

Fares Walid

Senior Cybersecurity Consultant
Part Time BugBounty Hunter
Content Creator



```
# Create buttons for keys
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']
in the following keys:
    button = tk.Button(root, text=key, width=5,
command=lambda k=key: key_pressed(k))
    button.pack(side=tk.LEFT)
```

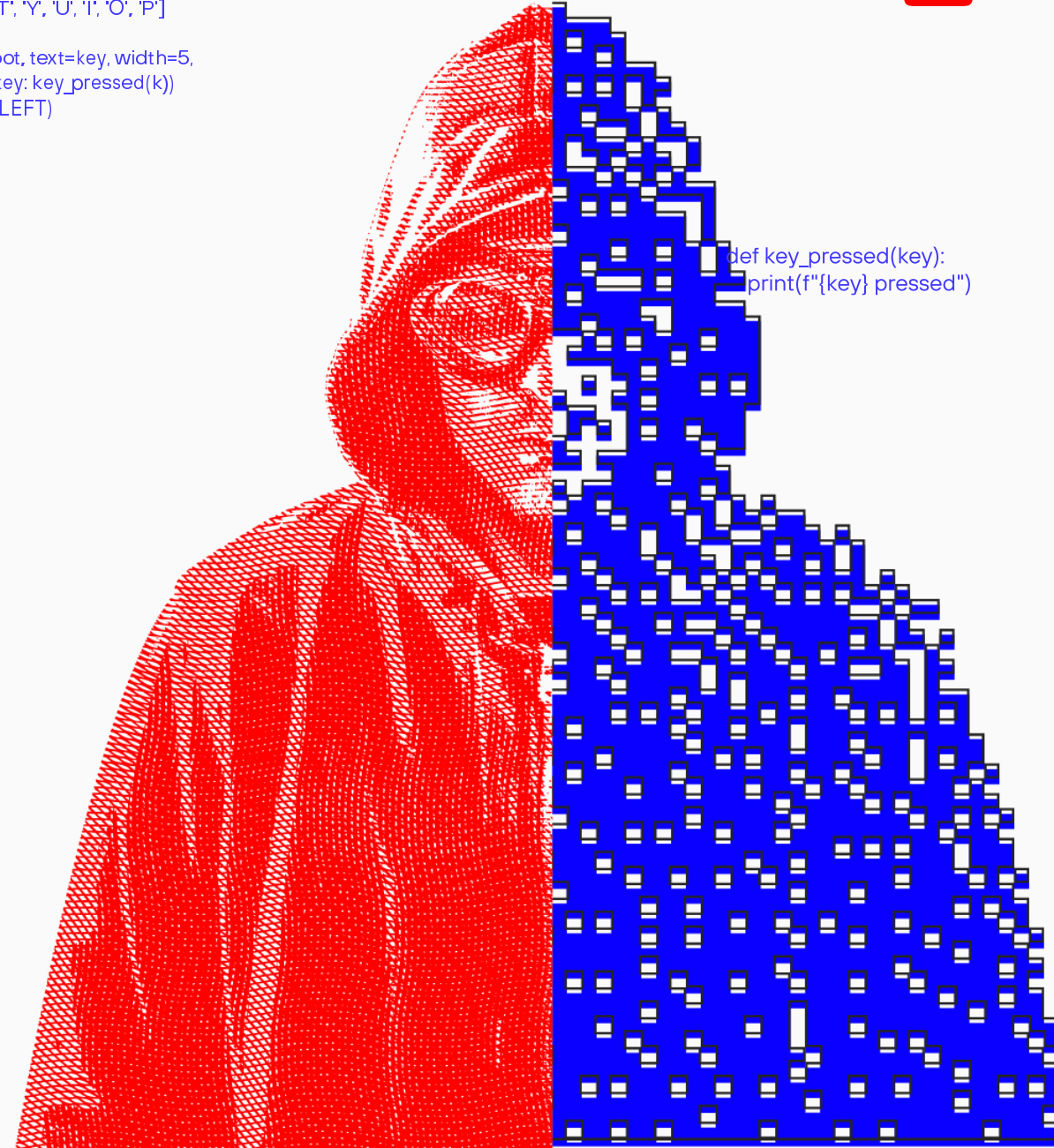
Logics & JS Files

```
def key_pressed(key):
    print(f"{key} pressed")
```

- Out-of-the-Box Approaches to Business Logic Bugs and JS Files Inspecting & Analysis



```
# Create the main window
root = tk.Tk()
root.title("Simple Keyboard")
```



About Me



Fares Walid (SirBugs – BugsV2)

- Senior Cybersecurity Consultant
- Part Time BugBounty Hunter
- Cybersecurity Content Creator
- Enjoying focusing on Logic Vulns & JS files in my hacking
- Love to lose my money in crypto (recently)

- X : SirBagoza
- YT: CyberBugz

01



Logic Vulnerabilities

- What are they, Why they are not well covered?
Techniques, Sharing some vulns ideas!

```
def key_pressed(key):  
    print(f"{key} pressed")
```

```
# Run the application  
root.mainloop()
```

```
# Create buttons for keys  
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']  
in the following keys:  
    button = tk.Button(root, text=key, width=5,  
        command=lambda k=key: key_pressed(k))  
    button.pack(side=tk.LEFT)
```



What are logics?

Business logic vulnerabilities occur when an application's design, functionality, or flow is exploited in ways that deviate from its intended use, allowing attackers to manipulate processes for unauthorized gain.

Unlike technical vulnerabilities (**like SQL injection**), they exploit the application's core functionality and user workflows.

These vulnerabilities often arise from flawed assumptions or overlooked conditions in the logic that governs business processes, making them harder to detect with traditional security measures.

Why not covered?

Business Logic Vulnerabilities are not usually covered. Because most of security teams often prioritize technical vulnerabilities like buffer overflows, code injection, or misconfigurations, leaving business logic vulnerabilities less scrutinized..

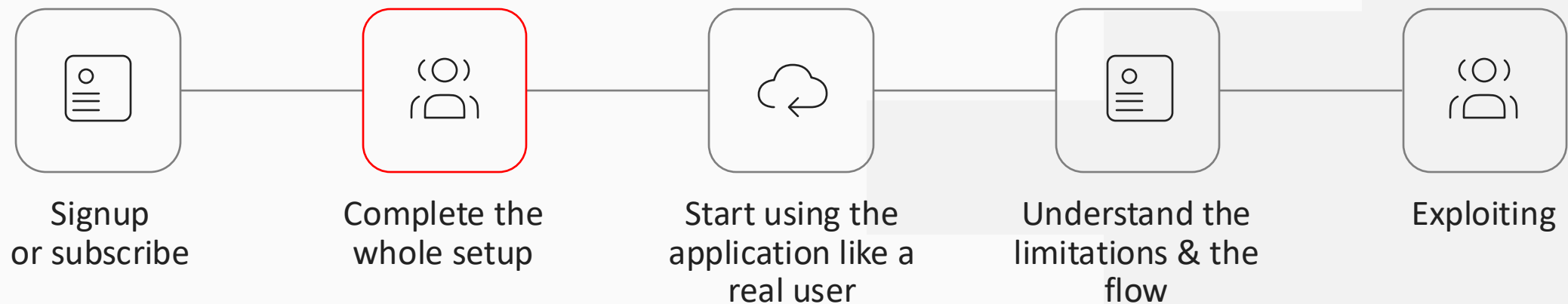
Not a lot of hackers focus on breaking the flow of the app itself, how it works, if there's smth limited, why is it?

And for the following additional reasons, it's not well covered in the hacking process:

- **Complexity of Business Rules:** Business logic is often intricate and specific to each organization.
- **Lack of Automation:** Automated tools and static analysis typically focus on common security issues like SQL injection or XSS, but they often don't catch business logic flaws.
- **Over-reliance on Developers:** Business logic vulnerabilities are often detected during development and testing.
- **Underestimated Risk:** Business logic vulnerabilities may be dismissed as less severe compared to other types of vulnerabilities like unauthorized data access or remote code execution, even though they can result in significant financial or reputational damage.

Completing flow

Completing the whole process, dealing with the app like a real user, someone who daily use the application, gives you a full understanding for the process and how it works – what can you do, what you can't, when can you do? Etc...



Qs to be asked

What

What this request is responsible for?
What is this request doing?

Never miss what a request is doing. It's in the process, so it's doing smth!

Why

Why this request is here? Why not before or after?

Breaking In

Whereeeee & Howww cannnn IIII breakkkkkk innnnnn?

NOs Detection

When faced with blockers or functions saying 'No, you cannot do that,' it's my turn to ask: Why not?

Is it related to the plan? Is it because of limitations? Or is it against the policy? And if that's the case, what would be the impact of breaking this rule? What consequences are we looking at?

- You can't delete this file before doing a check-in
- You can't do this actions because you are in the free plan
- You can't access the user's information before his acceptance
- You can't make new rides before your papers get verified
- You can't contact or access the owner only if he created a case
- You can't rate someone before the lesson is ended

AI Thinking

AI Could be helpful in generating some cases too, If you well described the case you have.

Hi currently I'm doing a penetration testing for an application of something like Uber about people taking rides and something like that. I will give you the requests in the whole flow and you give me some business logic vulnerabilities ideas to try.

First thing the user said the location and the point that he wants to go to then in the middle of the process, he can change the fear add another checkpoint in the right and when he search for a driver, some drivers send him offers and he can choose from them And if he accepted a driver, this driver now gets access to his personal data to contact him then they meet up then they complete the right and the user now can vote for the ride and can send some tip for the driver.

Give me direct and clear answers ideas only not too much talk

Prompt

1. Location Manipulation:

- Test if a user can manipulate their pickup/ dropoff location after the ride has been accepted by the driver, affecting fare calculation or altering the driver's route.

2. Fare Modification:

- Check if the user can change the fare after the ride request has been confirmed and the driver is en route, potentially leading to ride fare exploitation.

3. Checkpoint Addition Exploit:

- Attempt to add multiple checkpoints during the ride (even after accepting the driver's offer) to see if it can modify the fare or increase the ride duration unfairly.

4. Driver Offer Exploitation:

- Test if a user can view and accept an offer from multiple drivers simultaneously, and possibly exploit this to get the cheapest option or cause a driver to lose time by rejecting later.

5. Personal Data Exposure:

- Investigate if the driver can access the user's personal data (phone number, address, etc.) prematurely or if there is a race condition where the user's data is accessible before the driver's acceptance.

6. Driver Rating Manipulation:

- Check if users can manipulate or repeatedly submit ratings and tips for a ride, potentially inflating ratings or giving excessive tips.

7. Tip Abuse:

- Test if a user can alter or undo a tip after the ride has been completed, potentially manipulating the final payment to the driver.

8. Driver Data Persistence:

- Check if the driver can access the user's data beyond the scope of the ride (e.g., seeing future bookings or history), which would be a data leakage issue.

9. Ride Completion Process:

- Test if the user can mark a ride as "completed" without the driver's confirmation or if they can give a false vote and/or tip without completing the ride.

10. Fake Driver/Impersonation:

- Test if a user can impersonate a driver or send false offers by manipulating the user-driver relationship through API requests.

11. Fare Calculation Bug:

- Try manipulating variables like distance, time, or traffic conditions (if applicable) in the fare calculation process to ensure there's no logical flaw in fare estimation.

Answer

Vulns Disclosure

Payment Bypass (Price Lowering)



```
1 POST /api/v1/apply/change HTTP/2
2 Host: Vulnerable.com
3 Cookie: x=1;
4 .....
5
6 {"lessons":4, "tutor_id": 143454}
```



```
1 POST /api/v1/apply/change/order HTTP/2
2 Host: vulnerable.com
3 Cookie: x=1; ...
4 ....
5
6 {"order_id":123456, "lessons":3}
```



```
1 HTTP/2 200 OK
2 Content-Type: application/json; charset=utf-8
3 ....
4
5 {"order_id":123456}
```

By creating a new order, taking the lessons and tutor, calculating the price and reflecting an order_id

We can edit how many lessons regarding this order_id too from another request

```
1 POST /api/v1/order/issue HTTP/2
2 Host: vulnerable.com
3 Cookie: x=1; ...
4 ....
5
6 {"order_id":123456}
```

```
1 POST /api/v1/order/complete HTTP/2
2 Host: vulnerable.com
3 Cookie: x=1; ...
4 ....
5
6 {"order_id":123456} //Checking how many lessons
```

After paying, a new request checking if I paid or not, and how many lessons linked to the order_id gets issued

Vulns Disclosure

Payment Bypass (Price Lowering)

```
1 POST /api/v1/order/complete HTTP/2
2 Host: vulnerable.com
3 Cookie: x=1; ...
4 ....
5
6 {"order_id":123456} //Checking how many lessons
```

Ok, Now what If I **Intercepted** this request before it add to me the lessons linked to this **order_id**,

```
1 POST /api/v1/apply/change/order HTTP/2
2 Host: vulnerable.com
3 Cookie: x=1; ...
4 ....
5
6 {"order_id":123456, "lessons":40}
```

& Sending the following request to add more lessons (editing my order_id lessons) ??????????????

Vulns Disclosure

Payment Bypass (Price Lowering)



Shout-out to @0xa1mn



#281 Payment Bypass & Price Lowering (Ability to buy too many lessons with lower price)
To: High

Vulns Disclosure

Pos, Neg Voting

Rating is one of the most underrated functionalities in the applications. It's available these days too much in different applications:

- User rating a driver after the ride is finished: **Indrive, Uber, Careem, Ego**
- User rating a tutor/course after the lesson is finished: **Udemy, Preply**
- User rating a movie/clip after watching: **warner bros**

What about submitting Positive or Negative ratings?

Default rating is: **1-5**, what if we submitted **50**?

Default rating is: **1-5**, what if we submitted **-50**?

That can increase or decrease the rating of the profile.



☐
#2125047 Unlimited fake rate to driver's profile in city to city, Affected endpoint
`/api/v1/reviews/order/<ID>/passenger`
To: inDrive • ■■■■ Low
August 27, 2023

☐
#2125049 Unlimited fake rate to the passenger in city to city, Affected endpoint
`/api/v1/reviews/ride/<ID>/driver`
To: inDrive • ■■■ Medium
August 27, 2023

```

1 POST /api/v1/reviews/ride/[redacted]/driver HTTP/2
2 Host: intercity-3.eu-east-1.indriverapp.com
3 X-City-Id: 9415
4 Authorization: Bearer [redacted]
5 Content-Type: application/json; charset=utf-8
6
7 {"message":"Prince","rating":5}
  
```

Vulns Disclosure

Requests & Flow Arrangement (Returning Shipments)

Normally in a shipment application, we can make a return only after we receive the shipment.

But is it accepted to make a return for the shipment before we even receive it? It's not shipped yet?

Normally the app said **NO** in the UI. But via the request?

Luckily it was vuln to IDOR too, Since the shipment referrer was an Int, It was easy to make a returns to any shipment.

Forcing changing the shipment status

From:

- Pending
- Out for delivery
- Shipped
- Delivered

To: ▪ Returning

☐ ● #2965: IDOR chained with Business Logic Error led to return any shipment before delivery 3 days ago

To: [redacted] • [redacted] High

Vulns Disclosure

Student Leaking Answers (through helper mode)

In a class / school application, The teacher can make 2 kinds of exams.

- **Helper Mode**
- **Direct Exam**

Helper Mode: This mode is for the beginner students, the teacher can activate or disable it for any exam. It shows to the student the right answer if he clicked on a wrong one.

```
1 POST /api/v1/exam/question/23489689734/help_answer HTTP/2
2 Host: vuln.com
3 X-City-Id: 9415
4 Authorization: Bearer [REDACTED]
5
6 {"answers":"....."}
```

```
1 POST /api/v1/exam/question/23489689734/answer/submit HTTP/2
2 Host: vuln.com
3 X-City-Id: 9415
4 Authorization: Bearer [REDACTED]
5
6 {"answers":"....."}
```

Vulns Disclosure

Student Leaking Answers (through helper mode)

If the helper mode is **not activated**, the student will submit the answer directly.

- `/api/v1/exam/question/xxxxxx/answer/submit`

The helper endpoint will not be called. **But what about Intercepting and sending a request with the question Id in the helper endpoint?**

- `/api/v1/exam/question/xxxxxx/help_answer`

Helper Mode: will show the correct answer, Then we can send it in the answer submit endpoint.

```
1 POST /api/v1/exam/question/23489689734/answer/submit HTTP/2
2 Host: vuln.com
3 X-City-Id: 9415
4 Authorization: Bearer [REDACTED]
5
6 {"answers":"....."}
```

Vulns Disclosure

GraphQL Bypassing Admin Approval

```
{
  "operationName": "RemoveUserWithAdminApproval",
  "variables": {
    "userId": "00ucj2h0htHvreja1d7"
  },
  "query": "mutation RemoveUserWithAdminApproval($userId: String!) {\n  removeUserWithAdminApproval(userId: $userId)\n}"
}
```

```
{
  "operationName": "RemoveUser",
  "variables": {
    "userId": "00ucj2h0htHvreja1d7",
    "email": "bvgza@yahoo.com",
    "isActiveUser": true
  },
  "query": "mutation RemoveUser($userId: ID!, $email: String, $isActiveUser: Boolean) {\n  removeUser(userId: $userId, email: $email, isActiveUser: $isActiveUser)\n  __typename\n}"
}
```

Logic vulns are not related to a specific technology, api, server, **NO**.

You can exploit it anywhere, anytime, Just the right Idea!

Hi admin admin,

mango tango (bag0za@outlook.com) has proposed to change the user removal settings.

Please review the proposed changes to approve or reject them.

USER SETTINGS CHANGES

Settings type	User Remove Settings
Approval steps	Requires Admin Approval Requires at least 1 admin approver

APPROVE

REJECT

- ☐ #2335 Logic workflow breaking allowing the Administrator to bypass removing users function approval request

August 6, 2024

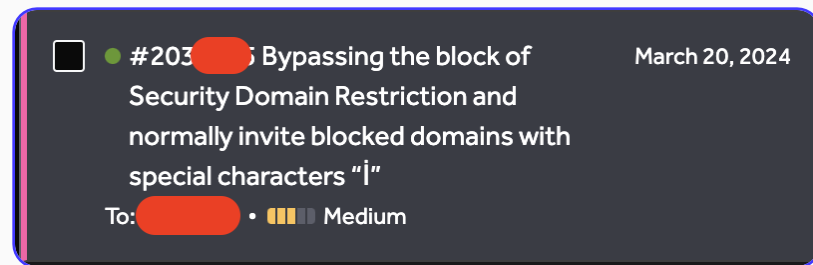
To: • Medium

Vulns Disclosure

Special Chars (Bypassing Filters)

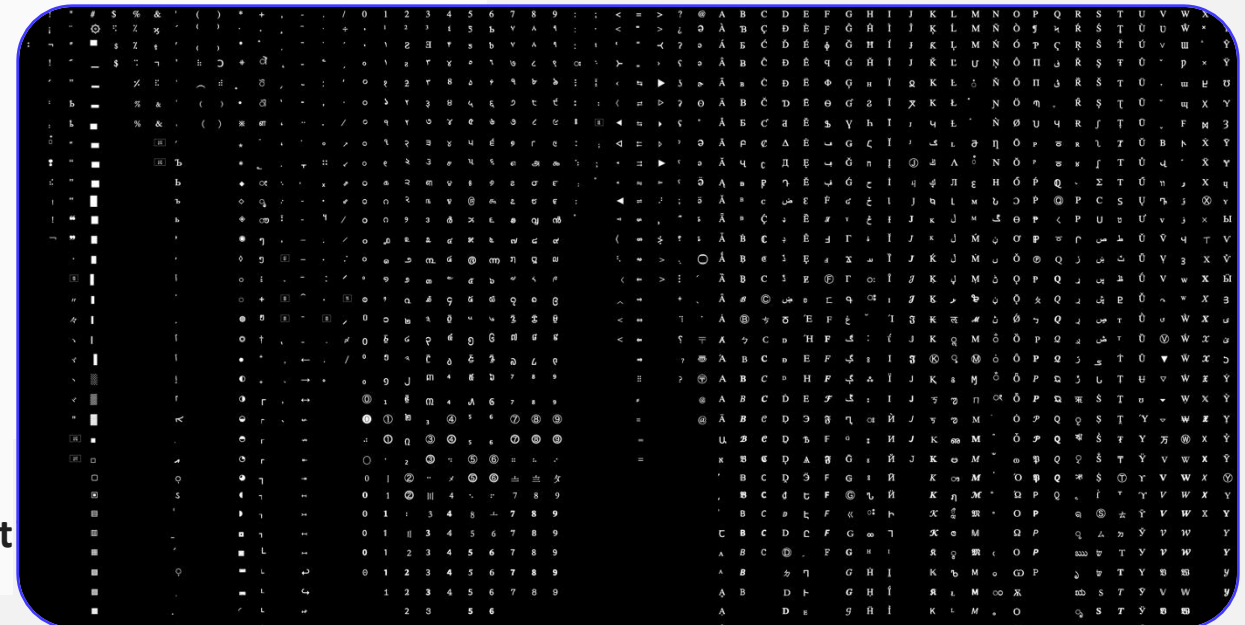
Special Characters Injection is being used for bypassing limitation too.

In some recent cases, I've been seeing smth like: "blacklist a **domain, email, ip**", so what if I was able to invite a blacklisted email, domain?



But since these functionalities, maybe most of the time will check for identical texts .. We can bypass!

- I usually use keys from: <https://0xacb.com/normalizat>



Vulns Disclosure

Special Chars (Bypassing Filters)

If you want to find out Why this happens in a little bit depth, check the following write-up:



Special Characters Manipulating in BugBounty

Special Characters in BugBounty Hunting Write-up

Aug 13, 2023 473 5



INCLUDED							
1	2	3	4	5	6	7	8
9	0	!	"	#	\$	%	&
'	(	)	*	+	,	-	.
:	;	<	=	>	?]		

JS Files

- Analyzing & Inspecting JS Files
For fun & Profit for BB Hunters

JS Files are your friends!

```
def key_pressed(key):  
    print(f"{key} pressed")
```

```
# Run the application  
root.mainloop()
```

```
# Create buttons for keys  
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']  
in the following keys:  
    button = tk.Button(root, text=key, width=5,  
        command=lambda k=key: key_pressed(k))  
    button.pack(side=tk.LEFT)
```



JS Files Importance

Why JS Files are **very very very very very very very** Important in your testing?

It's the treasure of your findings, the non-duplicated findings lead, the strong vulns helper

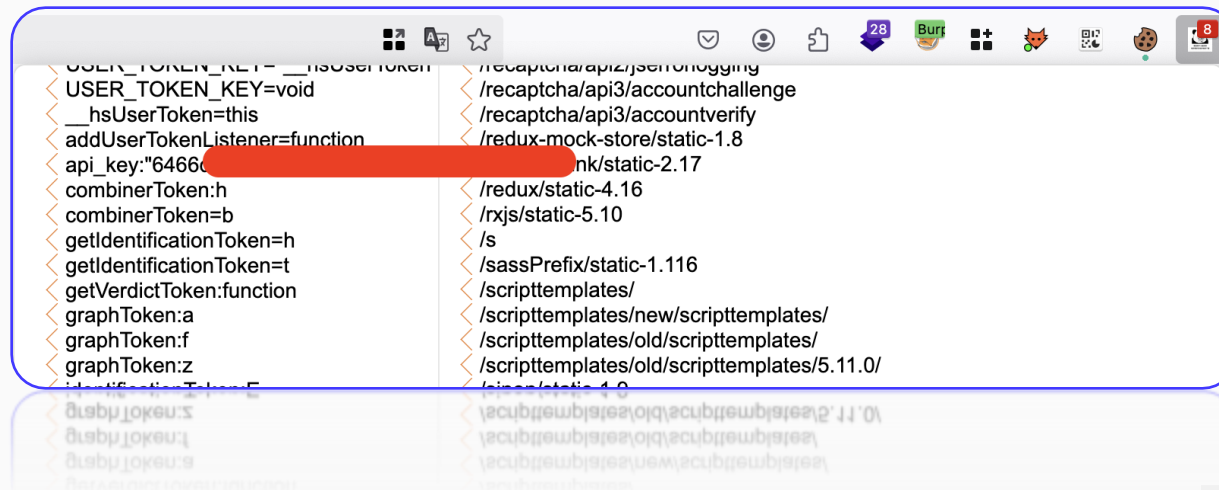
- All Endpoints Disclosure (Hidden from you)
- Hardcoded Credentials
- Leaked API Keys & Tokens
- Credentials Encryption Process

Each case of these points has its own slides to discuss.

But is exploring & exploiting JS Files really hard? **I don't think so...**

Finding Keys?

Find something



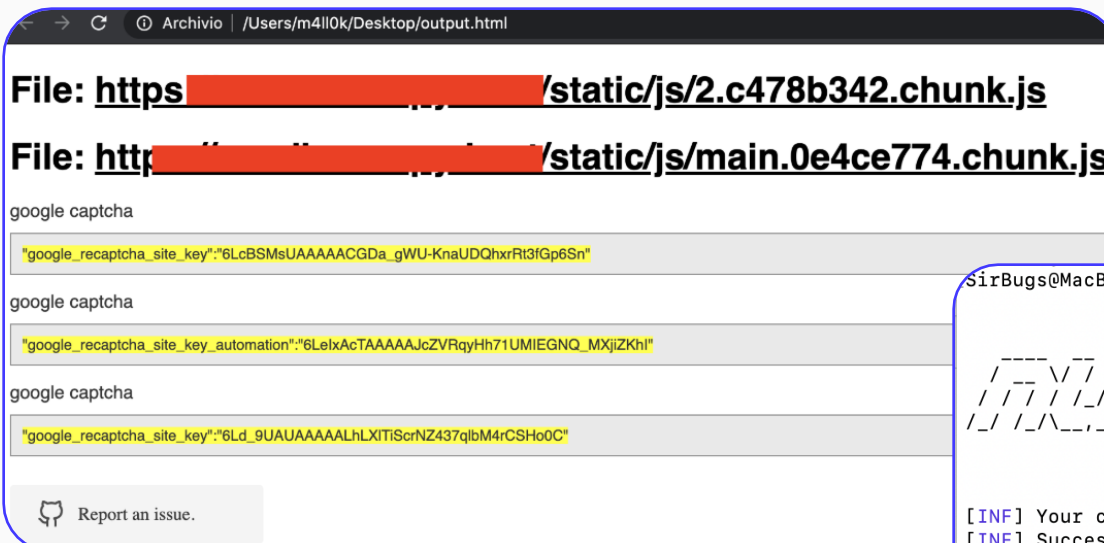
- Get it from:
<https://addons.mozilla.org/en-US/firefox/addon/findsomething/>



FindSomething
by residual.laugh

Finding Keys?

Nuclei & Secret Finder



- <https://github.com/m4ll0k/SecretFinder>
- <https://github.com/projectdiscovery/nuclei>

nuclei-templates / http / exposures /

ghost chore: sign templates e495bae · yesterday History

Name	Last commit message	Last commit date
..		
apis	chore: sign templates	3 months ago
backups	chore: sign templates	yesterday
configs	chore: sign templates	last week
files	chore: sign templates	last month
logs	chore: sign templates	3 months ago
tokens	chore: sign templates	last month

```
SirBugs@MacBook-Pro Downloads % nuclei -u https://example.com/ -tags exposure -s low,medium,high,critical
```

```

  ____  _
 / ___|| | | |
| |___| |_| |
 \___|_||_|_|_|

v3.3.7

projectdiscovery.io

```

```

[INF] Your current nuclei-templates v10.1.1 are outdated. Latest is v10.1.2
[INF] Successfully updated nuclei-templates (v10.1.2) to /Users/SirBugs/nuclei-templates. GoodLuck!
[INF] Current nuclei version: v3.3.7 (outdated)
[INF] Current nuclei-templates version: v10.1.2 (latest)
[WRN] Scan results upload to cloud is disabled.
[INF] New templates added in latest release: 52
[INF] Templates loaded for current scan: 668
[INF] Executing 663 signed templates from projectdiscovery/nuclei-templates
[WRN] Loading 5 unsigned templates for scan. Use with caution.
[INF] Targets loaded for current scan: 1
[INF] Templates clustered: 161 (Reduced 142 Requests)

```

Confirming Keys

Recently I've been using **secf.ai** for confirming all of the secret keys.

They have a very good KeysKit (+750 API Keys Calls)

SecF KeysKit

Securely validate API keys, tokens, and secrets across multiple providers with our comprehensive service. Ensuring robust security measures at every level.

767
Templates

ZeroBounce

ZeroBounce

ZeroBounce is an email validation and deliverability platform helping more than 200,000 customers reduce bounces and land their emails in the inbox. Their services include email validation, email scoring, email testing, sender reputation monitoring, an...

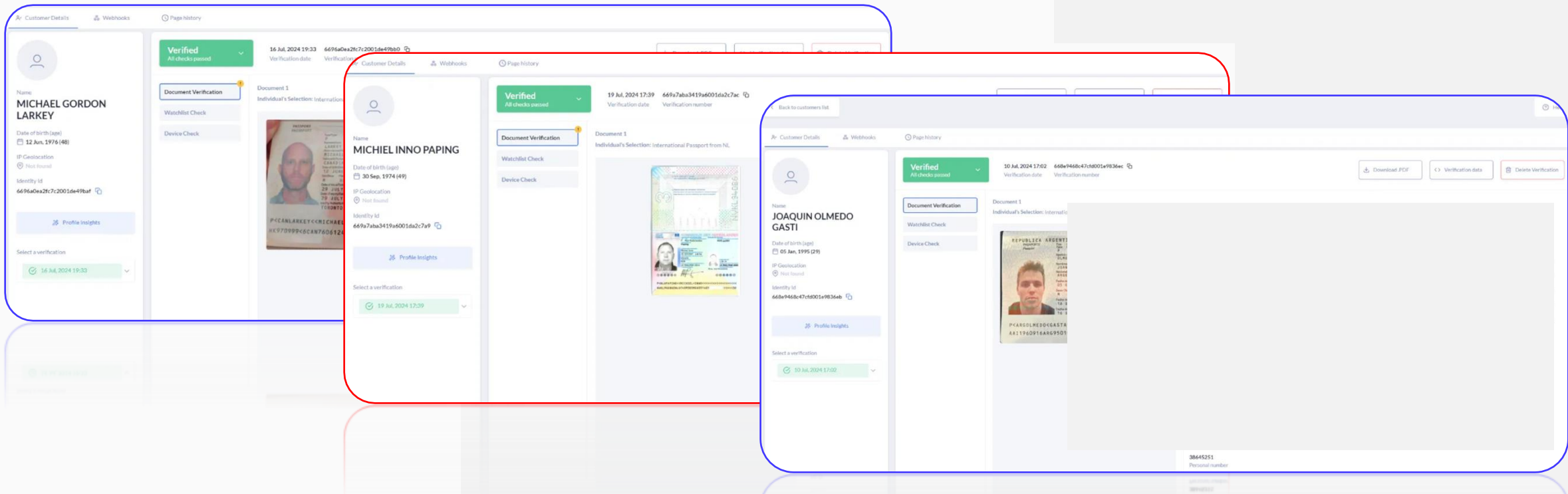
Command

```
$ curl -X GET "https://api.zerobounce.net/v2/activity?email=$Email&api_key=$API_KEY"
```

Shout-out to **Hossam Mesbah**
& The great team of SecF.AI

To access and start using the app, you apply with your NID + PII, they check it with a 3rd-party to confirm.

Just with applying their JWT in response in a login request, I accessed:



Creds Encryption

JS Too is being used to encrypt the credentials in some applications

You can Use the devtools In chrome to check which file exactly does this encryption process to be able to test the application.

It's too much helpful too in testing `postmessage()`

Login

Encrypt

Encrypted Username:
U2FsdGVkX1/QqZTZANKIXp5l6VV3N9cuuJzIgoSd2bk=

Encrypted Password:
U2FsdGVkX198UPwaR4PczLXcAJ7D95s760ASokWizwY=

Decrypt

Decrypted Username: John Doe

Decrypted Password: 12345678

Decrypted Password: 12345678

Decrypted Username: John Doe

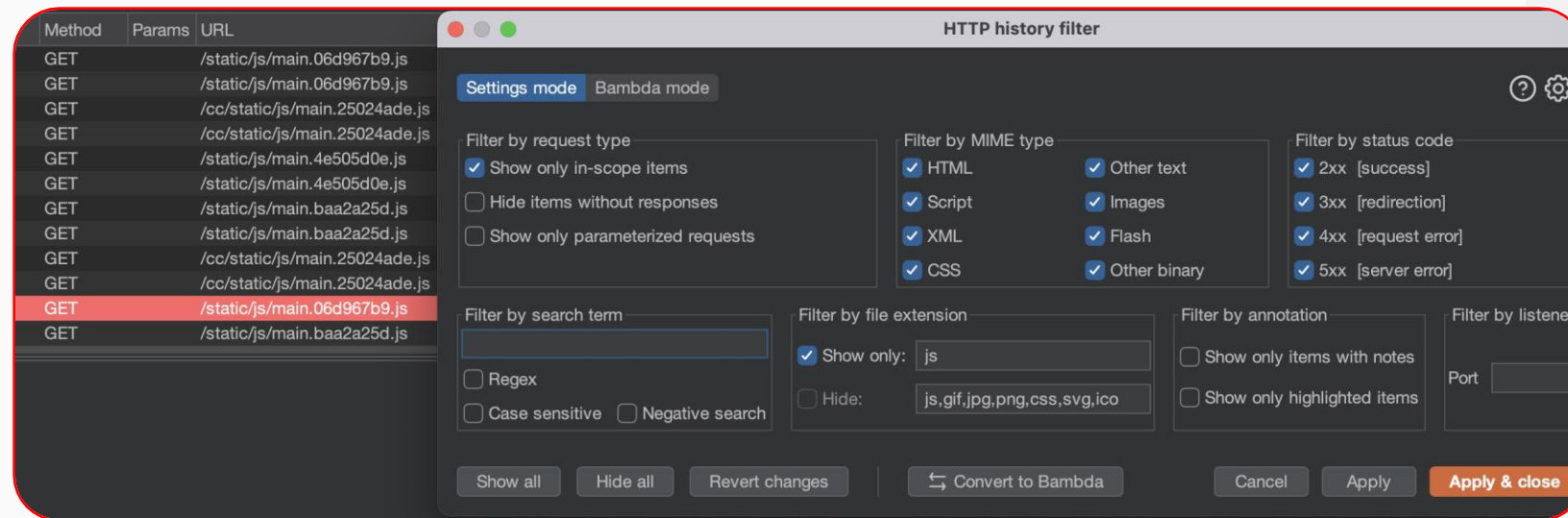
Too many JS Files

Burp History .js Files

Most of the time, I collect the basic JS Files in the application.

Any way it will be called because it's being used into the pages, requests being issued from it.

Just a simple filter to show only the **.js files**, and select all + Copy URL



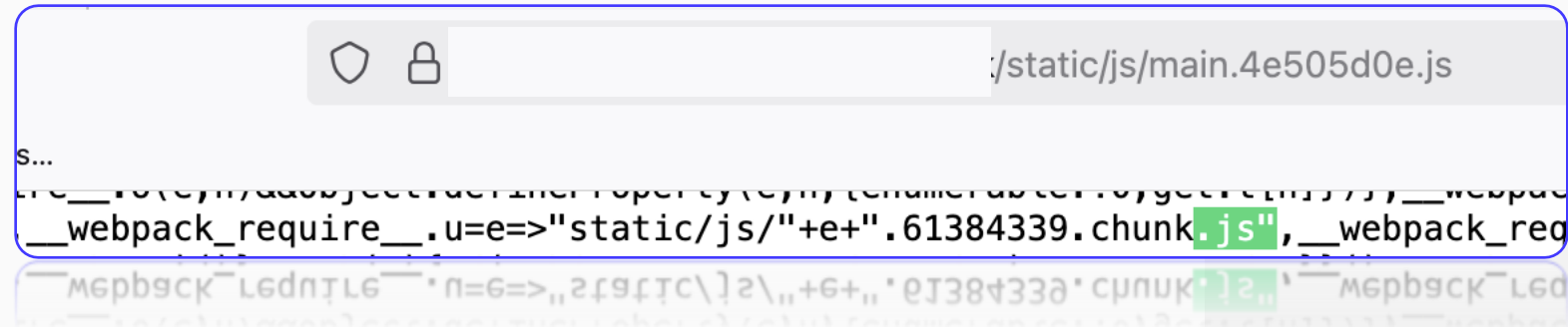
Too many JS Files

JS Files Into JS Files

If you didn't know, Some JS Files are calling JS Files!

As we see in the following SS, the file: **4e505d0e** is calling **61384339**

Maybe this doesn't appear in your burp history, because it wasn't used, So make sure you cover it.



I usually search in the mentioned tools for **.js** files.

- `cat * | grep "\.js"`

- | httpx

- | httpx -mc 200

- <https://github.com/SirBugs/JsValidator>

Sometimes it has an old or removed endpoints and they are still active in the backend server.

```

/public/PCOM/assets/index.4004ebf7.js
/public/PCOM/assets/index.446abf9b.js
/public/PCOM/assets/index.463d0d4f.js
/public/PCOM/assets/index.472b6624.js
/public/PCOM/assets/index.4ff6ca45.js
/public/PCOM/assets/index.58b563bb.js
/public/PCOM/assets/index.59695eaf.js
/public/PCOM/assets/index.630e3d70.js
/public/PCOM/assets/index.66ad1759.js
/public/PCOM/assets/index.680f2cb0.js
/public/PCOM/assets/index.6cca8f50.js
/public/PCOM/assets/index.788ad9a1.js
/public/PCOM/assets/index.79a04a06.js
/public/PCOM/assets/index.992f5932.js
/public/PCOM/assets/index.9ee23528.js
/public/PCOM/assets/index.a251a234.js
/public/PCOM/assets/index.a33caba9.js
/public/PCOM/assets/index.a4b830e7.js
/public/PCOM/assets/index.aac7bd3d.js
/public/PCOM/assets/index.b43abab1.js
/public/PCOM/assets/index.c0a16b60.js
/public/PCOM/assets/index.c2b1e944.js
/public/PCOM/assets/index.c390bc2b.js
/public/PCOM/assets/index.c5ecb4b5.js
/public/PCOM/assets/index.e6faa937.js
/public/PCOM/assets/index.e7be28d9.js
/public/PCOM/assets/index.e8c374be.js
/public/PCOM/assets/index.e8ea2ecb.js
/public/PCOM/assets/index.esm-7efde3e0.js
/public/PCOM/assets/index.esm-legacy-8e680285.js
/public/PCOM/assets/index.esm-legacy-ca921cf.d.js
/public/PCOM/assets/index.esm.8e494e23.js
/public/PCOM/assets/index.f07679bb.js
/public/PCOM/assets/index.f0789976.js
/public/PCOM/assets/index.fe59de37.js
/public/PCOM/assets/index2-613e7545.js
/public/PCOM/assets/index2-88f723ca.js
/public/PCOM/assets/index2-legacy-8884e297.js
/public/PCOM/assets/index2-legacy.180aaefc.js
/public/PCOM/assets/index2-legacy.9bbd7660.js
/public/PCOM/assets/index2.429b94d1.js
/public/PCOM/assets/index2.4c99644b.js
/public/PCOM/assets/index2.912d1289.js
/public/PCOM/assets/index2.b233465e.js
/public/PCOM/assets/index3-legacy.398bf2f6.js

```

[illegible]

Too many JS Files

Additional Scope (Old JS Files)

Additionally, If you have JS Files with no hashes (**not called main.834jf98j.js**)

It's just normally

- **main.js**
- **index.js**

Using the wayback, You can find old versions of this JS File!

Contains too many Endpoints (**got removed but still active in back-end**),
Tokens/Keys didn't get revoked (**Just removed**)

INTERNET ARCHIVE

WayBackMachine

Explore more than 916 billion [web pages](#) saved over time

Enter a URL or words related to a site's home page

We can sometimes fuzz for **.js** files, But not the ones with hashes.

- **ffuf -e js**

- **ffuf -u https://google.com/bla/bla/bla/FUZZ.js**

```
SirBugs@MacBook-Pro Downloads % ffuf -u [REDACTED]M/assets/FUZZ -w ~/Desktop/wordlists/onelistforallshort.txt -e js -recursion -mc 200
v2.1.0-dev

:: Method      : GET
:: URL         : [REDACTED]M/assets/FUZZ
:: Wordlist     : FUZZ: [REDACTED]onelistforallshort.txt
:: Extensions  : js
:: Follow redirects : false
:: Calibration  : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200
```

JS Mapping

Source Mapper

If you didn't know, the JS .map files can lead us to the source of React for example.

Giving us the ability to extract all the endpoints in a clear view.

You can find the map files in:

- The start/end of the files
- Try adding .map after .js
- Try checking the wayback for old JS files you may use .map for it if it doesn't has hash

```
doi@asov:~$ ./sourcemap -output dhbsrc -url https://hub.docker.com/public/js/client.356c14916fb23f85707f.js.map
[+] Retriving Sourcemap from https://hub.docker.com/public/js/client.356c14916fb23f85707f.js.map
[+] Read 23045027 bytes, parsing JSON
[+] Retrieved Sourcemap with version 3, containing 1828 entries
[+] Writing 9076765 bytes to dhbsrc/webpack:/js/client.356c14916fb23f85707f.js
[+] Writing 1014 bytes to dhbsrc/webpack:/webpack/bootstrap 356c14916fb23f85707f
[+] Writing 3174 bytes to dhbsrc/webpack:/app/scripts/client.js
[+] Writing 281 bytes to dhbsrc/webpack:/~/babel-runtime/helpers/interop-require-default.js
[+] Writing 151 bytes to dhbsrc/webpack:/~/babel-core/polyfill.js
{snip}
[+] Writing 271 bytes to dhbsrc/webpack:/~/rc-tooltip/~core-js/library/fn/object/set-prototype-of.js
[+] Writing 315 bytes to dhbsrc/webpack:/~/rc-tooltip/~core-js/library/modules/es6.object.set-prototype-of.js
[+] Writing 1044 bytes to dhbsrc/webpack:/~/rc-tooltip/~core-js/library/modules/_set-proto.js
[+] Writing 308 bytes to dhbsrc/webpack:/~/rc-tooltip/~core-js/library/fn/object/create.js
[+] Writing 307 bytes to dhbsrc/webpack:/~/rc-tooltip/~core-js/library/modules/es6.object.create.js
[+] Writing 360 bytes to dhbsrc/webpack:/~/rc-animate/~core-js/library/fn/object/define-property.js
[+] Writing 371 bytes to dhbsrc/webpack:/~/rc-animate/~core-js/library/modules/es6.object.define-property.js
[+] Writing 1041 bytes to dhbsrc/webpack:/~/rc-animate/~babel-runtime/helpers/createClass.js
[+] done
doi@asov:~$ cd dhbsrc/
doi@asov:~/dhbsrc$ du -hs .
20M .
doi@asov:~/dhbsrc$ cd webpack\:/
~/ app/ js/ webpack/ (webpack)/
doi@asov:~/dhbsrc$ cd webpack\:/app/scripts/
actions/ components/ middlewares/ reducers/ selectors/ stores/ vendor/
doi@asov:~/dhbsrc$ cd webpack\:/app/scripts/components/
```

```
$ ./sourcemap -output test -jsurl http://localhost:8080/main.js
2024/01/05 18:43:53 [+] Retrieving JavaScript from URL: http://localhost:8080/main.js.
2024/01/05 18:43:53 [.] Found SourceMap in JavaScript body: data:application/json;charset=utf-
2024/01/05 18:43:53 [+] Retrieving Sourcemap from data:application/json;charset=utf-8;base64,e
2024/01/05 18:43:53 [+] Read 4708918 bytes, parsing JSON.
2024/01/05 18:43:54 [+] Retrieved Sourcemap with version 3, containing 535 entries.
2024/01/05 18:43:54 [+] Writing 4262 bytes to test/webpack:/app/node_modules/ansi-html-communi
2024/01/05 18:43:54 [+] Writing 40 bytes to test/webpack:/app/node_modules/axios/index.js.
```

JS Endpoints

Extraction

We simply know that it's not always constant ways to write the front-end JS Files:

- Different Developers / Mindsets
- Different Frameworks
- Different API Structure

So, **LITERALLY ALL** the tools may miss some endpoints

That's why I always depend on custom regex depending on the target I am testing.

So what is the best ways I use to extract endpoints?

```
saveOrganization: async a => _({
  method: "PUT",
  endpoint: "/endpoints/organization/update",
  body: a
}).then(r => n(r)),
```

```
removeBillingPlan:function(n){
  var r=n.billing_plan_name;
  return e.post(''.concat(t,"/billing_plans/remove"),{
    billing_plan_name:r,change_type:"full_remove"
  },
  Object(Mt.a)("2021-11"))
},
getInvoices:function(){
  var n=arguments.length>0&&void 0!==arguments[0]?arguments[0]:1;
  return e.get(''.concat(t,"/invoices?limit=").concat(n),Object(Mt.a)(
    "2021-11"))
},
getAddOnBillingPlanAvailability:function(){
  return e.get(''.concat(t,
    "/store_billing_plan_availability?include=billing_plan_pricing_templates"),Object(Mt.a)("2021-11"))
},
```

```
var u=L(73994);|
class $e{
  constructor(Ie){
    this.suppressContextMenu=!0,this.loadingCellRenderer=j.cvY,this.getUrl
    =( )=>`/light/reports/advertisers/${Ie}/schedule`,this.
    defaultPagination=!1,this.colDef=[{
      headerName:"Report Name",sortable:!1,lockPosition:!0,field:"name",
      cellClass:["name-cell"],flex:1,resizable:!1
    }],
  },
}
```

Ways to extract?

- ## Ways to extract?

```

>/returns
>/search
>/shop
>/submitfeedback
>/ums
>/ums/user/advancedConfiguration
>/ums/user/customerAddress
>/ums/user/preferences
>/ums/user/securitySettings
>/ums/user/shippingSettings
>/update-password
>/user
>/user/advancedConfiguration
>/user/customerAddress
>/user/preferences
>/user/securitySettings
>/user/shippingSettings
>/user/updateAppVersion
>/v1/driver/
>/v1/filter
>/v1/filters/returns
>/v1/parcel/returns
>/v1/parcelcode
>/v1/parcels
>/v1/reference/pickupcountries
>/v1/returns/shipment
>/v1/returns/shipment/FIRED=S
>/v1/returns/WILL_BE_FIRED=cu
>/v1/returns/csv
>/v1/returns/csv?key=
>/v1/returns/dashboard
>/v1/returns/dashboard/pot
>/v1/returns/logs/
>/v1/routes/
>/v1/user/
>/validate-invite

```

JS Link Finder

Burp Extension for a passively scanning JavaScript files for endpoint links. - Export results the text file - Exclude specific 'js' files e.g. jquery, google-analytics

Estimated system impact

Overall: **Low**

Memory	CPU	Time	Scanner
Low	Low	Low	Low

Author: InitRoot
Version: 1.0.0
Source: <https://github.com/portswigger/js-link-finder>
Updated: 05 Sep 2019

Rating: Submit rating

Popularity:

Reinstall

A screenshot of the GitHub repository page for 'endext'. The repository name 'endext' is prominently displayed in the top left. In the top right corner, there is a 'Public' badge. Below the repository name, a description states: 'EndExt is a .go tool for extracting all the possible endpoints from the JS files'. Underneath the description, the repository statistics are shown: 'Go' (indicated by a blue circle icon), '188' stars (indicated by a star icon), and '20' forks (indicated by a fork icon). On the right side of the page, a code snippet is displayed within a light blue box, showing a terminal command and its output. The command is 'SirBugs@MacBook-Pro endext % cat regex.tmp'. The output consists of several lines of Go code defining regular expressions for path and to fields in JSON, and a function call to fetch the data from a specific URL.

Request Crafting

Manual Crafting

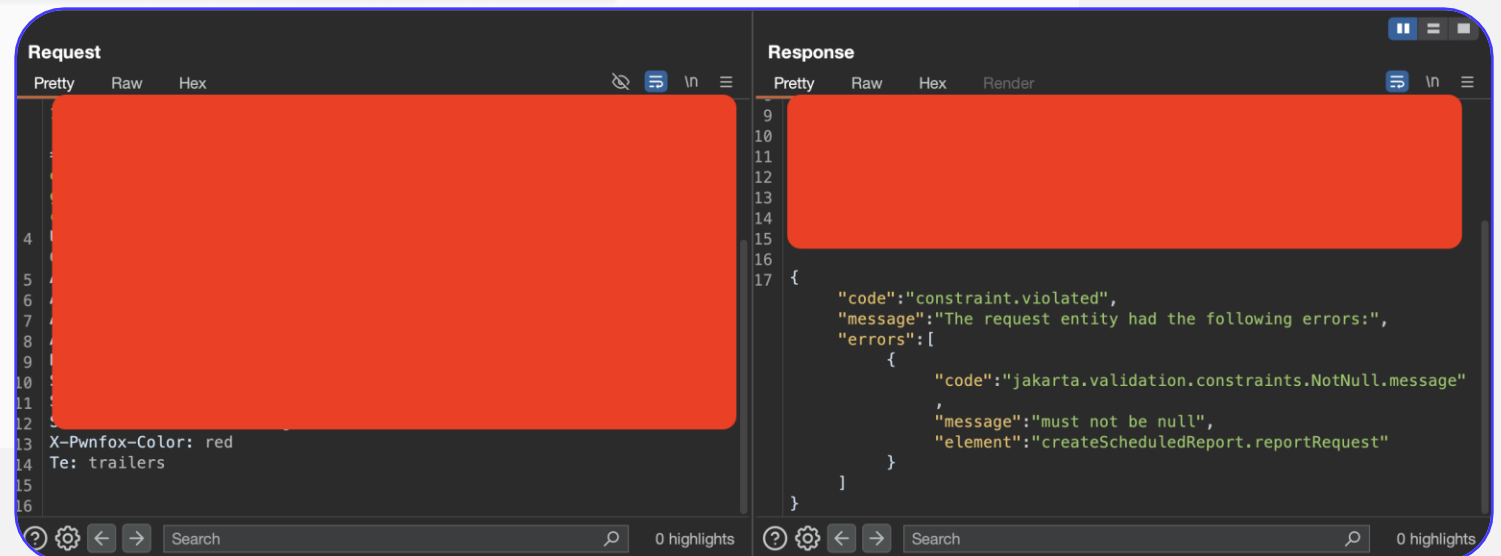
I cannot use a function to create report, because i don't have an active campaign (**I need 500\$ man**)

The endpoint is into the JS Files, that's my time!

```
[ 19 ] p_modules_secured_secured_module_ts.6d34891445542d62.js : `light/reports/advertisers/${Ie}/schedule`
[ 20 ] p_modules_secured_secured_module_ts.6d34891445542d62.js : `${Ee}d`
[ 21 ] p_modules_secured_secured_module_ts.6d34891445542d62.js : `${this.LIGHT_URL}/agency/${Ue}`
[ 22 ] p_modules_secured_secured_module_ts.6d34891445542d62.js : `${Ue}`
[ 23 ] p_modules_secured_secured_module_ts.6d34891445542d62.js : `${this.LIGHT_URL}/accounts/upgrades`
[ 24 ] p_modules_secured_secured_module_ts.6d34891445542d62.js : `${this._appConfigService.apiUrl}`
[ 25 ] p_modules_secured_secured_module_ts.6d34891445542d62.js : `${this.API_URL}/users-management`
[ 26 ] p_modules_secured_secured_module_ts.6d34891445542d62.js : `${this.USERS_MANAGEMENT_URL}/${Ue}`
[ 27 ] p_modules_secured_secured_module_ts.6d34891445542d62.js : `${this.USERS_MANAGEMENT_URL}/change-access`
```

On trying to call it:

So you need anything right? **Not null !**



Request Crafting

Manual Crafting

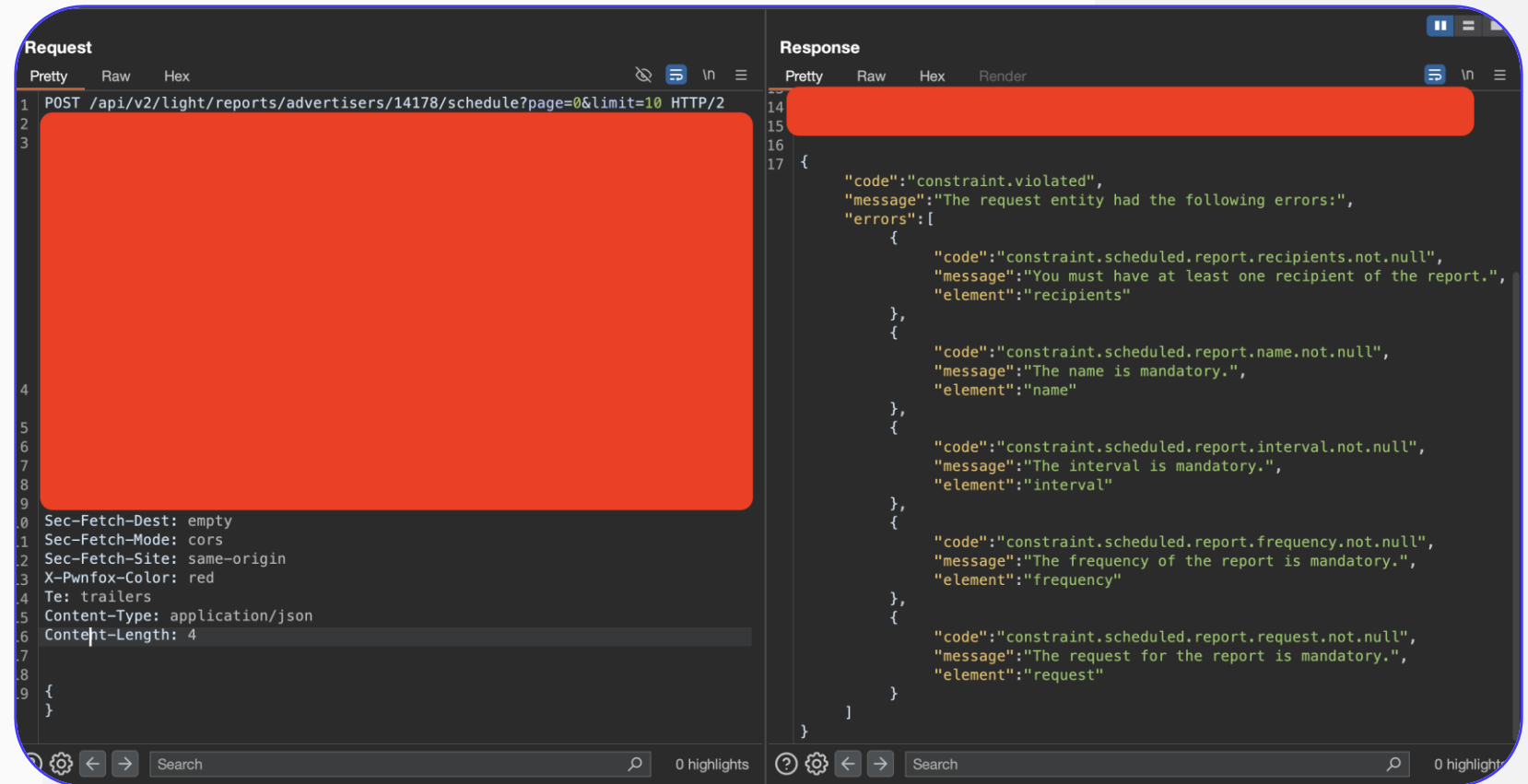
OK

BRO

CHILLLLLL

I started submitting
value after value ...

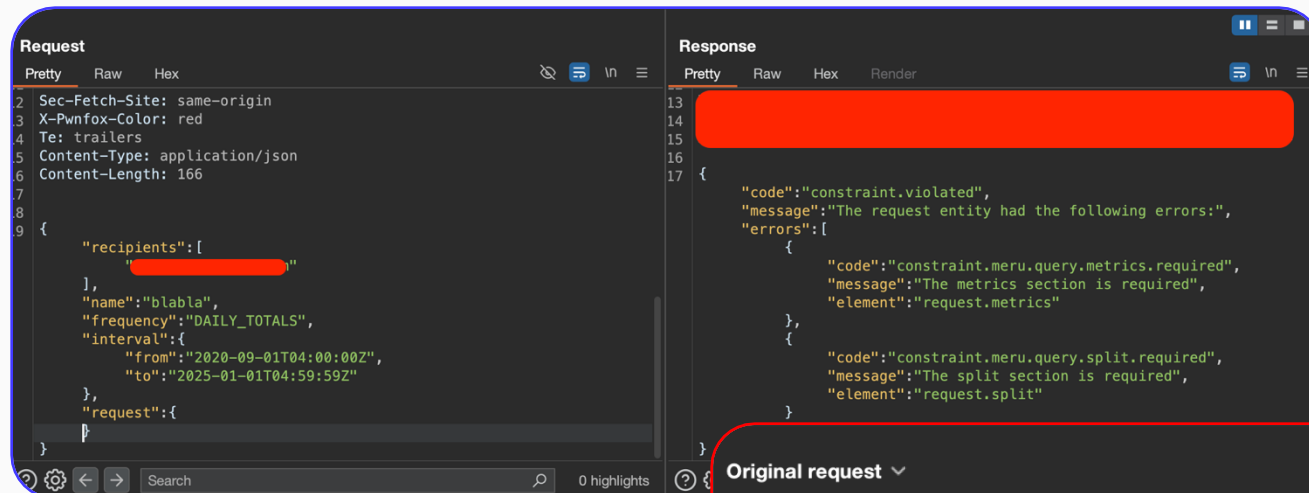
BUT



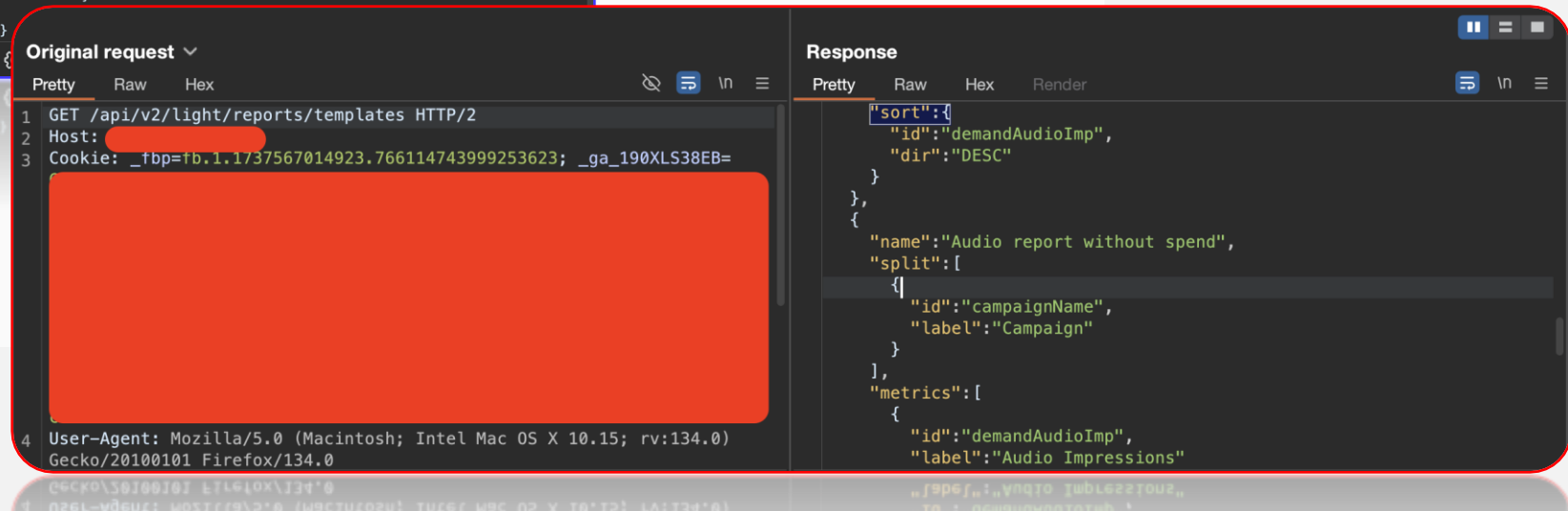
Request Crafting

History Crafting

Sometimes, our burp history can help us completing building the full request too! The request asked for:



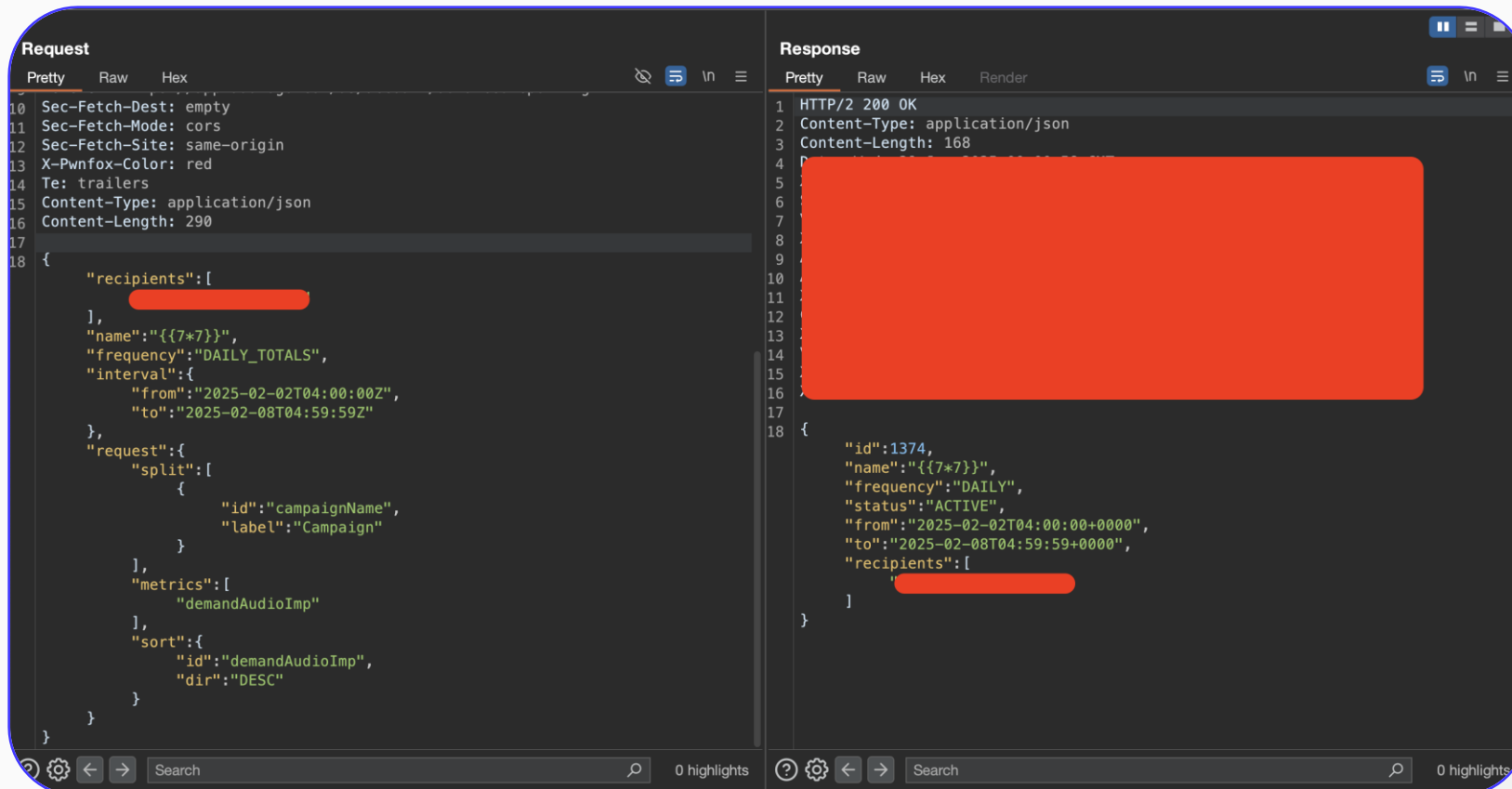
The remaining part of the request I grepped it from the history! After finding this request & response:



Request Crafting

Final Manual & History Crafting

Sometimes, our burp history can help us completing building the full request too! The request asked for:



FINALLY !

AND AS WE KNOW ...
BECAUSE THE HARD WORK
ALWAYSSSS PAYSSS OFF

Request Crafting

Final Manual & History Crafting

I found no bugs in this endpoints after
all of that ...



Request Crafting

App Flow Crafting

In some cases too, The app flow helps us in identifying what exactly parameters are needed for requests.

In a previous hunting, a request reflected the following response:

```
HTTP/2 500 Internal Server Error]
```

```
{"message":"Error: This user does not have admin  
access!","type":"error"}
```

So what's needed? From the app workflow, most of the requests had one of the following params:

- **Session_id**
- **Access_token**
- **User_token**
- **Token**
- **Session**
- **Access_id**

So knowing this + seeing the err msg, let me know that it needs the token/identifier that it uses in most of the requests.

Request Crafting

AI Crafting

Maybe AI Is helpful in crafting requests If you don't understand what's in response!

I've sent the following POST data:

```
{ "recipients":  
  [ "bag0za@outlook.com", "name": "blabla", "frequency": "DAILY_T  
    OTALS", "interval": { "from": "2020-09-01T04:00:00Z", "to": "2025-  
      01-01T04:59:59Z", "request": { "split":  
        [ { "id": "campaignName", "label": "Campaign" } ], "metrics": {} } }
```

received the following response:

```
{ "code": "invalid.json", "message": "Cannot deserialize value of type  
  java.util.ArrayList<java.lang.String> from Object value (token  
  JsonToken.START_OBJECT)" }
```

what does this mean?

"Cannot deserialize value of type
(token JsonToken.START_OBJECT)"

1. `java.util.ArrayList<java.lang.String>`: This refers to an array of strings, so the server is expecting a list of strings.
2. `Object value (token JsonToken.START_OBJECT)`: This suggests that the system is receiving an object `{ }` where it expected a list of strings `[ ]`.

Now, looking at your POST data, the issue seems to be related to the `"metrics"` field. In your request, `"metrics"` is an empty object `{ }`. However, if the server is expecting an array of strings, it may be the cause of this error.

Possible Fixes:

1. If the `"metrics"` field should contain an array of strings, you like:

```
json  
  
"metrics": [ "metric1", "metric2" ]
```

```
"bag0za@outlook.com"  
},  
"name": "blabla",  
"frequency": "DAILY_TOTALS",  
"interval": {  
  "from": "2020-09-01T04:00:00Z",  
  "to": "2025-01-01T04:59:59Z"  
},  
"request": {  
  "split": [  
    {  
      "id": "campaignName",  
      "label": "Campaign"  
    }  
  ],  
  "metrics": {  
    }  
  }  
}
```

```
16 {  
17   "code": "invalid.json",  
    "message":  
      "Cannot deserialize value of type `java.util.ArrayList<java.lang.String>` fr  
om Object value (token `JsonToken.START_OBJECT`)"  
}
```

Q: Can you send me edited POST Data?

```
"request": {  
  "split": [  
    {  
      "id": "campaignName",  
      "label": "Campaign"  
    }  
  ],  
  "metrics": [  
    "demandAudioImp"  
  ],  
  "sort": {  
    }  
  }  
}
```

```
18 {  
  "id": 1374,  
  "name": "{*7}",  
  "frequency": "DAILY",  
  "status": "ACTIVE",  
  "from": "2025-02-02T04:00:00+0000",  
  "to": "2025-02-08T04:59:59+0000",  
  "recipients": [  
    "  
  ]  
}
```

Monitoring JS Files

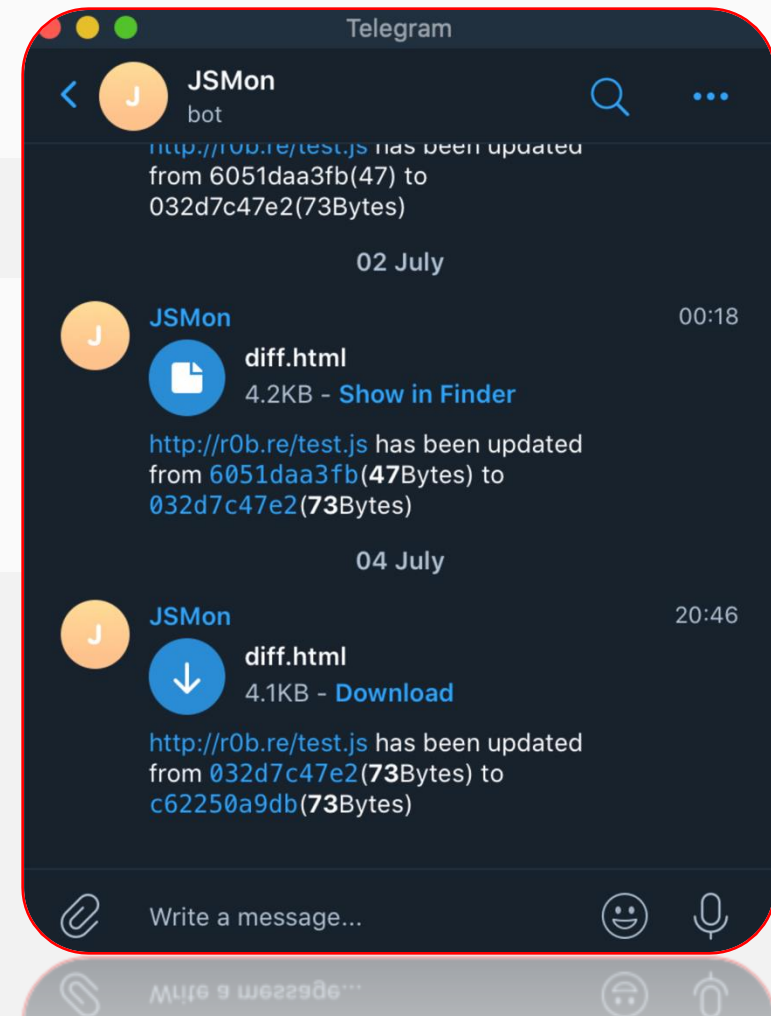
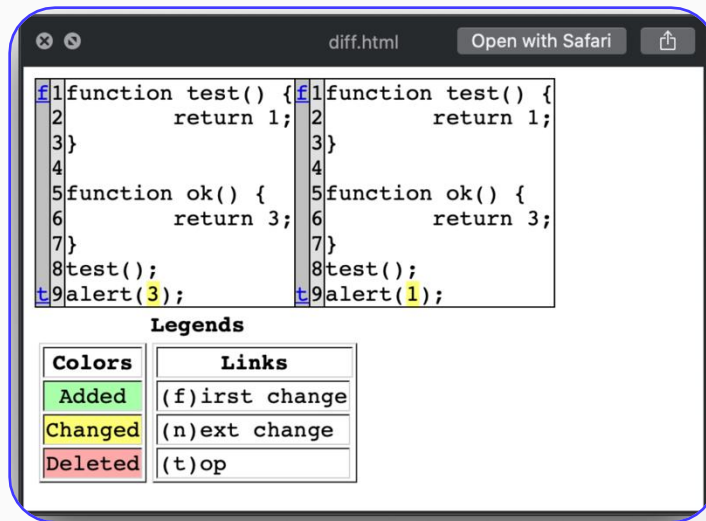
Jsmo

If you are a JS Files Addict, you can monitor them for any changes.

If they don't have hash for sure ..

If they have hashes, check if the file still there or not.

If the file is not there, so there's something changed.



Vulns Disclosure

SSRF (Leaked Internal Host & Error Msg)

In an old case, I had an SSRF .. But I could not read internal files!
That's why I reported it Blind at first :(

Bug After Inspecting the JS Files? A friend of me fetched this host!
Which by calling this host later, disclosed internal API routes, disclosing API Keys Info.

```
72450);t.randomFill=p.randomFill,t.randomFillSync=p.randomFillSync,t.createCred  
Error(["sorry, createCredentials is not implemented yet","we accept pull  
requests","http://uas-internal.uas.svc.cluster.local/github/requests"].join("\n"  
E_PRIME:2,DH_CHECK_P_NOT_PRIME:1,DH_UNABLE_TO_CHECK_GENERATOR:4,DH_NOT_SUITABLE_
```



#237

Blind SSRF in

March 14, 2024

`V3\_CreateWebhook` via callbackUrl`
parameter

To: [redacted] • Critical

Vulns Disclosure

SSRF (Leaked Internal Host & Error Msg)

Disclosing any api key information (Since it's an internal call)

Thanks to JS Files!

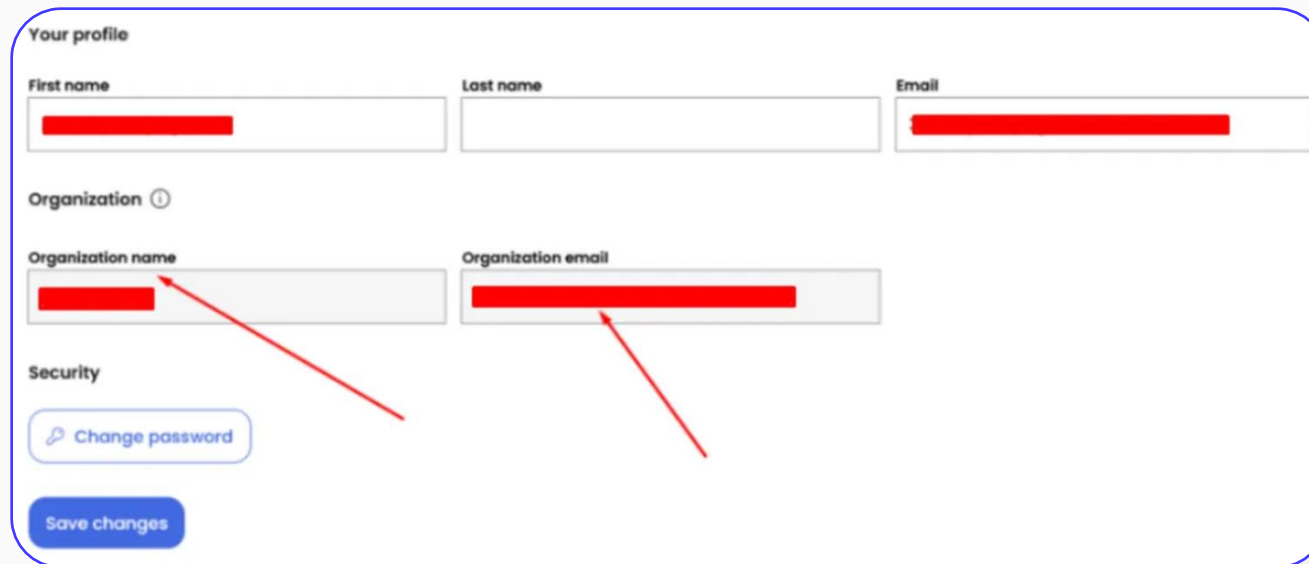
```
"operationName": "V3_CreateWebhook",
"variables": {
  "appId": "087fcce7-059a-455a-ba21-8a781fc414b8",
  "webhook": {
    "triggerTypes": [
      "calendar.created",
      "calendar.updated",
      "calendar.deleted",
      "event.created",
      "event.updated",
      "event.deleted",
      "grant.created",
      "grant.updated",
      "grant.deleted",
      "grant.expired",
      "message.created",
      "message.updated",
      "message.send_success",
      "message.send_failed",
      "message.bounce_detected",
      "message.opened",
      "message.link_clicked",
      "thread.replied"
    ],
    "callbackUrl": "http://uas-internal.uas.svc.cluster.local/v3/admin/applications/19edc893-f800-4c50-89db-a0a3caf2a038/apikeys/",
    "description": "aaa",
    "notificationEmails": [
      "circle0x01@gmail.com"
    ]
  }
},
"query": "mutation V3_CreateWebhook($appId: String!, $webhook: WebhookInput!) {\n  createWebhook(appId: $appId, webhook: $webhook) {\n    webhookSecret\n    __typename\n  }\n}"

22 X-Download-Options: noopen
23 X-Content-Type-Options: nosniff
24 Origin-Agent-Cluster: 71
25 X-Permitted-Cross-Domain-Policies: none
26 Referrer-Policy: no-referrer
27 Cache-Control: no-store
28 Etag: W/"lba-90ksh6r4ZzrMf4ADcyXD16sysH0"
29 Via: 1.1 google
30 Cf-Cache-Status: DYNAMIC
31 Server: cloudflare
32 Cf-Ray: 85575bf9fbad0c50-MRS
33
34 {
  "errors": [
    {
      "message": "unable.verify.webhook_url : expected 101256ee-26b7-4a76-8ff6-38bb0bfb7ca, got (\\"data\\" : [{"id\\" : \\"d7bd653e-2be2-4bf4-8b21-e3f2476fcafc\\", \\"name\\" : \\"asdasd\\", \\"permissions\\" : [{"all\\"}], \\"status\\" : \\"active\\", \\"expires_at\\" : 1865787441, \\"created_at\\" : 1707934641, \\"updated_at\\" : 1707934641}]), \\"locations\\" : [{"line": 2, "column": 3}]",
      "path": [
        "createWebhook"
      ],
      "extensions": {
        "code": "BAD_REQUEST"
      }
    }
  ],
  "data": {
    "createWebhook": null
  }
}
35
```

Vulns Disclosure

Editing Only Admin Editable Values

Normally we cannot edit the following inputs of the org, we need to contact the admin



The screenshot shows a 'Your profile' form. It contains input fields for 'First name', 'Last name', and 'Email', all of which are redacted with black bars. Below these are 'Organization' fields, including 'Organization name' and 'Organization email', which are also redacted. Two red arrows point from the 'Organization name' and 'Organization email' fields towards the right side of the slide. At the bottom of the form, there is a 'Security' section with a 'Change password' button and a 'Save changes' button.

But finding this endpoint in JS files changed it!

```
saveOrganization: async a => _({  
  method: "PUT",  
  endpoint: "/endpoints/organization/update",  
  body: a  
}).then(r => n(r)),
```

Is this endpoint related to editing these fields? **/organization/update** ?? Maybe! Why not?

Vulns Disclosure

Editing Only Admin Editable Values

But because I know the application, I know it need this “session_id” in the POST data **(it needs it for most of the requests)**

```
PUT /endpoints/organization/update HTTP/2
Host: home.target.com
Cookie: XXXXXXXXXXXXXXXXXXXXXXXXXXXX
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101
Accept: application/json
Referer: https://home.target.com/
Content-Type: application/json
Content-Length: 76
Origin: https://home.target.com/

{"hello":"goodbye"}
```

HTTP/2 500 Internal Server Error]

```
{"message":"Error: This user does not have admin
access!","type":"error"}
```

```
PUT /endpoints/organization/update HTTP/2
Host: home.target.com
Cookie: XXXXXXXXXXXXXXXXXXXXXXXXXXXX
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 F
Accept: application/json
Referer: https://home.target.com/
Content-Type: application/json
Content-Length: 76
Origin: https://home.target.com/

{"session_id":"9n5n6xxxxxxxxxc05llye4ul", "name":"changevalueeeee"}
```

HTTP/2 200 OK

```
{"billing_enabled":false,"billing_type":"trial","created_at":16769974
71,"data_streams_plan":"starter","email_address":"xxxxxxxxxxx@xxxxxxx
x.xxx","feature_flag":
{"allow_usage_billing":false,"forever_free_tier":false},"first_non_auth_api_call_made":null,"id":"2lxxxxxxxxxxzf49rtjd8ia","invoices":null
,"is_marketplace_org":false,"marketplace_id":null,"name":"changevalue
eeee","stripe_customer_id":null,"trial_end_date":1678207071}
```

Vulns Disclosure

Plan Limitations Bypasses

Most of the applications that have plans, have a similar page.

But since we don't have these plans (**not always we can buy too**), we can't even click the buttons or access these functions at all.

Here the JS files can lead us to these endpoints directly, let's make a POST / GET and see if we can use these paid functions endpoints in the free plan.

Compare plans				
 EXECUTION	Free	Basic	Professional	Organization
Email	1	1	5	15
Daily Email Sending Limit	250	Unlimited	Unlimited	Unlimited
AI Email Composer	5,000 / Month	250,000 / Month	800,000 / Month	1,000,000 / Month
Chrome Extension ▾	✓	✓	✓	✓
Meetings ▾	✓	✓	✓	✓
Conversations ▾	✓	✓	✓	✓
Deals	✓	✓	✓	✓
Sequences	2	2	✓	✓
Advanced Sequences ▾			✓	✓
Automated Workflows	2	2	500	500
US Dialer			✓	✓
International Dialer				✓

Contacts



Fares Walid

Security Consultant @BuguardLLC

✉ bag0zathev2@gmail.com

Thank you

```
def key_pressed(key):  
    print(f"{key} pressed")
```

```
# Create the main window  
root = tk.Tk()  
root.title("Simple Keyboard")
```

```
# Create buttons for keys  
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']  
in the following keys:  
    button = tk.Button(root, text=key, width=5,  
        command=lambda k=key: key_pressed(k))  
    button.pack(side=tk.LEFT)
```

