# Buffer Overflow Report
# Epicode - Placeholder Buffer Overflow Test

Valentino Pizzi - pizziv97@gmail.com

**November 1, 2025**

## Contents

# Executive Summary

Test di come funziona un buffer overflow in un programma in C creato al momento

## Scope

The following targets were added to the scope:

- bof.c

## Vulnerabilities at a Glance

Table of the identified vulnerabilities:

| **Vulnerability Name** | **Severity** | $\Sigma$ |
|---|---|---|
| Buffer Overflow | Critical | 1 |
| **Total Vulnerabilities Found** | | 1 |

### Description

Buffer Overflow, non sanitizzando l'input dell'utente, un attaccante puo' sovrascrivere la memoria.

### Remediation

Una soluzione per limitare il numero dei caratteri letti e' impostare il parametro width dello specificatore:

`%[*][width][lenght]specifier`

width specifica il numero massimo di caratteri da leggere nell'operazione di lettura

### PoC

programma in C non sanitizzato:

```
include <stdio.h>

int main () {

char buffer [10];

printf ("si prega di inserire il nome utente:");
scanf ("%s", buffer);

printf ("Nome utente inserito: %s\n", buffer);

return 0;

}
```

**Practical Remediation**

programma in C sanitizzato:

```
include <stdio.h>

int main () {

char buffer [10];

printf ("Si prega di inserire il nome utente:");
scanf ("9s", buffer);

printf ("Nome utente inserito: %s\n", buffer);

return 0;

}
```