# A blue and white sign with textSistemas de Segurança de Software

## Mestrado em Engenharia Informática

# Segurança de Software

## Mestrado em Informática

# Class Project: Experiments with Fuzzer

Gustavo Henriques Nº 64361

Leonardo Monteiro Nº 58250

Maria Figueirinhas Nº 46494

**2. Fuzzing GNU Coreutils**

1. **Checking that the vulnerability exists**

Neste exercício começamos por correr os comandos que o enunciado mostrava. De seguida, experimentamos alguns dos comandos dados e depois, para explorar a vulnerabilidade metemos o seguinte comando:  
  
./date --date=’TZ="123"345" @1”’  
  
Com isto o terminal deixou de se comportar como devia, como é possível ver na imagem abaixo, concluindo assim que causamos um crash:

1. **Fuzz the date program**

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamenteNesta alínea fizemos os comandos que estavam no enunciado e o fuzzer começou a correr, como vemos na imagem:

1. **Confirm crash**

Neste exercício fomos buscar um dos crashes e testar se realmente dava um crash. Como podemos ver na imagem abaixo, o crash aconteceu mesmo:

1. **Locate the vulnerability**

Para este exercício corremos o comando gdb ./date no terminal para abrir o gdb. Dentro do gdb fizemos run --date='TZ="ETZ="1"'. Depois disto obtemos o crash dizendo “double free or corruption (out)” e, com o comando where percebemos que o crash occoreu quando o ficheiro date.c tenta chamar a função parse\_datetime que por sua vez no ficheiro parse\_datetime.y irá tentar fazer free. O crash está a acontecer porque estamos a tentar libertar memória que já foi libertada.

1. **Determine if the "unique" crashes are really "unique"**

Aqui tentamos o comando run –date=’TZ="m\\E"1="1"1’. Com este comando obtivemos exatamente o mesmo erro. O que significa que apesar de se chamarem unique crash todos estão a explorar a mesma vulnerabilidade.

1. **Use AFL\_HARDEN to check for bugs**

Para este exercício usamos o AFL\_HARDEN para tentar encontrar diferentes tipos de crashes que explorassem diferentes vulnerabilidades. Para isso fomos correr o mesmo comando no gdb que fizemos anteriormente e reparamos que a vulnerabilidade explorada foi a mesma.