


Buffer Overflows

Parte II: Vulnerabilidades

Segurança de Software

2019
Nuno Santos



Aula passada

► **Mecanismos** para evitar vulnerabilidades e proteger contra software malicioso ou incorrecto

```
graph LR; A[Código Fonte] --> B[Binário]; B --> C[Processo];
```

Implementados nestes dois componentes:

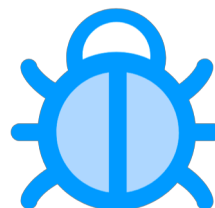
	Compilador da Linguagem de Programação	Sistema Operativo
	Type, memory, e control flow safety Sandboxes	Protecção de memória Controlo de acesso

► 2 SS - Nuno Santos 2019



Apesar de tudo, existem vulnerabilidades

- ▶ Infelizmente esses mecanismos ou são insuficientes, ou por vezes não estão implementados...
- ▶ Resultado: potenciais vulnerabilidades que podem ser exploradas por atacantes
- ▶ Entramos na Parte II: Vulnerabilidades



Onde estamos

- ▶ **Parte I: Enquadramento e protecção (2 aulas)**
 - ▶ Conceitos de segurança de software, mecanismos básicos de segurança
- ▶ **Parte II: Vulnerabilidades (3 aulas)**
 - ▶ Buffer overflows, corridas e validação de entradas, vulnerabilidades na web e em bases de dados
- ▶ **Parte III: Técnicas de protecção (4 aulas)**
 - ▶ Auditoria e teste de software, análise estática de código, protecção dinâmica, validação e codificação
- ▶ **Parte IV: Tópicos avançados (1 aula)**
 - ▶ Trusted computing



Nesta aula

- ▶ Vamos estudar uma classe de bugs muito comum e muito séria chamada **buffer overflow**



▶ 5

SS - Nuno Santos

2019



Exemplo de ataque devido a um buffer overflow

A newly discovered vulnerability in Microsoft's Outlook and Outlook Express programs leave **thousands of computers open to attack** from malicious email (...)

The bug is a classic “**buffer overflow**” error in the section of Outlook that parses the Date field of each incoming email. By padding the date with a long string of characters, an attacker can escape from the area of memory reserved for storing it, and into a section that executes instructions. From there, the attacker's email could secretly infect a victim computer with a “back door” program (...)

- ▶ [Kevin Poulsen, MS Battles Outlook Bug, Security Focus, 19 de Julho de 2000](#)

▶ 6

SS - Nuno Santos

2019



Plano para esta aula

- ▶ Natureza de um buffer overflow
- ▶ Overflow da stack e da heap
- ▶ Integer overflows



Natureza de um buffer overflow



Causas de buffer overflows



- ▶ **C e C++ (ao contrário do Java)**
 - ▶ Não são memory safe, ou seja, não verificam limites da memória aquando do acesso feito pelo programa
 - ▶ O programador assume que o utilizador nunca vai introduzir uma cadeia de caracteres maior do que X
- ▶ **Outros factores**
 - ▶ Número elevado de funções perigosas, ex., strcpy, sprintf
 - ▶ Ensino divulga práticas de programação inseguras

▶ 9

SS - Nuno Santos

2019



O que faz um buffer overflow?

- ▶ **O que acontece quando há um BO acidental?**
 - ▶ O program torna-se instável
 - ▶ O programa termina (crash)
 - ▶ O programa continua de forma aparentemente normal
- ▶ **Os efeitos colaterais dependem de:**
 - ▶ Quantos dados são escritos no final do buffer
 - ▶ Que dados (se é que algum) são reescritos
 - ▶ Se o programa tenta ler dados reescritos
 - ▶ Que dados acabam por substituir a memória que é reescrita
- ▶ **Depurar um programa que tem um bug desta natureza é difícil**
 - ▶ Pois os efeitos podem aparecer algumas linhas de código mais tarde

▶ 10

SS - Nuno Santos

2019



Porque são um problema de segurança?

- ▶ Porque podem ser explorados intencionalmente
- ▶ Podem permitir que um atacante execute o seu próprio código na máquina alvo
 - ▶ O objectivo é normalmente correr código com privilégios superuser (root)
 - ▶ ...o que é imediato se o servidor estiver a correr com UID 0
 - ▶ ...ou conseguido através de um **ataque de escalada de privilégios**
 - ▶ Paper importante (tornou estes ataques vulgares): Aleph One, "Smashing the Stack for Fun and Profit", Phrack 49-14.1996
- ▶ O objectivo também pode ser roubar dados
 - ▶ Pode ser chamado, neste caso, **buffer overread**
 - ▶ Exemplo, o famoso HeartBleed que permitia roubar material criptográfico da biblioteca SSL instalada em milhões de máquinas



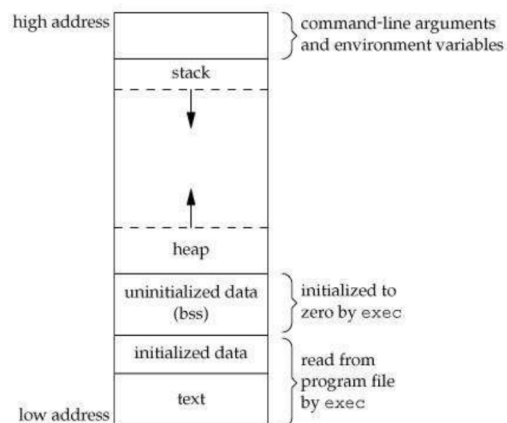
Overflow da stack e da heap



BO mais comuns na pilha e na heap

- Organização da memória virtual de um **processo**
- **Pilha:** guarda variáveis locais e estado de invocação das funções
- **Heap:** variáveis alocadas dinamicamente

Espaço de endereçamento virtual de um processo



► 13

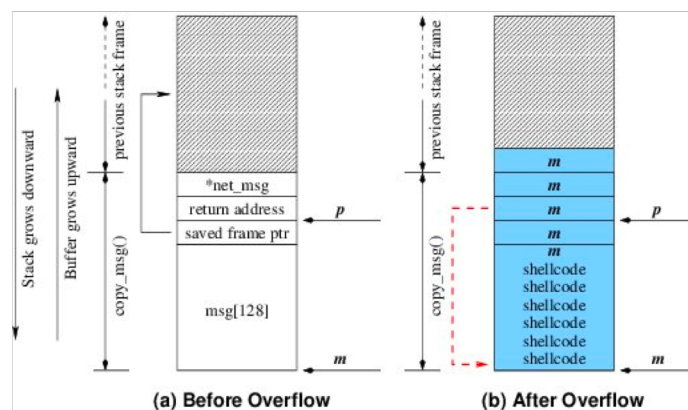
SS - Nuno Santos

2019



Stack smashing

- Quando um atacante consegue explorar um buffer overflow para obter execução de código malicioso



► 14

SS - Nuno Santos

2019



Heap overflow

- ▶ Programa vulnerável à modificação de dados alocados na heap

```
main(int argc, char **argv) {  
    int i;  
    char *str = (char *)malloc(4);  
    char *critical = (char *)malloc(9);  
    strcpy(critical, "secret");  
    strcpy(str, argv[1]); //heap overflow vuln.  
    printf("%s\n", critical);  
}
```

Heap



TÉCNICO
LISBOA

Integer overflow



Aspectos básicos sobre integer overflows

- ▶ O que acontece quando um inteiro com sinal é atribuído a um inteiro sem sinal?

```
unsigned x; int y = -10; x = y;
```

- ▶ Muitas vezes a semântica de operações com inteiros é complexa e os programadores não conhecem os resultados
- ▶ É um problema em várias linguagens, especialmente C/C++
 - ▶ Mas também acontece em linguagens *type safe* (Java, C#)

▶ 17

SS - Nuno Santos

2019



Podem surgir quatro tipos de problemas

- ▶ **Overflow**
 - ▶ Resultado de uma expressão excede o valor máximo do tipo
- ▶ **Underflow**
 - ▶ Resultado de uma expressão é menor que o valor mínimo
- ▶ **Erro de sinal**
 - ▶ Inteiro com sinal é interpretado como sem sinal ou vice versa
- ▶ **Truncagem**
 - ▶ Atribuir inteiro com um tamanho maior a um com tamanho menor

▶ 18

SS - Nuno Santos

2019



Alguns exploits possíveis

- ▶ Alocação de memória insuficiente
→ BO → execução de código do atacante
- ▶ Alocação excessiva de memória
→ Loop infinito → negação de serviço
- ▶ Ataque contra índice do array
→ Escreve por cima de bytes arbitrários em memória
- ▶ Ataque para contornar sanitização
→ causa BO → ...
- ▶ Erros lógicos
 - ▶ Ex. modificar variáveis para modificar o comportamento do programa

▶ 19

SS - Nuno Santos

2019



Conclusões

- ▶ Buffer overflows são vulnerabilidades muito sérias que resultam de incompatibilidade entre memória alocada a um tipo e o acesso a esse tipo
- ▶ Os buffer overflows mais comuns acontecem por problemas de programação que envolvem manipulação de memória na pilha e na heap
- ▶ Outra classe de overflows envolve operações relacionadas com inteiros

▶ 20

SS - Nuno Santos

2019



Referências e próxima aula

► Bibliografia

- [Correia17] Capítulo 5

► Próxima aula

- Vulnerabilidades em software: Corridas e validação de entradas