int a = 10;
4     char str[4];
5     scanf("%s",str);
6     printf("%d\n",a);
7     return 0;
8  }
```

Code 1: Variable overflow (Hypothetic code).

In Practice.

```
marcus@tux:/tmp$ ./variable_overflow.bin
a
10
marcus@tux:/tmp$ ./variable_overflow.bin
aa
10
marcus@tux:/tmp$ ./variable_overflow.bin
aaa
10
marcus@tux:/tmp$ ./variable_overflow.bin
aaaa
0
marcus@tux:/tmp$ ./variable_overflow.bin
aaaaa
97
marcus@tux:/tmp$ ./variable_overflow.bin
aaaab
98
```

Figure: Variavle overflow in practice.

Telling the truth.

Implementation a bit harder.

- Compiler optimizations.
- Stack smashing mitigation.

Try to implement.

- How to put variables together?
- Which compiler flags to use?

Topics

Vulnerable code

```
1 int main(int argc, char *argv[])
2 {
3   char str[32];
4   strcpy(str,argv[1]);
5   return 0;
6 }
```

Code 2: Vulnerable code (Hypothetic).

For fun and for profit

.oO Phrack 49 Oo.

Volume Seven, Issue Forty-Nine

File 14 of 16

BugTraq, r00t, and Underground.Org
bring you

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Smashing The Stack For Fun And Profit
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

by Aleph One
aleph1@underground.org

Figure: Aleph One's article.

[0] http://phrack.org/issues/49/14.html

A typical payload

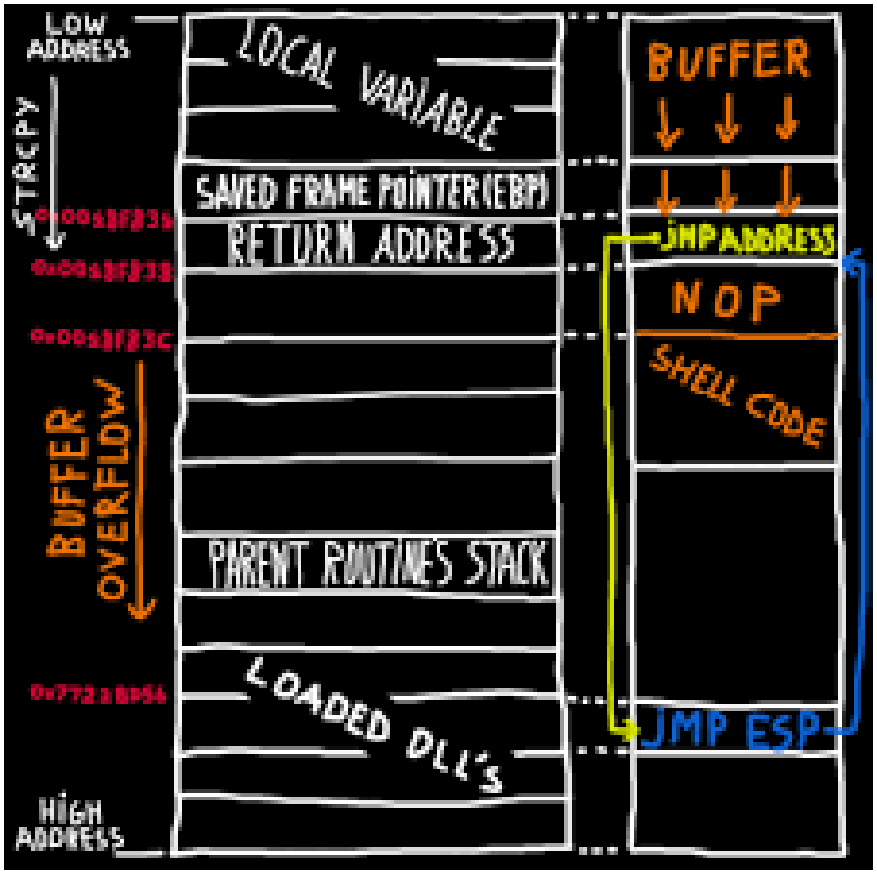

Figure: Buffer overflow payload structure

[0] https://tinyurl.com/y9dornpu

Shellcodes

Twitter Profile
GitHub Profile
Google+ Profile
Linkedin Profile
RSS feeds
Email

Shellcodes database for study cases

- Windows - sp3 (Tr) calc.exe Shellcode 53 bytes *by ZoRLu*
- Windows - sp3 (Tr) cmd.exe Shellcode - 42 bytes *by ZoRLu*
- Windows - sp3 (Tr) cmd.exe Shellcode 52 bytes *by ZoRLu*
- Windows - Xp Pro SP3 Fr (calc.exe) - 31 Bytes *by agix*
- Windows - XP PRO SP3 - Full ROP calc shellcode *by b33f*
- Windows - xp pro sp3 (calc) - 57 bytes *by cr4wl3r*
- Windows - win32/xp pro sp3 MessageBox shellcode - 11 bytes *by d3c0der*
- Windows - download & exec shellcode - 226 bytes+ *by darkeagle*
- Windows - Shellcode Checksum Routine *by dijital1*
- Windows - IsDebuggerPresent ShellCode (NT/XP) - 39 bytes *by ex-pb*
- Windows - PEB method (9x/NT/2k/XP) - 29 bytes *by loco*
- Windows - connectback, receive, save and execute shellcode *by loco*
- Windows - Bind Shell (NT/XP/2000/2003) - 356 bytes *by metasploit*
- Windows - Create Admin User Account (NT/XP/2000) - 304 bytes *by metasploit*
- Windows - Vampiric Import Reverse Connect - 179 bytes *by metasploit*
- Windows - PEB method (9x/NT/2k/XP) *by oc192*
- Windows - eggsearch shellcode - 33 bytes *by oxff*
- Windows - XP-sp1 portshell on port 58821 - 116 bytes *by silicon*
- Windows - XP SP3 addFirewallRule *by sinn3r*

Figure: Shellcode Repository.

[0] `http://shell-storm.org/shellcode/`

Concept Definitions

Buffer

"Região **contígua** de memória, de tamanho **limitado**, utilizada para armazenamento **temporário**."

Buffer Overflow

"Exceder a capacidade de armazenamento de um *buffer*."

Buffer Overflow Attack

"Utilizar-se de um *buffer overflow* para alterar **intencionalmente** o estado de um programa."

Concept Definitions

Payload

"Conteúdo de preenchimento do *buffer* em um ataque do tipo *overflow*"

Shellcode

"Conjunto de instruções que compõem um *payload*"

NOP

"Uma operação que não altera estados"

NOP Sled

"Sequência de NOPs que compõem um *payload*"

Buffer Overflow Classification

Types

- Stack-based.
- Heap-based.

Topics

No eXecutable (NX) pages

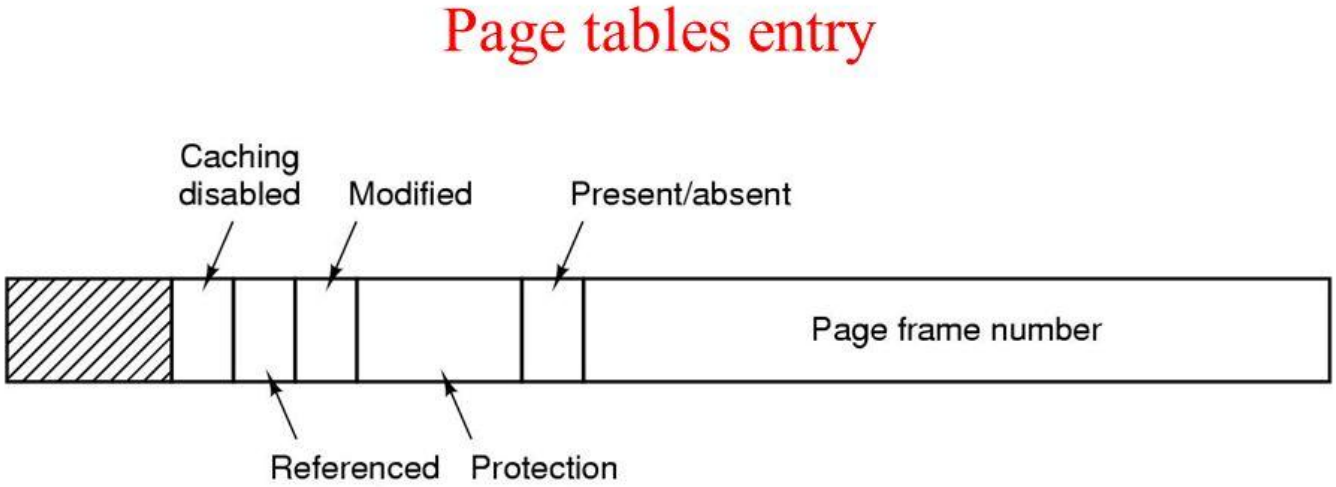

Figure: Page protection bits

[0] `https://tinyurl.com/y8wodhn2`

Back to O.S (again!!)

Libs and permissions

```
marcus@malware-lab:~/Documentos/aula-ROP$ cat /proc/self/maps
00400000-0040c000 r-xp 00000000 08:05 11022740                           /bin/cat
0060b000-0060c000 r--p 0000b000 08:05 11022740                           /bin/cat
0060c000-0060d000 rw-p 0000c000 08:05 11022740                           /bin/cat
00d56000-00d77000 rw-p 00000000 00:00 0                                  [heap]
7f3411e47000-7f3412121000 r--p 00000000 08:05 9052298                    /usr/lib/locale/locale-archive
7f3412121000-7f34122e1000 r-xp 00000000 08:05 20189578                   /lib/x86_64-linux-gnu/libc-2.23.so
7f34122e1000-7f34124e1000 ---p 001c0000 08:05 20189578                   /lib/x86_64-linux-gnu/libc-2.23.so
7f34124e1000-7f34124e5000 r--p 001c0000 08:05 20189578                   /lib/x86_64-linux-gnu/libc-2.23.so
7f34124e5000-7f34124e7000 rw-p 001c4000 08:05 20189578                   /lib/x86_64-linux-gnu/libc-2.23.so
7f34124e7000-7f34124eb000 rw-p 00000000 00:00 0
7f34124eb000-7f3412511000 r-xp 00000000 08:05 20189562                   /lib/x86_64-linux-gnu/ld-2.23.so
7f34126ca000-7f34126ef000 rw-p 00000000 00:00 0
7f3412710000-7f3412711000 r--p 00025000 08:05 20189562                   /lib/x86_64-linux-gnu/ld-2.23.so
7f3412711000-7f3412712000 rw-p 00026000 08:05 20189562                   /lib/x86_64-linux-gnu/ld-2.23.so
7f3412712000-7f3412713000 rw-p 00000000 00:00 0
7ffea5db9000-7ffea5dda000 rw-p 00000000 00:00 0                          [stack]
```

Figure: Memory mapping and protection

Topics

Still overflowing

```
1  int main(int argc, char *argv[])
2  {
3     int a = 10;
4     char str[4];
5     scanf("%s",str):
6     printf("%d\n",a);
7     return 0;
8  }
```

Code 3: Variable overflow (Hypothetic).

Concept Definitions

Program Hijacking

- **Variable Hijacking**: “Alterar uma variável que controla um estado”.
- **Instruction Hijacking**: “Inserção de instruções para alterar o estado”.
- **Control Flow Hijacking**: ”Uso das técnicas anteriores para alterar o estado”.

Challenge

How to avoid control flow hijacking?

Topics

Challenge

How to implement a simple buffer overflow payload?

Topics

Conclusion

Questions?

mfbotacin@inf.ufpr.br