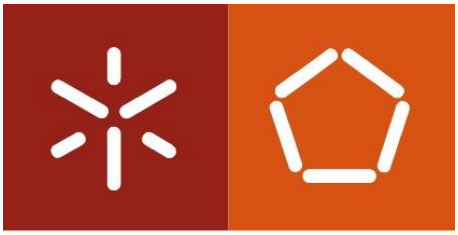


Universidade do Minho

Departamento de Informática

Mestrado em Engenharia Informática

Criptografia e Segurança de Informação

Engenharia de Segurança
Ficha Prática 9

Grupo Nº 3

Ariana Lousada (PG47034) Luís Carneiro (PG46541)
Rui Cardoso (PG42849)

17 de maio de 2022

1 Vulnerabilidade de codificação

1.1 Pergunta 1.1 - Common Weakness Enumeration (CWE)

Tendo por base o The CWE Top 25 de 2021 https://cwe.mitre.org/top25/,

1.1.1 Explique as características da Weakness no ranking correspondente ao número do seu grupo, em que linguagens (e tecnologias, se aplicável) são mais prevalentes, e quais são as suas consequências mais comuns. Dê exemplo (e explique) de código e de CVE que inclua essa Weakness.

A *weakness* no *rank* número 3 corresponde a *Out-of-bounds Read*. Esta vulnerabilidade consiste uma leitura fora dos limites(antes do início ou após o fim) de um determinado *buffer*.

Esta vulnerabilidade permite que o atacante leia informação sensível de outros locais que não sejam o *buffer* da memória ou até mesmo provocar um *crash*.

Um *crash* pode ocorrer quando o programa lê um valor que não é suposto, esperando por uma operação de *break*, como um valor NULL. Esta vulnerabilidade pode levar a que o programa leia uma quantidade demasiado grande de dados, o que acaba por provocar um *segmentation fault* ou um *buffer overflow*. O *software* pode também modificar ou calcular um índice que referencie um local na memória externo ao *buffer*. Uma leitura deste local de memória "calculado"pode levar a resultados inesperados ou indefinidos.

As linguagens mais propensas a este tipo de vulnerabilidades costumam ser linguagens de mais baixo nível que não contêm mecanismos na própria linguagem para impedir este tipo de leituras, como por exemplo o C e C++; o Java também apresenta vulnerabilidades deste tipo, apesar se tratar de uma linguagem de alto nível.

Um exemplo muito simples de código C no qual se pode observar esta vulnerabilidade é o excerto seguinte, no qual é acedido, neste caso, um valor de índice mais elevado que o limite superior de um array:

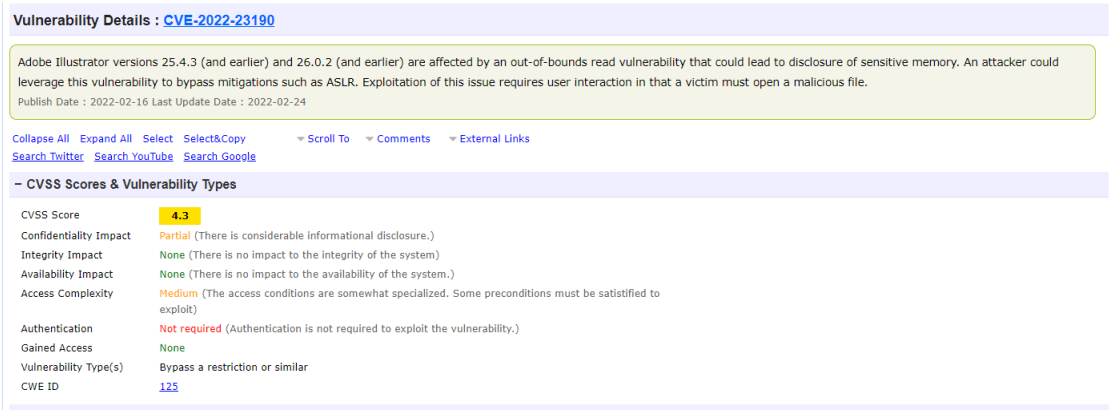
```c
// Program to demonstrate
// accessing array out of bounds
#include <stdio.h>
int main()
{
    int arr[] = {1,2,3,4,5};
    printf("arr[0] is %d\n", arr[0]);

    // arr[10] is out of bound
    printf("arr[10] is %d\n", arr[10]);
    return 0;
}
```

Output:

```
arr[0] is 1
arr[10] is -1786647872
```

Como se pode observar, o programa leu um valor "lixo"contido em memória.

Em termos de vulnerabilidades CVE, a seguinte figura expõe uma fraqueza *Out-of-Bounds-Read* do Adobe Illustrator detetada nas versões anteriores à 26.0.2. Esta vulnerabilidade pode levar ao acesso de memória sensível do sistema.



Vulnerability Details : CVE-2022-23190

Adobe Illustrator versions 25.4.3 (and earlier) and 26.0.2 (and earlier) are affected by an out-of-bounds read vulnerability that could lead to disclosure of sensitive memory. An attacker could leverage this vulnerability to bypass mitigations such as ASLR. Exploitation of this issue requires user interaction in that a victim must open a malicious file.

Publish Date : 2022-02-16 Last Update Date : 2022-02-24

Collapse All Expand All Select Select&Copy Scroll To Comments External Links
Search Twitter Search YouTube Search Google

– CVSS Scores & Vulnerability Types

CVSS Score	4.3
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	None (There is no impact to the integrity of the system)
Availability Impact	None (There is no impact to the availability of the system.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satistified to exploit)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Bypass a restriction or similar
CWE ID	125

Figura 1: CVE-2022-23190 - Adobe Illustrator

1.2 Pergunta P1.2

Considere os três tipos de vulnerabilidades: de projeto, de codificação e operacional. Apresente para cada um deles dois exemplos e discuta a dificuldade de correção.

Uma vulnerabilidade de projeto é introduzida durante a fase de projeto do software (obtenção de requisitos e desenho). Dois exemplos são a identificação incorreta das necessidades de segurança ao ignorar o requisito de não repúdio num sistema bancário e na criação de arquiteturas conceptuais que possuam erros de lógica ao misturar níveis de abstração no mesmo diagrama.

Uma vulnerabilidade de codificação é introduzida durante a programação do software, i.e., um bug com implicações de segurança. São exemplos o *buffer overflow* ao não restringir o tamanho máximo do *input* lido e o *Out-of-Bounds Read*, em que o *software* lê dados antes do início ou depois do fim do *buffer* pretendido.

Por último, uma vulnerabilidade operacional é causada pelo ambiente no qual o software é executado ou pela sua configuração. Estas podem ocorrer ao implementar o *software* de modo inapropriado (por exemplo numa máquina com um sistema operativo diferente do esperado) ou na inserção de falhas durante a manutenção ou a atualização ao introduzir código com *bugs*.

De uma maneira geral, as vulnerabilidades detetadas mais cedo no ciclo de vida de *software* são menos custosas e mais fáceis de corrigir, como é o caso das vulnerabilidades de projeto. Já as vulnerabilidades de codificação são mais difíceis de corrigir visto que o *software* pode já estar *deployed*, ou seja, para além da (difícil) deteção é também necessário efetuar a sua atualização rapidamente de modo a afetar o menor número de utilizadores. Finalmente, as vulnerabilidades operacionais tendem a ser muito específicas e podem ser tão simples como mudar um parâmetro de configuração do *software* ou exigir diversos testes de modo a identificar e corrigir o problema.

1.3 Pergunta P1.3

O que é que distingue uma vulnerabilidade dia-zero de outra vulnerabilidade de codificação que não seja de dia-zero?

As vulnerabilidades dia-zero destacam-se pelo facto de não serem conhecidas na comunidade de segurança informática, tais como as que não são dia-zero, mas sim num meio restrito, tal como estarem disponíveis à venda na *dark web* (sendo que quanto mais perigosa a vulnerabilidade, mais cara é) ou estarem na posse de um meio militar de um país.

Normalmente, estas vulnerabilidades são mais devastadoras que as normais, pois permitem atacar sistemas administrados por equipas competentes e com bons conhecimentos de segurança.

2 Referências

- CWE-125: Out-of-bounds Read : https://cwe.mitre.org/data/definitions/125.html
- Accessing array out of bounds in C/C++ : https://www.geeksforgeeks.org/accessing-array-bounds-ccpp/
- CVE-2022-23190 : https://www.cvedetails.com/cve/CVE-2022-23190/
- Common Mistakes in Architecture Diagrams : https://www.ilograph.com/blog/posts/diagram-mistakes/
- Out-of-bounds Read : https://cwe.mitre.org/data/definitions/125.html