

# Práctica 1 - Seguridad Informática

Pedro Tamargo

Juan José Tambo

25 de septiembre de 2020

## Índice

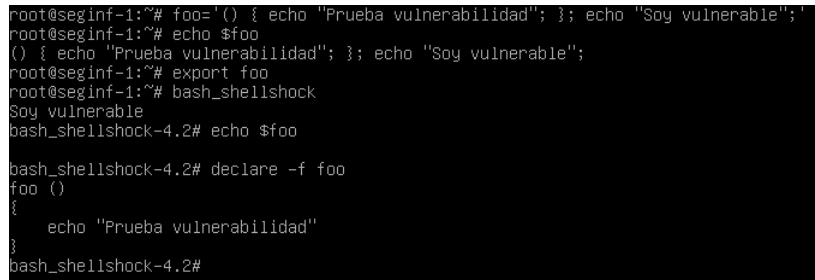
1. Tarea 1: Experimentar con las funciones en Bash	1
2. Tarea 2: Configuración de programas CGI	1
3. Tarea 3: pasar datos a Bash a través de las variables de entorno	1
4. Tarea 4: Lanzamiento del Ataque Shellshock	2
5. Tarea 5: Obtención de un Shell inverso a través de un ataque Shellshock	2

## 1. Tarea 1: Experimentar con las funciones en Bash

Para esta sección se ha creado una función `foo` con código extra y se ha ejecutado el siguiente código:

```
# Esta declaracion de funcion va precedida por las comillas
foo=() { echo "Prueba vulnerabilidad"; }; echo "Soy vulnerable";
echo $foo
export foo
bash_shellshock
```

Tras la ejecución de este código podemos observar como el intérprete *BASH\_SHELLSHOCK* es vulnerable (Figura 1) ya que ha ejecutado el código extra de la función `foo`.

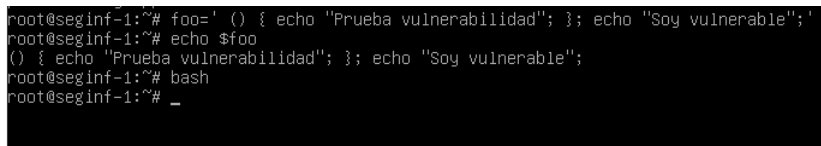


```
root@seginf-1:~# foo='() { echo "Prueba vulnerabilidad"; }; echo "Soy vulnerable";'
root@seginf-1:~# echo $foo
() { echo "Prueba vulnerabilidad"; }; echo "Soy vulnerable";
root@seginf-1:~# export foo
root@seginf-1:~# bash_shellshock
Soy vulnerable
bash_shellshock-4.2# echo $foo

bash_shellshock-4.2# declare -f foo
foo ()
{
    echo "Prueba vulnerabilidad"
}
bash_shellshock-4.2#
```

Figura 1: Intérprete afectado por el ataque *shellshock*

Si repetimos el experimento utilizando el intérprete *Bash* con la vulnerabilidad arreglada, se puede observar que al utilizar el código anterior no produce el mismo resultado que en el primer experimento (Figura 2).



```
root@seginf-1:~# foo='() { echo "Prueba vulnerabilidad"; }; echo "Soy vulnerable";'
root@seginf-1:~# echo $foo
() { echo "Prueba vulnerabilidad"; }; echo "Soy vulnerable";
root@seginf-1:~# bash
root@seginf-1:~# _
```

Figura 2: Intérprete **NO** afectado por el ataque *shellshock*

## 2. Tarea 2: Configuración de programas CGI

Hola

## 3. Tarea 3: pasar datos a Bash a través de las variables de entorno

Para enviar un *string* arbitrario al programa *CGI* se ha utilizado el siguiente *script*:

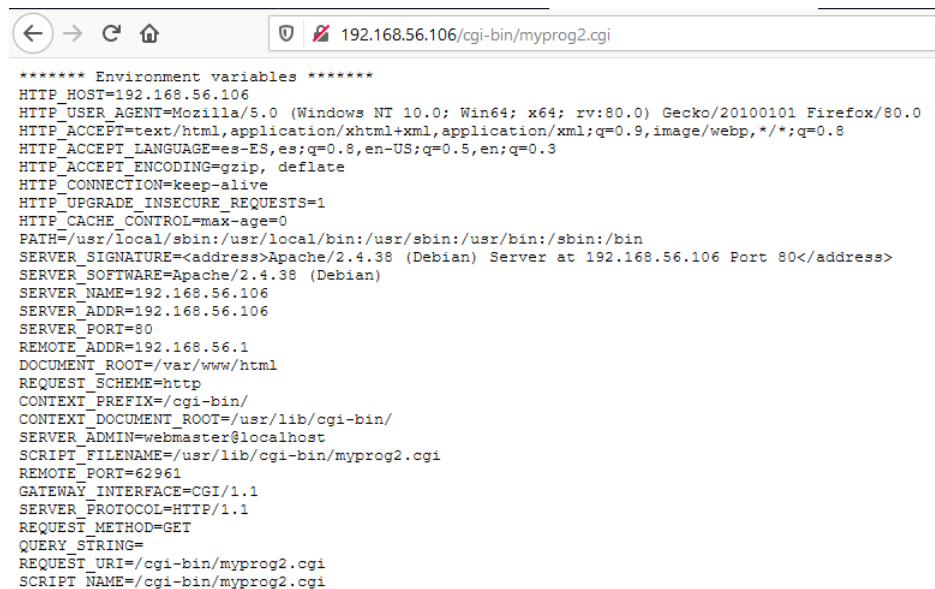
```
#!/bin/bash_shellshock
echo "Content-type: text/plain"
echo
echo "***** Environment Variables *****"
strings /proc/$$/environ
```

Este *script* muestra todas las variables de entorno de los procesos ejecutados. Si accedemos a la dirección: [http://IP\\_MV/cgi-bin/myprog2.cgi](http://IP_MV/cgi-bin/myprog2.cgi) se puede observar el resultado (Figura 3).

Para modificar el código de una de las variables de entorno se va a utilizar la cabecera *HTTP User-Agent*. Esta cabecera se modificará mediante el siguiente comando:

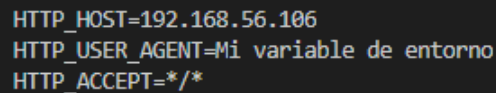
```
curl -A "Mi variable de entorno" http://192.168.56.106/cgi-bin/myprog2.cgi
```

Se puede observar que la respuesta del servidor contiene la variable de entorno *HTTP\_USER\_AGENT* pero con un valor distinto al ejemplo anterior (Figura 4).



```
***** Environment variables *****
HTTP_HOST=192.168.56.106
HTTP_USER_AGENT=Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0) Gecko/20100101 Firefox/80.0
HTTP_ACCEPT=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE=es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
HTTP_ACCEPT_ENCODING=gzip, deflate
HTTP_CONNECTION=keep-alive
HTTP_UPGRADE_INSECURE_REQUESTS=1
HTTP_CACHE_CONTROL=max-age=0
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE=<address>Apache/2.4.38 (Debian) Server at 192.168.56.106 Port 80</address>
SERVER_SOFTWARE=Apache/2.4.38 (Debian)
SERVER_NAME=192.168.56.106
SERVER_ADDR=192.168.56.106
SERVER_PORT=80
REMOTE_ADDR=192.168.56.1
DOCUMENT_ROOT=/var/www/html
REQUEST_SCHEME=http
CONTEXT_PREFIX=/cgi-bin/
CONTEXT_DOCUMENT_ROOT=/usr/lib/cgi-bin/
SERVER_ADMIN=webmaster@localhost
SCRIPT_FILENAME=/usr/lib/cgi-bin/myprog2.cgi
REMOTE_PORT=62961
GATEWAY_INTERFACE=CGI/1.1
SERVER_PROTOCOL=HTTP/1.1
REQUEST_METHOD=GET
QUERY_STRING=
REQUEST_URI=/cgi-bin/myprog2.cgi
SCRIPT_NAME=/cgi-bin/myprog2.cgi
```

Figura 3: Resultado del acceso al programa cgi



```
HTTP_HOST=192.168.56.106
HTTP_USER_AGENT=M
HTTP_ACCEPT=/*/*
```

Figura 4: Respuesta del servidor con la variable de entorno *HTTP\_USER\_AGENT* modificada

## 4. Tarea 4: Lanzamiento del Ataque Shellshock

Hola

## 5. Tarea 5: Obtención de un Shell inverso a través de un ataque Shellshock

Hola