



Universidade do Minho  
Escola de Engenharia

## UNIVERSIDADE DO MINHO

---

### Guião – Aula 11

### VALIDAÇÃO DE INPUT

---

#### **Autores:**

Bruno Rodrigues: pg41066

Carlos Alves pg41840

## QUESTÃO 1.1

---

Analise o programa filetype.c que imprime no ecrã o tipo de ficheiro passado como argumento.

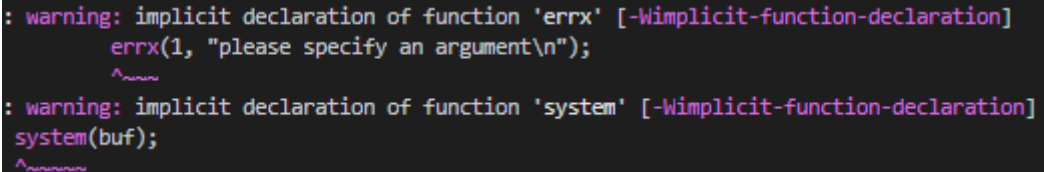
### QUESTÃO 1.1.1

Existem pelo menos dois tipos de vulnerabilidades estudadas na aula teórica de "Validação de Input" que podem ser exploradas. Identifique-as.

---

As vulneráveis que podemos referir, são:

- Falta de verificadores de tamanho nos inputs.
- O uso da função system(), pois com este podemos executar comandos na linha de comandos, e visto que não existe qualquer verificação no argumento inserido pelo utilizador, isto pode levar a um invasor executar diversos comandos que coloquem em causa a segurança, *realizar path traversal* e ainda o fato desta função utilizar variáveis do ambiente do processo-pai.
- O uso de snprintf em vez de sprintf é uma excelente escolha em termos de segurança, visto que sprintf() não verifica os limites do buffer.



```
: warning: implicit declaration of function 'errx' [-Wimplicit-function-declaration]
  errx(1, "please specify an argument\n");
  ^~~~~
: warning: implicit declaration of function 'system' [-Wimplicit-function-declaration]
  system(buf);
  ^~~~~~
```

### QUESTÃO 1.1.2

Forneça o código/passos/linha de comando que permitem explorar cada uma das vulnerabilidades identificadas na linha anterior.

---

Bastava um comando do género:

`./filetype "teste.txt;<comando>"` – Com o ponto e virgula poderíamos colocar diversos comandos.

Para fazer o *Path traversal* o processo é semelhante, apenas é necessário usar `/path`.

### QUESTÃO 1.1.3

O que aconteceria se o seu programa tivesse permissões *setuid root*?

---

O sistema na sua totalidade estaria comprometido (CIA – Confidencialidade, Integridade e Disponibilidade), pois ao ter permissão *setuid root* poderia executar o comando que desejasse. Podendo executar programas maliciosos, ter a capacidade de ler todos os ficheiros, alterar comandos do sistemas, como no exemplo acima com o ls, roubar ficheiros entre outros.

### QUESTÃO 1.2

---

Desenvolva um programa (na linguagem em que tiver mais experiência) que pede:

- valor a pagar,
- data de nascimento,
- nome,
- número de identificação fiscal (NIF),
- número de identificação de cidadão (NIC),
- número de cartão de crédito, validade e CVC/CVV.

Valide esse input de acordo com as regras de validação "responsável", apresentadas na aula teórica.

Resposta encontra-se no ficheiro ex2.py.