INSTITUTO SUPERIOR TÉCNICO

Debugging

de aplicações em C e C++

Nuno Lopes, NEIIST – 7º Ciclo de Apresentações. 26/Abril/2007

INSTITUTO SUPERIOR TÉCNICO

Agenda

- Erros/Warnings do compilador
- Bugs típicos
- Ferramentas úteis (GDB, Valgrind, et all)
- Casos Reais

- Questões

INSTITUTO SUPERIOR TÉCNICO

Agenda

- ⇨ **Erros/Warnings do compilador**
- Bugs típicos
- Ferramentas úteis (GDB, Valgrind, et all)
- Casos Reais

- Questões

INSTITUTO SUPERIOR TÉCNICO

Erros/Warnings do compilador

- Detecção preventiva de bugs
- Mas... Compillar != Funcionar != Correcto
- Compiladores mais recentes detectam mais problemas

- ‘gcc –O2 –Wall -g’ vs ‘gcc –O0 –Wall -g’

INSTITUTO SUPERIOR TÉCNICO

Exemplos

- format '%ld' expects type 'long int', but argument 3 has type 'size_t‘
- pointer targets in passing argument 1 of 'gdGetInt' differ in signedness
- 'php_safe_m' defined but not used

INSTITUTO SUPERIOR TÉCNICO

Exemplos

- implicit declaration of function 'php_gd_lex'
- return type defaults to 'int‘
- enumeration value 'PDO_FREE' not handled in switch
- value computed is not used

INSTITUTO SUPERIOR TÉCNICO

Agenda

- Erros/Warnings do compilador
- ⇨ **Bugs típicos**
- Ferramentas úteis (GDB, Valgrind, et all)
- Casos Reais

- Questões

Double-free

- free(pointer) mais do que uma vez
- Origina crash na maioria das libcs

```
char *ptr = malloc(sizeof(xpto));
...
if (error) {
        free(ptr);
        ...
}
...
free(ptr);
```

*** glibc detected *** myApp: double free or corruption (fasttop): 0xdeadbeef ***

INSTITUTO SUPERIOR TÉCNICO

output do valgrind

```
Invalid free() / delete / delete[]
   at 0x402142C: free (in /usr/.../vgpreload_memcheck.so)
   by 0x8048409: main (test.c:6)
 Address 0x4162028 is 0 bytes inside a block of size 1 free'd
   at 0x402142C: free (in /usr/.../vgpreload_memcheck.so)
   by 0x80483FE: main (test.c:5)
```

Memory Leak

- Memória que é alocada não é devolvida

```
char *ptr = malloc(sizeof(int));
...
if (error) {
      rollback_stuff();
      return NULL; // não fez free

}
...
free(ptr);
```

INSTITUTO SUPERIOR TÉCNICO

output do valgrind

```
4 bytes in 1 blocks are definitely lost in loss record 1 of 1
   at 0x4021888: malloc (in /usr/.../vgpreload_memcheck.so)
   by 0x80483E5: leak (memleak.c:5)
   by 0x804842B: main (memleak.c:18)
```

Buffer Overflow

- Escrever fora da área alocada

```
char *strdup(const char *str)
{
        char *copy = malloc(strlen(str));
        char *tmp  = copy;

        do {
                *tmp++ = *str;
        } while (*str++)

        return copy;
}
```

INSTITUTO SUPERIOR TÉCNICO

output do valgrind

```
Invalid write of size 1
   at 0x804842D: strdup (in buffer-overflow.c:10)
   by 0x8048469: main (in buffer-overflow.c:17)
 Address 0x416202C is 0 bytes after a block of size 4 alloc'd
   at 0x4021888: malloc (in /usr/.../vgpreload_memcheck.so)
   by 0x804841A: strdup (in buffer-overflow.c:6)
   by 0x8048469: main (in buffer-overflow.c:17)
```

Memória não inicializada

- Acesso a memória não inicializada. Pode induzir comportamentos “aleatórios”

```
void readBuffer(void)
{
      char *buffer = malloc(1024);

      if (/* cond */ 0) {
            strcpy(buffer, "ola");
      }

      printf("%s\n", buffer);
      free(buffer);
}
```

INSTITUTO SUPERIOR TÉCNICO

output do valgrind

```
Conditional jump or move depends on uninitialised value(s)
   at 0x40226BB: strlen (in /usr/.../vgpreload_memcheck.so)
   by 0x408E834: puts (in /lib/libc-2.5.so)
   by 0x8048423: readBuffer (mem-not-init.c:13)
   by 0x8048446: main (mem-not-init.c:18)
```

INSTITUTO SUPERIOR TÉCNICO

Format strings

- printf("%s", str) vs printf(str)
- Erro de segurança (*exploitável*)

Dangling Pointers

- Ponteiros para regiões inválidas (stack ou heap)

```
char* I_hate_pointers(void)
{
        char buf[16];
        *buf = '\0';
        return buf;
}

main() {
        printf("%s", I_hate_pointers());
}
```

```
dangling-pointers.c: In function 'I_hate_pointers':
dangling-pointers.c:7: warning: function returns address of local variable
```

long vs int

- int != long != size_t nalgumas plataformas (e.g. 64 bits)

```
void readBuffer(char *buf, long *read) {
        *read = 0xdeadead;
}
main() {
        char buf[10];
        int x;
        readBuffer(buf, &x);
}
```

```
int-vs-long.c: In function 'main':
int-vs-long.c:11: warning: passing argument 2 of 'readBuffer' from incompatible
pointer type
```

Integer Overflow

- Cuidado com a aritmética modular..

```
char* alloc_img_buffer(int width, int height)
{
      // 1 byte per pixel + 13 byte
      return malloc(width * height + 13);
}

int main()
{
      // overflow...
      char *buffer = alloc_img_buffer(2147483647, 10);

      strcpy(buffer, "cabum!!!!!");
}
```

Integer Underflow

- Não assumir nada sobre o input

```
void copy_skip_5(char *src, char *dest, size_t len)
{
      len -= 5;
      memcpy(dest, src, len);
}

int main()
{
      char src[]="oi";
      char dest[10];
      size_t len = strlen(src); // 2

      copy_skip_5(src, dest, len);
}
```

INSTITUTO SUPERIOR TÉCNICO

Signed vs Unsigned

```
int table2[] = {50, 51, 52, 53};
int table[] = {1, 2, 3, 4, 5};

int lookup_char(char c)
{
      return table[(int)c];
}

int main()
{
      int i = -5;
      if ((unsigned)i > 5) printf("why -5 > 5?\n");


      printf("%d\n", lookup_char(0xFF)); // 53, why??
}
```

INSTITUTO SUPERIOR TÉCNICO

(des)Alinhamento de memória

- Certos tipos de dados têm que estar alinhados na memória
- Pouco relevante em x86 (excepto SSE)
- Em SPARC origina crashes (SIGBUS)

```
void tricky_archs(void)
{
        char buf[sizeof(double)];
        double *n = (double*)buf;
        *n = 66.6;
}
```

INSTITUTO SUPERIOR TÉCNICO

Agenda

- Erros/Warnings do compilador
- Bugs típicos
- ⇨ **Ferramentas úteis (GDB, Valgrind, et all)**
- Casos Reais

- Questões

printf()

- O mais usado pelos novatos

```
// a ordem de avaliação dos parâmetros é arbitrária
printf("%d %d\n", systemcall, errno)


printf("%d\n", systemcall);
// aqui lemos o errno do printf em vez do errno da
// syscall. usar perror() ajuda
printf("%d\n", errno)
```

INSTITUTO SUPERIOR TÉCNICO

strace

- Lista system calls e respectivos argumentos
- -e trace=(file|process|network|signal|...) restringe system calls a interceptar

INSTITUTO SUPERIOR TÉCNICO

ltrace

- Lista chamadas a funções de bibliotecas dinâmicas (*.so), e.g. Libc
- Crasha com alguma frequência (especialmente com programas multi-threaded)

INSTITUTO SUPERIOR TÉCNICO

GDB

```
# gdb ./test

(gdb) break strdup
Breakpoint 1 at 0x80483fb: file prog.c, line 6.

(gdb) run
Starting program: /tmp/test
Breakpoint 1, strdup (str=0x8048558 "xpto") at prog.c:6
6                 char *copy = malloc(strlen(str));

(gdb) bt
#0  strdup (str=0x8048558 "xpto") at prog.c:6
#1  0x0804846a in main () at prog.c:17

(gdb) frame 1
#1  0x0804846a in main () at prog.c:17
17                free(strdup("xpto"));
```

INSTITUTO SUPERIOR TÉCNICO

GDB

```
(gdb) watch n
Hardware watchpoint 2: n

(gdb) continue
Continuing.
Hardware watchpoint 2: n
Old value = (double *) 0x80495dc
New value = (double *) 0xbfcedf1c
tricky_archs () at unaligned-memory.c:88
             *n = 66.6;

(gdb) p n
$1 = (double *) 0xbfcedf1c

(gdb) step
9                   return *n;

(gdb) p *n
$2 = 66.599999999999994
```

INSTITUTO SUPERIOR TÉCNICO

Comandos GDB

- break x – adiciona breakpoint
- bt – imprime backtrace
- step – salta para a próxima linha
- continue – salta até próximo breakpoint
- run <args> - corre programa
- print x – imprime variável
- watch – adiciona watchpoint
- help – ajuda ☺

INSTITUTO SUPERIOR TÉCNICO

valgrind

Framework com múltiplas ferramentas:

- Callgrind – profiler
- Massif – heap profiler
- Helgrind – detector de race-conditions
- Memcheck – ferramenta mais usada. Detecta problemas de memória

INSTITUTO SUPERIOR TÉCNICO

Opções Valgrind/Memcheck

- --trace-children=yes
- --db-attach=yes – corre gdb quando encontra erro
- --leak-check=full – lista exaustiva de leaks
- --show-reachable=yes – lista de memória não liberta mas referenciada
- --track-fds=yes – listagem de FDs não fechados

INSTITUTO SUPERIOR TÉCNICO

Outras Ferramentas

- IBM Rational Purify – semelhante ao valgrind/memcheck, mas multi-plataforma
- Coverity – análise estática de código (incluindo problemas de segurança)
- Intel Thread Checker – debugger para problemas de multi-threading
- dtrace – debugger e profiler para Solaris

INSTITUTO SUPERIOR TÉCNICO

Agenda

- Erros/Warnings do compilador
- Bugs típicos
- Ferramentas úteis (GDB, Valgrind, et all)
- ⇨ **Casos Reais**

- Questões

INSTITUTO SUPERIOR TÉCNICO

Signed char

- http://news.php.net/php.cvs/42544
- /php-src/ext/pdo/pdo_sql_parser.re
- Loop infinito com chars com valor > 127

INSTITUTO SUPERIOR TÉCNICO

Agenda

- Erros/Warnings do compilador
- Bugs típicos
- Ferramentas úteis (GDB, Valgrind, et all)
- Casos Reais

⇨ **Questões**