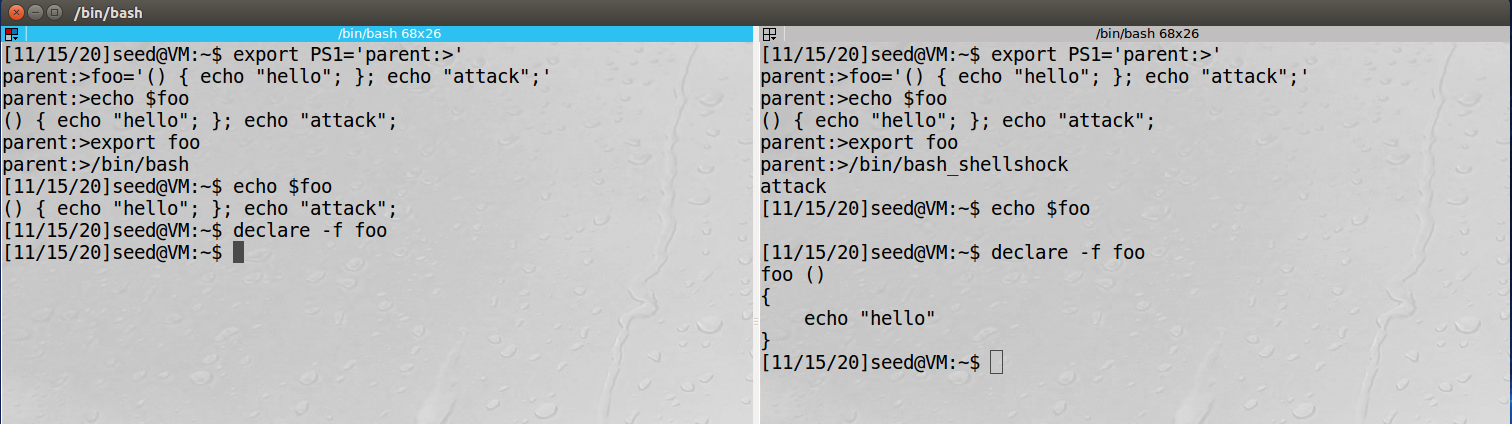
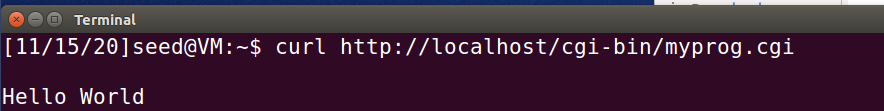
Exercicio 1:

# Guião 2.1)

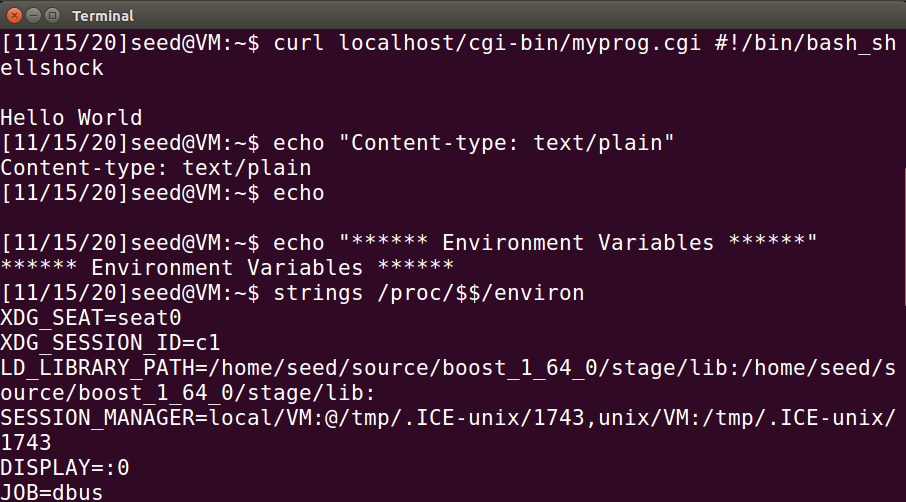
Esta experiência apresentada na figura, apresenta do lado esquerdo o caso de uma tentativa de ShellShock num Bash atualizado, e do lado direito uma tentativa do mesmo ataque no Bash\_Shellshock. A tentativa consiste em fazer uma função no bash, num processo pai, que possui código fora da função. No caso do Bash atualizado, este código é assumido como parte da variável de ambiente. No entanto o caso do Bash\_Shellshock, o código extra é deixado de fora da variável de ambiente, pelo que acaba por correr quando se troca de processo para o Bash\_ShellShock. O ataque é funcional e corre o código extra no Bash\_ShellShock, enquanto que o Bash não.

# Guião 2.2)



Na imagem acima, encontra-se o conteúdo do ficheiro a correr o servidor.

# Guião 2.3)



Para a os dados de um user remoto entrarem nas variáveis de ambiente, basta este utilizar o Bash\_Shellshock, e na string extra colocar uma declaração de variável de ambiente. Se o user remoto fizer curl com esses dados em header para um servidor vulnerável, será colocada uma variável de ambiente no servidor vulnerável, pois a primeira coisa a ser analizada são os headers. Como após isso é aberto o Bash\_Shellshock, acontece o mesmo problema do exercício 2.1 no qual o código extra é corrido, e é portanto criada uma nova variável de ambiente no servidor.

Exercício 2:

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamenteAtravés da estrutura de um ficheiro contendo um programa no formato de CGI desenvolvido nas questões anteriores, executou-se o ataque por parte da máquina virtual atacante, para a máquina virtual alvo. Foi necessário atribuir inicialmente um IP estático de modo a permitir a comunicação entre ambas as máquinas. Após a verificação da comunicação entre ambas, procedeu-se à definição de uma função, passando a mesma como argumento para o comando **“curl”**. Juntamente ao comando, foi acrescentado ao pedido o **header** de **content-type**: **text/plain** para ser possível obter uma leitura do conteúdo do ficheiro, bem como a localização do ficheiro ao qual o ataque é direcionado. Finalmente, especificamos o endereço de IP ao qual está afeta a máquina contendo o ficheiro para o ataque, bem como a localização do ficheiro com a estrutura CGI. O ataque é executado, conforme se apresenta nas imagens seguintes, através do comando: **curl -A ‘() {echo “stealin”;}; echo Content-type: text/plain; echo; /bin/cat /var/www/SQLInjection/safe\_home.php’ 10.0.4.5/cgi-bin/myprog.cgi**

Ao ser bem-sucedido, conseguimos obter as informações pretendidas, de que os dados usados pela aplicação web em causa para aceder à base de dados, são:

***Utilizador – root***

***Password - seedubuntu***

Exercício 3:

Não é possível. Para aceder ao ficheiro /etc/shadow, é necessário haver a permissão mais alta do sistema, **root**. Uma vez que o servidor corre em uma conta de utilizador que não é a **root**, esta conta não tem as permissões mais elevadas do sistema, logo o ataque não seria bem-sucedido para tentar obter o conteúdo do ficheiro.

Falar sobre pq e que elevar as permissões ñ funciona