

**JavaScript Static Analysis
(Arabic)**

بسم الله الرحمن الرحيم

السلام عليكم ورحمة الله وبركاته

الكاتب:

عبدالرحمن عبدالله

تويتر:

[@Infosec 90](#)

الفئة المستهدفة :

مطور المواقع

مختبر اختراق

صائد مكافآت

مستوى الصعوبة

من متوسط الى متقدم

(سيتم ذكر سيناريو متوسط ومتقدم بحيث يمكن اكتشاف الثغرات عن طريق محرر نصوص او بشكل يدوي)

متطلبات الشرح:

PhpStrom تثبيت محرر نصوص برمجي بالشرح تم استخدام

(يمكن للطلاب او الخريجين اخذ نسخة مجانية باستخدام ايميل الجامعة)

نظام لينكس او يمكن استخدام النسخة المصغرة على ويندوز

يفضل المعرفة مسبقا في قراءة اكواد الجافا سكريبت

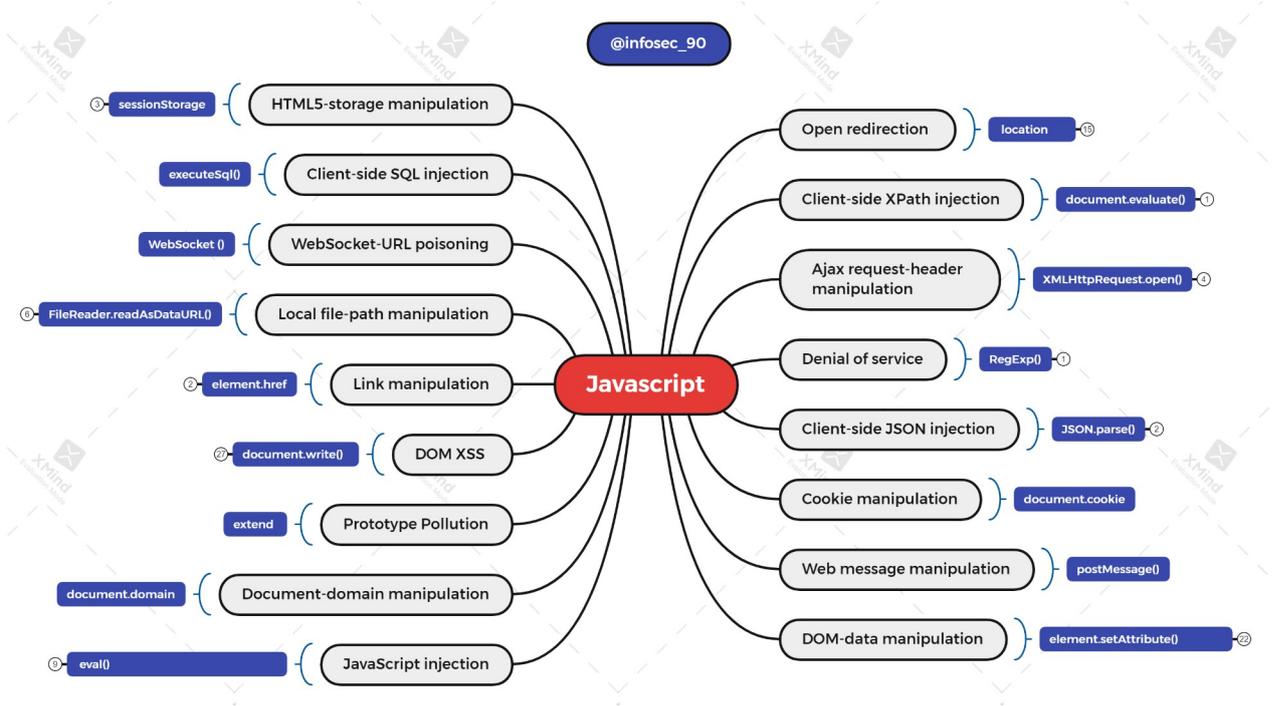
يفضل معرفة في التعامل مع التيرمنال

متوقع مخرجات المقالة:

جمع ملفات الجافا سكريبت من الهدف

اكتشاف عدة ثغرات في ملفات الجافا سكريبت

الحصول على اول CVE لك



<https://github.com/Ph33rr/attack/tree/main/Javascript>

عندما نتحدث عن سيناريو فحص blackbox عملية recon هي الاساس اذا كانت اعرق تعني نتائج افضل. والنتيجة ثغرات اكثر وسرعة في الاكتشاف في هذه المقالة لجمع ملفات الجافا سكريبت نستخدم recon ثم عمل تحليل للملفات وعمل اختبار اختراق بعدة طرق حيث تحدثت سابقا في مقالة عن webstorage بناء على تجربة المستخدم وايضا تحدثت عن ماهي ملفات الجافا سكريبت ووظيفتها واهميتها بحيث اذا كنت لاتعلم ما هي يمكنك الرجوع للمقدمة الموجودة في المقالة والتي تشرح ماهي ملفات الجافا سكريبت بتفصيل اكثر حيث لن يتم اعادة تعريفها هنا وايضا تجدون لاب لتجربة الثغرة وفهمها مع شرح امثلة حقيقية منها رواتر هواوي حيث كانت الإصابة بعدة إصدارات وتم إصلاحها

<https://github.com/Ph33rr/webstorage>

قبل البدء في عملية التحليل لملفات الجافا سكربت علينا ان نتعلم بعض الأساسيات
نسرده لكم قصة قصيرة في احدى الأيام قرر شخص السفر لحدى الدول كسائح
لتمضية بعض الوقت كفترة نقاهة في يوم الثلاثاء استلم صديقنا ايميل
من قبل احدى مواقع الحجوزات في الفنادق وما يتعلق بالسفر يفيد الايميل
بوجود خصم 50% على جميع الفنادق في حال تم الحجز خلال 24 ساعة
فقرر صديقنا الحجز عن طريق هذا الموقع لاستغلال الخصم والاستفادة منه
ولكن حدثت مشكلة اثناء الحجز

بدون بطاقة؟

كيف ترغب بالدفع؟

رمز CVC



يمكنك أن تجد رمز CVC في الجزء الخلفي من بطاقة الائتمان الخاصة بك. إنها الأرقام الثلاثة الأخيرة المطبوعة على شريط التوقيع.

*** اسم حامل البطاقة**
Abdullah Abdulrahman

*** نوع البطاقة**
MasterCard

*** رقم بطاقة الائتمان / الخصم**
[Redacted]

*** تاريخ انتهاء الصلاحية**
يرجى إدخال تاريخ انتهاء صلاحية صالح خلال تواريخ حرك
2018 / 01

*** رمز CVC**
454

أضف بطاقة الائتمان هذه لحسابك للحصول على حجز أسرع

بطاقة الائتمان انتهت صلاحياتها حاول صديقنا الاتصال بالبنك

لمحاولة الاستفادة من العرض كان رد البنك سيتم ارسال بطاقة جديدة بعد 15 يوم عمل
عندها سيكون العرض منتهي فكر وفكر صديقنا المبرمج هل يوجد حل خصوصا ان الموقع
لايحتاج الى خصم المبلغ بالوقت الحالي فقط يحتاج الى بطاقة انتمان لسوء حظ هذا الموقع
ان صديقنا لديه خلفية في البرمجة صديقنا يرغب في هذا العرض بشدة
وبعد بحث مطول بطريقة مختصرة في الموقع باستخدام أدوات المطور في المتصفح
توصل صديقنا للتالي:



"East Side Studio عن طريق Booking.com" <[REDACTED]@booking.com>



[REDACTED]
om>

يتحمل مزود مكان الإقامة مسؤولية محتوى هذه الرسالة بالكامل. مرسله عبر Booking.com.

Dear ABDULRAHMAN,

Thanks for booking East Side Studio. We are looking forward to welcoming you in [REDACTED]!

Once you have arrived at the airport or are 30 minutes away from [REDACTED], can you please give me a call at this number

+965 21716951. If you want to check in always call 30 minutes ahead

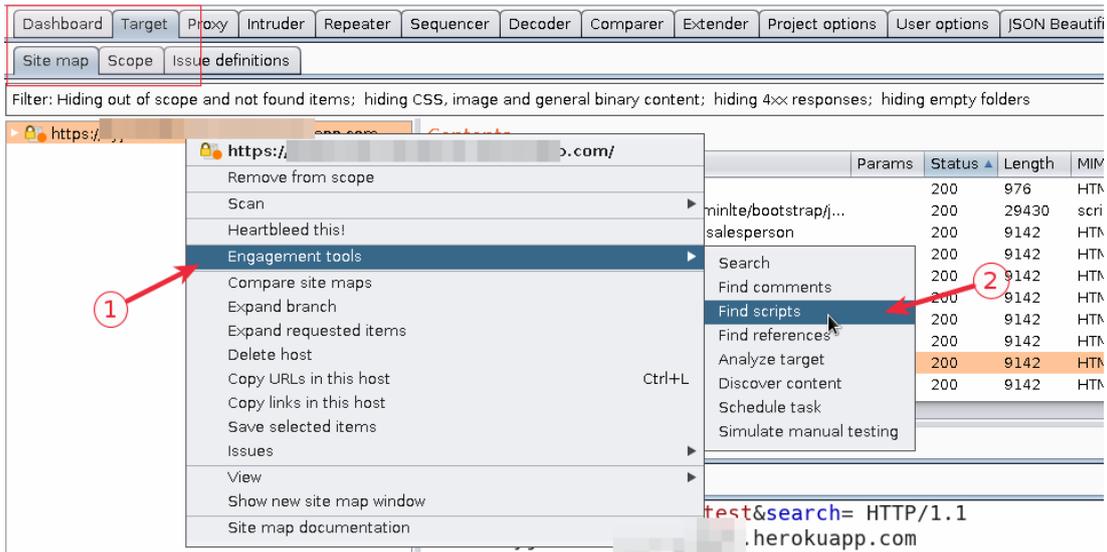
وبعدها عاش صديقنا في سعادة غامرة شكرا جافا سكرت

لنذهب الى الى الجزء العملي ولا يتضمن قصص خيالية

جمع ملفات الجافا سكريبت

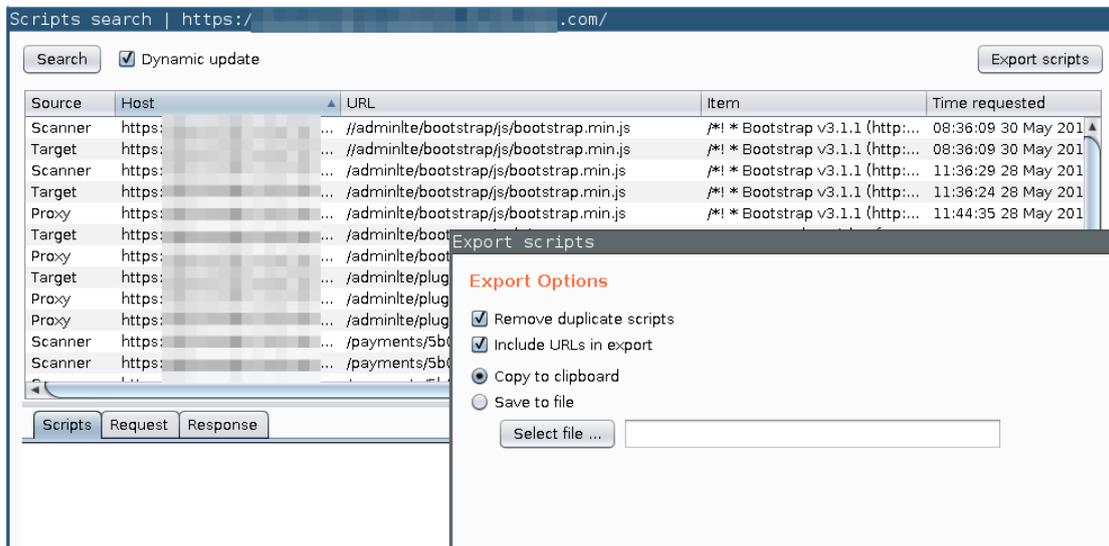
توجد عدة طرق في جمع ملفات الجافا سكريبت من موقع محدد او مجموعة مواقع ولكل شخص
Methodology خاصة به نذكر منها التالي :

استخدام burpsuite لحفظ ملفات الجافا سكريبت بجهازك



The screenshot shows the Burp Suite interface with the 'Find scripts' option highlighted in the context menu. A red circle '1' points to the 'Engagement tools' menu, and a red circle '2' points to the 'Find scripts' option. The background shows a table of scan results with columns for Params, Status, Length, and MIME type.

Params	Status	Length	MIME
	200	976	HTM
adminlte/bootstrap/j...	200	29430	scri
salesperson	200	9142	HTM
	200	9142	HTM



The screenshot shows the 'Scripts search' window in Burp Suite. It displays a table of search results with columns for Source, Host, URL, Item, and Time requested. Below the table, there are 'Export scripts' and 'Export Options' sections.

Source	Host	URL	Item	Time requested
Scanner	https://	... //adminlte/bootstrap/js/bootstrap.min.js	/*! * Bootstrap v3.1.1 (http...	08:36:09 30 May 201
Target	https://	... //adminlte/bootstrap/js/bootstrap.min.js	/*! * Bootstrap v3.1.1 (http...	08:36:09 30 May 201
Scanner	https://	... /adminlte/bootstrap/js/bootstrap.min.js	/*! * Bootstrap v3.1.1 (http...	11:36:29 28 May 201
Target	https://	... /adminlte/bootstrap/js/bootstrap.min.js	/*! * Bootstrap v3.1.1 (http...	11:36:24 28 May 201
Proxy	https://	... /adminlte/bootstrap/js/bootstrap.min.js	/*! * Bootstrap v3.1.1 (http...	11:44:35 28 May 201
Target	https://	... /adminlte/bo...		
Proxy	https://	... /adminlte/bo...		
Target	https://	... /adminlte/plug...		
Proxy	https://	... /adminlte/plug...		
Proxy	https://	... /adminlte/plug...		
Scanner	https://	... /payments/5b...		
Scanner	https://	... /payments/5b...		

Export Options

- Remove duplicate scripts
- Include URLs in export
- Copy to clipboard
- Save to file

Select file ...

هذه فقط متواجدة في النسخة المدفوعة لكن للنسخة المجانية نتبع الطريقة التالية لنسخ الروابط

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options JSON Bea

Intercept HTTP history WebSockets history Options

Filter: Hiding out of scope items; hiding CSS, image and flash content; showing specific extensions

Filter by request type

- Show only in-scope items
- Hide items without responses
- Show only parameterized requests

Filter by MIME type

- HTML
- Script
- XML
- CSS
- Other text
- Images
- Flash
- Other binary

Filter by status code

- 2xx [success]
- 3xx [redirection]
- 4xx [request error]
- 5xx [server error]

Filter by search term

Filter by file extension

- Show only: js
- Hide: *.gif,*.jpg,*.png,*.css

Filter by annotation

- Show only commented items
- Show only highlighted items

Show all Hide all Revert changes

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options JSON Beautifier SHELING CSP Hea

Intercept HTTP history WebSockets history Options

Filter: Hiding out of scope items; hiding CSS, image and flash content; showing specific extensions

#	Host	Me...	URL	Status	Extension	MIME type	Length	Para...	Edited	Title
142	https://	...	/adminlte/bootstrap/js/dt.js	200	js	script	2954			
158	https://	...	/adminlte/bootstrap/js/bootstrap.min.js	200	js	script	29430			
159	https://	...	/adminlte/plugins/iCheck/checkbox.min.js	200	js	script	4835			
175	https://	...	/adminlte/bootstrap/js/dt.js	304	js	script	287			
176	https://	...	/adminlte/bootstrap/js/bootstrap.min.js							
177	https://	...	/adminlte/plugins/iCheck/checkbox.min.js							
185	https://	...	/adminlte/bootstrap/js/dt.js							
186	https://	...	/adminlte/bootstrap/js/bootstrap.min.js							
187	https://	...	/adminlte/plugins/iCheck/checkbox.min.js							
195	https://	...	/adminlte/bootstrap/js/dt.js							
196	https://	...	/adminlte/bootstrap/js/bootstrap.min.js							
197	https://	...	/adminlte/plugins/iCheck/checkbox.min.js							
205	https://	...	/adminlte/bootstrap/js/dt.js							
206	https://	...	/adminlte/bootstrap/js/bootstrap.min.js							

Request Response

Raw Params Headers Hex

GET /adminlte/bootstrap/js/bootstrap.min.js HTTP/1.1

Copy URLs Copy links

والآن نحفظ روابط ملفات الجافا سكريبت في ملف txt مثال:

targetJS.txt

بعد حفظ الروابط في ملف txt

نكتب Loop لتنزيل ملفات الجافا سكريبت المتواجدة روابطها في الملف

```
for i in $(cat targetJS.txt);do wget $i;done
```

```
https://github.githubassets.com/assets/chunk-profile-860a1228.js
https://github.githubassets.com/assets/chunk-readme-toc-element-5bf0869b.js
https://github.githubassets.com/assets/chunk-ref-selector-fd78f451.js
https://github.githubassets.com/assets/chunk-responsive-underlinenav-0ff33106.js
https://github.githubassets.com/assets/chunk-runner-groups-496cb7e9.js
https://github.githubassets.com/assets/chunk-series-table-49574ad0.js
https://github.githubassets.com/assets/chunk-severity-calculator-element-b64efa7a.js
https://github.githubassets.com/assets/chunk-sortable-behavior-8fb3dbd4.js
https://github.githubassets.com/assets/chunk-three.module-9ca6b751.js
https://github.githubassets.com/assets/chunk-toast-58af155f.js
https://github.githubassets.com/assets/chunk-tweetsodium-d6f499bf.js
https://github.githubassets.com/assets/chunk-unveil-7ef70e39.js
https://github.githubassets.com/assets/chunk-user-status-submit-50e14d5b.js
https://github.githubassets.com/assets/chunk-webgl-warp-70abbff9.js
```

```
(kali@kali)-[~/Desktop]
└─$ cat FileJs.txt|cut -d = -f 9|cut -d \" -f 2 > targetJS.txt
```

```
(kali@kali)-[~/Desktop]
└─$ for i in $(cat targetJS.txt);do wget $i;done
```

```
14.9 MB/s - 'chunk-unveil-7ef70e39.js' saved [682/682]
https://github.githubassets.com/assets/chunk-user-status-submit-50e14d5b.js
Resolving github.githubassets.com (github.githubassets.com)... 185.199.110.154, 185.199.111.154,
Connecting to github.githubassets.com (github.githubassets.com)|185.199.110.154|:443 ... connecte
HTTP request sent, awaiting response... 200 OK
Length: 8038 (7.8K) [application/javascript]
Saving to: 'chunk-user-status-submit-50e14d5b.js'
chunk-user-status-submit-50e14d5b.js 100%[=====]
792 KB/s - 'chunk-user-status-submit-50e14d5b.js' saved [8038/8038]
--
https://github.githubassets.com/assets/chunk-webgl-warp-70abbff9.js
Resolving github.githubassets.com (github.githubassets.com)... 185.199.110.154, 185.199.108.154,
Connecting to github.githubassets.com (github.githubassets.com)|185.199.110.154|:443 ... connecte
HTTP request sent, awaiting response... 200 OK
Length: 7409 (7.2K) [application/javascript]
Saving to: 'chunk-webgl-warp-70abbff9.js'
chunk-webgl-warp-70abbff9.js 100%[=====]
1.27 MB/s - 'chunk-webgl-warp-70abbff9.js' saved [7409/7409]
(kali@kali)-[~/Desktop]
└─$
```

الآن تم حفظ الملفات المراد فحصها في الجهاز

سيناريو اخر نستخدم اداعة JS_Scrapper

برمجة الاخ مشعل @Mesh3l_911

: لتنزيل الاداعة

https://github.com/Mesh3l911/JS_Scrapper

```
Twitter: Mesh3l_911
+-----+
crossorigin="anonymous" integrity="sha512-tk76eo-
/vq+v2WjDF96t4qfUkZLA==<script type="application/javascript" data-mod
/github_githubassets.com/assets/chunk-severity-calculator-elem
JS for https://medium.com/ : defer="defer" integrity="sha512-j7Pb1H-
jgBP0DD_3Ka06Yp7A+BB4VJubHRjwMxhw/0icqA==" type="applicati
https://cdn-client.medium.com/lite/static/js/manifest.9a8ac41b.js
https://cdn-client.medium.com/lite/static/js/9115.1a9358c4.js
https://cdn-client.medium.com/lite/static/js/main.444bb8d9.js
https://cdn-client.medium.com/lite/static/js/5573.159bf40f.chunk.js
https://cdn-client.medium.com/lite/static/js/instrumentation.79ae5839.chunk.js
https://cdn-client.medium.com/lite/static/js/reporting.6471519f.chunk.js
https://cdn-client.medium.com/lite/static/js/2499.851d7502.chunk.js
https://cdn-client.medium.com/lite/static/js/192.961ccdd2.chunk.js
https://cdn-client.medium.com/lite/static/js/1645.857c77e3.chunk.js
https://cdn-client.medium.com/lite/static/js/8940.b1644406.chunk.js
https://cdn-client.medium.com/lite/static/js/1801.e27003de.chunk.js
https://cdn-client.medium.com/lite/static/js/7883.1f12fa26.chunk.js
https://cdn-client.medium.com/lite/static/js/2018.cda2d533.chunk.js
https://cdn-client.medium.com/lite/static/js/2264.1681143f.chunk.js
https://cdn-client.medium.com/lite/static/js/1478.b1452ac2.chunk.js
https://cdn-client.medium.com/lite/static/js/3955.f6285986.chunk.js
https://cdn-client.medium.com/lite/static/js/1373.2fa5a199.chunk.js
https://cdn-client.medium.com/lite/static/js/Homepage.1d77d301.chunk.js
(kali@kali)-[~/Desktop/JS_Scrapper]
└─$ for i in $(cat targetJS.txt);do wget $i;done
```

الان نحفظ الروابط في ملف txt ثم ننفذ Loop مثل المره السابقة

لحفظ ملفات الجافا سكربت في الجهاز

طرق اخرى

استخدام اداة Linkfinder

<https://github.com/GerbenJavado/LinkFinder>

وطريقة اخرى تستخدم Archive

[https://web.archive.org/cdx/search/cdx?](https://web.archive.org/cdx/search/cdx?url=google.com&matchType=domain&fl=original&collapse=urlkey&output=text&filter=statuscode:200)

[url=google.com&matchType=domain&fl=original&collapse=urlkey&output=text&filter=statuscode:200](https://web.archive.org/cdx/search/cdx?url=google.com&matchType=domain&fl=original&collapse=urlkey&output=text&filter=statuscode:200)

وتنفيذ نفس Loop لتنزيل ملفات الجافا سكربت ولاتنسى استخدام الامر `grep` لفلتره النتائج JS

خارج الموضوع هدية

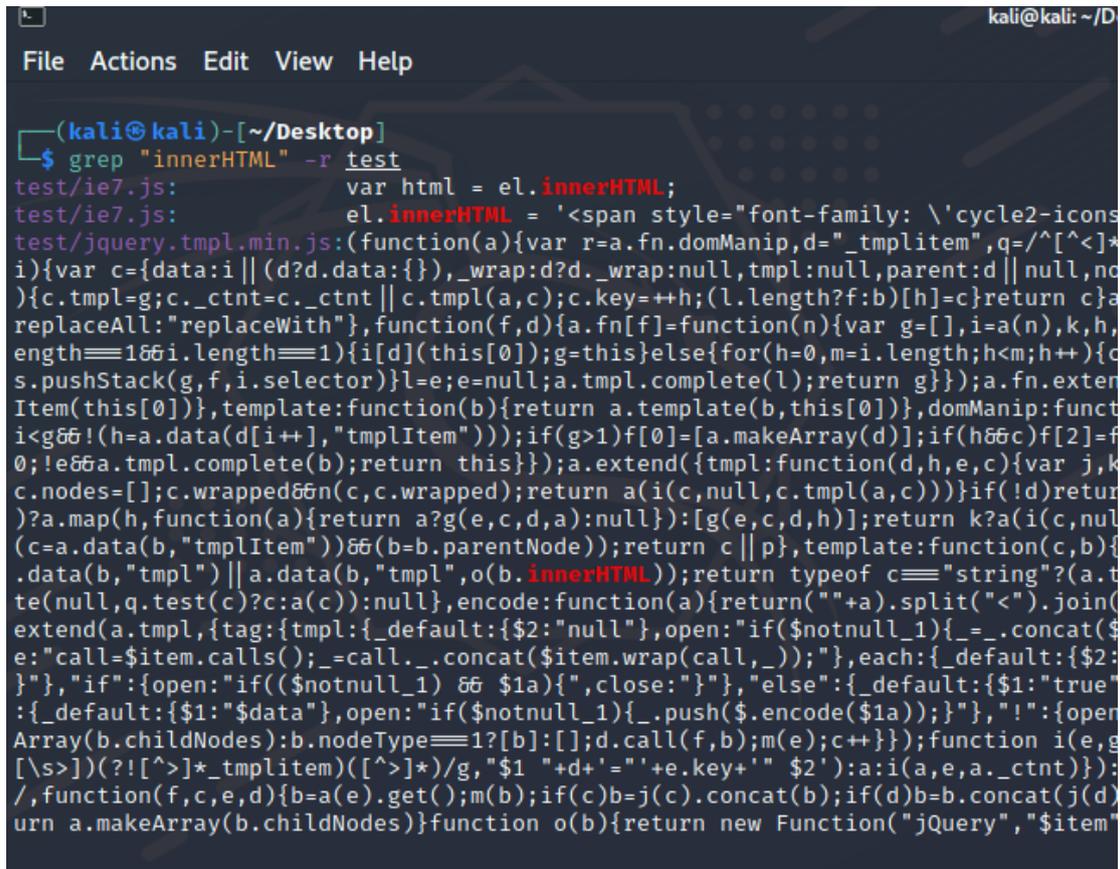
تغيير `statuscod` من الرقم 200 الى الرقم 403 ثم استخدام Loop على كل الروابط وعند وجود رابط الرد 200 يكون تخطيت 403 وحصلت على ثغرة بشكل سريع سيتم شرحها في تغريدة

توجد ايضا عدة طرق اخرى وادوات لم يتم ذكرها يجب عليك بناءا Methodology خاصة بك

في جمع ملفات الجافا سكربت

بعد الانتهاء من جمع الملفات تنتقل للخطوة التالية التحليل والبحث:

بعد ما تم حفظ الملفات في مجلد الان نستخدم التيرمنال للبحث عن بعض الدوائل الموجودة في الملفات السيناريو سيكون كالتالي البحث عن Domxss واكثر دالة مصابة هي innerHTML موضحة في الخريطة الذهنية ثم تعديلها او إعادة كتابتها بحيث يمكن تتبعها بسهولة اذا كانت تحتوي على regex نتخطاها او نتعلم كيفية عملها بطريقة اسرع واسهل نبدا في البحث بالطريقة التقليدية حيث تم حفظ الملفات في مجلد باسم test



```
kali@kali: ~/D
File Actions Edit View Help

(kali@kali)-[~/Desktop]
└─$ grep "innerHTML" -r test
test/ie7.js:         var html = el.innerHTML;
test/ie7.js:         el.innerHTML = '<span style="font-family: \'cycle2-icons
test/jquery.tpl.min.js:(function(a){var r=a.fn.domManip,d="_tmplitem",q=/^[^<]*
i){var c={data:i||(d?d.data:{}),_wrap:d?d._wrap:null,tmpl:null,parent:d||null,nc
){c.tmpl=g;c._cnt=c._cnt||c.tmpl(a,c);c.key=++h;(l.length?f:b)[h]=c}return c}a
replaceAll:"replaceWith"},function(f,d){a.fn[f]=function(n){var g=[],i=a(n),k,h,
ength===1&&i.length===1){i[d](this[0]);g=this}else{for(h=0,m=i.length;h<m;h++){c
s.pushStack(g,f,i.selector)}l=e;e=null;a.tmpl.complete(l);return g}};a.fn.exten
Item(this[0])},template:function(b){return a.template(b,this[0])},domManip:func
i<g&&(h=a.data(d[i++],"tplItem"));if(g>1)f[0]=[a.makeArray(d)];if(h&&c)f[2]=f
0;!e&&a.tmpl.complete(b);return this}};a.extend({tmpl:function(d,h,e,c){var j,k
c.nodes=[];c.wrapped&&n(c,c.wrapped);return a(i(c,null,c.tmpl(a,c)))}if(!d)retur
)?a.map(h,function(a){return a?g(e,c,d,a):null}):[g(e,c,d,h)];return k?a(i(c,nul
(c=a.data(b,"tplItem"))&&(b=b.parentNode));return c||p},template:function(c,b){
.data(b,"tpl")||a.data(b,"tpl",o(b.innerHTML));return typeof c=="string"?a.t
te(null,q.test(c)?c:a(c)):null},encode:function(a){return(""+a).split("<").join(
extend(a.tmpl,{tag:{tmpl:{_default:{$2:"null"},open:"if($notnull_1){_=_concat($
e:"call=$item.calls();_=_call._concat($item.wrap(call,_));"},each:{_default:{$2:
}}},"if":{"open:"if(($notnull_1) && $1a){","close:"},"else":{"_default:{$1:"true"
:{"_default:{$1:"$data"},open:"if($notnull_1){_._push($.encode($1a));"},"!":{"oper
Array(b.childNodes):b.nodeType===1?[b]:[];d.call(f,b);m(e);c++}});function i(e,g
[\\s>])(?![<^>]*_tmplitem)([<^>]*)/g,"$1 "+d+'="'+e.key+' "$2'):a:i(a,e,a._cnt)}):
/,function(f,c,e,d){b=a(e).get();m(b);if(c)b=j(c).concat(b);if(d)b=b.concat(j(d)
urn a.makeArray(b.childNodes)}function o(b){return new Function("jQuery","$item"
```

لم نجد شيء مهم او يمكن تتبعه بسهولة نبحث عن دالة أخرى document.location

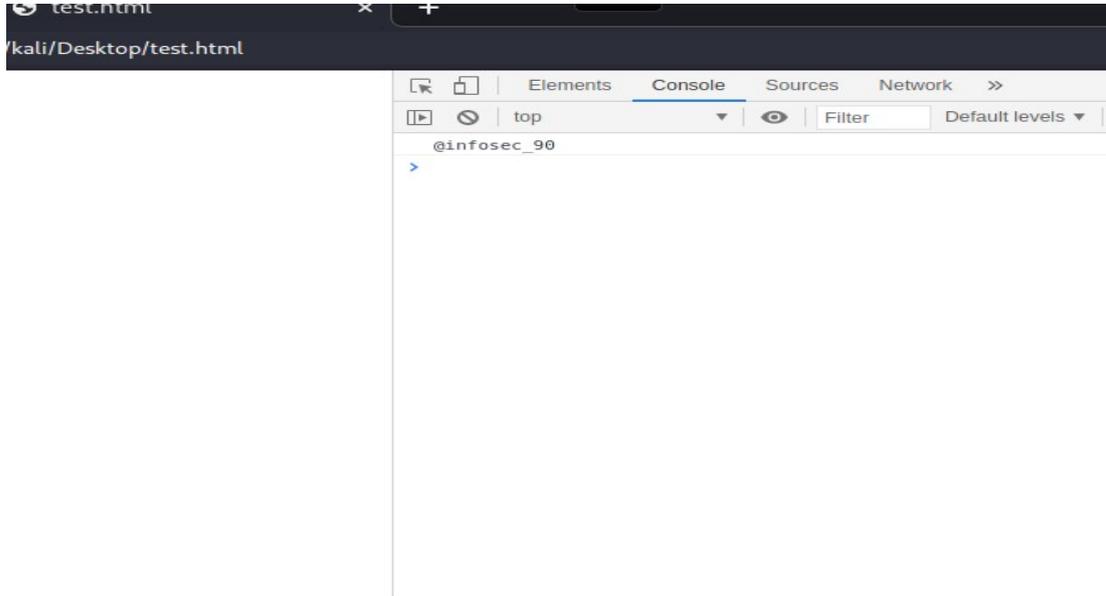
```
(kali@kali)-[~/Desktop]
└─$ grep "document.location" -r test
test/redirect1.txt:if(isValidUrl(document.location.hash.slice(1))) {
test/redirect1.txt:    document.location = document.location.hash.slice(1);
test/tinyMCE.min.js:!function(){use strict;var o=function(){for(var e=[],t=0;t<ar
r(var e=[],t=0;t<arguments.length;t++)e[t]=arguments[t];return n(r.apply(null,e))}}
on d(r){for(var o=[],e=1;e<arguments.length;e++)o[e-1]=arguments[e];return functi
t(e);return r.apply(null,n)}}var e,t,n,r,i,a,u,s,c,l,f,m,g,p,h,v,b,y=function(n){re
return!n.apply(null,e)},C=j(!1),x=j(!0),w=C,N=x,E=function(){return S},S=(r={fold:
n e},getOrThunk:t=function(e){return e()},getOrDie:function(e){throw new Error(e||
Undefined:function(){return undefined},or:n,orThunk:t,map:E,ap:E,each:function(){},
isNone()},equals_:e,toArray:function(){return[]},toString:j("none()")),Object.freez
(){return o},r=function(e){return e(n)},o={fold:function(e,t){return t(n)},is:funct
OrNull:e,getOrUndefined:e,or:t,orThunk:t,map:function(e){return k(e(n))},ap:functio
bind:r,flatten:e,exists:r,forall:r,filter:function(e){return e(n)?o:S},equals:funct
return t(n,e)}),toArray:function(){return[n]},toString:function(){return"some("+n+"
fined?S:k(e)},T=function(t){return function(e){return function(e){if(null===e)retu
e)"array":"object"===t&&String.prototype.isPrototypeOf(e)"string":t}(e)===t}},R=T
tion"),L=T("number"),I=(i=Array.prototype.indexOf)===undefined?function(e,t){return
},$=function(e,t){for(var n=e.length,r=new Array(n),o=0;o<n;o++){var i=e[o];r[o]=t(
)},W=function(e,t){for(var n=[],r=[],o=0,i=e.length;o<i;o++){var a=e[o];(t(a,o,e)?n
ngth;r<o;r++){var i=e[r];t(i,r,e)&&n.push(i)}return n},U=function(e,t,n){return F(e
{var o=e[n];if(t(o,n,e))return A.some(o)}return A.none()),K=function(e,t){for(var n
unction(e,t){for(var n=0,r=e.length;n<r;++n)if(e[n]===t)return n;return-1},Y=Array.
gth;n<r;++n){if(!Array.prototype.isPrototypeOf(e[n]))throw new Error("Arr.flatten i
,J=function(e,t){for(var n=0,r=e.length;n<r;++n)if(!0===t(e[n],n,e))return!1;return
M(t,e)}},ee=function(e){return 0===e.length?A.none():A.some(e[0])},te=function(e){
//var keyNav,
```

```
",t<r?Math.floor((r-t)/2):0);
test/mctabs.js: var pos, url = document.location.href;
(kali@kali)-[~/Desktop]
└─$
```

وجدنا شيء يستحق التتبع في ملف **mctabs.js**

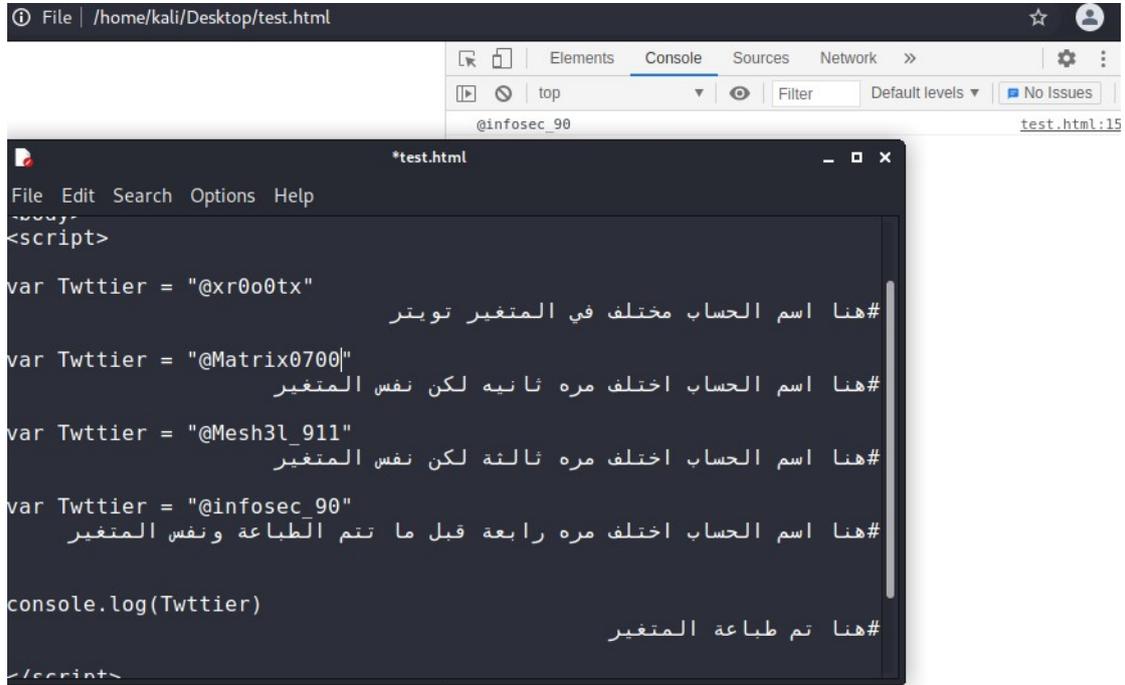
```
mctabs.js
File Edit Search Options Help
...
tabElem.focus();
}
// Show selected panel
t.showPanel(panelElem);
};
MCTabs.prototype.getAnchor = function() {
var pos, url = document.location.href;
if ((pos = url.lastIndexOf('#')) != -1)
return url.substring(pos + 1);
return "";
};
//Global instance
var mcTabs = new MCTabs();
```

وجدنا فنكشن الان نحتاج نعدل على الفنكشن لاختبارها وتتبعها وهذه من اهم الخطوات فهمك لهذه Methodology تختصر الوقت لك خصوصا عند محاولة تتبع دالة معينة او متغير في تطبيق فيه الالف الاسطر اذا لم تستطع فهمها من المره الأولى لاياس يمكنك السؤال او إعادة قراءتها الطريقة جدا بسيطة وهيا عند محاولة تتبع دالة معينة او متغير نستخدم `print` الان بشرح Methodology بطريقة مبسطة على جافا سكريبت وتعمل على كل اللغات لنفرض دخلنا على موقع وفيه ملف جافا سكريبت يطبع اسم حساب في المتصفح كيف نتتبع المتغير



الان نعمل عرض مصدر ونتتبع كود الجافا سكريبت ضروري نتتبع قبل الطباعة هل تم تنفيذ اكواد برمجية اخرى ام لا وكيف تمت الطباعة

بعد عرض المصدر وجدنا ان قبل طباعة الحساب تم تغيير الحساب اكثر من مره



```
File | /home/kali/Desktop/test.html
Elements Console Sources Network
top Filter Default levels No Issues
@infosec_90 test.html:15

*test.html
File Edit Search Options Help
<script>
var Twttier = "@xr0o0tx"
# هنا اسم الحساب مختلف في المتغير تويتر
var Twttier = "@Matrix0700|"
# هنا اسم الحساب اختلف مره ثانيه لكن نفس المتغير
var Twttier = "@Mesh3l_911"
# هنا اسم الحساب اختلف مره ثالثة لكن نفس المتغير
var Twttier = "@infosec_90"
# هنا اسم الحساب اختلف مره رابعة قبل ما تتم الطباعة ونفس المتغير
console.log(Twttier)
# هنا تم طباعة المتغير
</script>
```

المبرمج قبل طباعة النتيجة النهائية المتغير اختلف اكثر من 3 مرات وظهرت لنا النتيجة الأخيرة كان بالاول الاسم

xr0o0tx ثم اختلف الى **matrix0700** ثم اختلف الى **mesh3l_911**

وقبل الطباعة اصبح **infosec_90**

هذه الاختلافات في المتغير او الدالة لما تظهر لنا في البداية عند زيارة الموقع

فقط ظهر لنا اسم واحد وهو اسم **infosec_90**

في الكود المبرمج استخدم المتغير اكثر من مرة بطريقة مختلفة فيه كم شخص يقول كود جافا سكربت جدا واضح وتتبعه سهل صحيح كلامك لكن اذا كان فيه اكثر من 10 الالف سطر واكثر من 50 ملف جافا سكربت كيف تتبع الدالة

ببساطة نستخدم **Methodology** ال **print**

الا هيا في كل مره لتتبع المتغير في جافا سكربت او اي لغة برمجية نستخدم **print** بين الاسطر

لتتبع المتغير او الدالة الان نحاول نتتبع المتغير باستخدام امر الطباعة في جافا سكربت وهو

console.log

عدلنا الكود بحيث في كل مره المتغير يتغير يتم طباعته لنا لمعرفة اين وصل المتغير وماهي قيمته في كل مره تنفيذ اكثر في حالة كان عدد الاسطر يتجاوز 10 الالف او عدد كبير ونختصر الوقت بهذا الطريقة

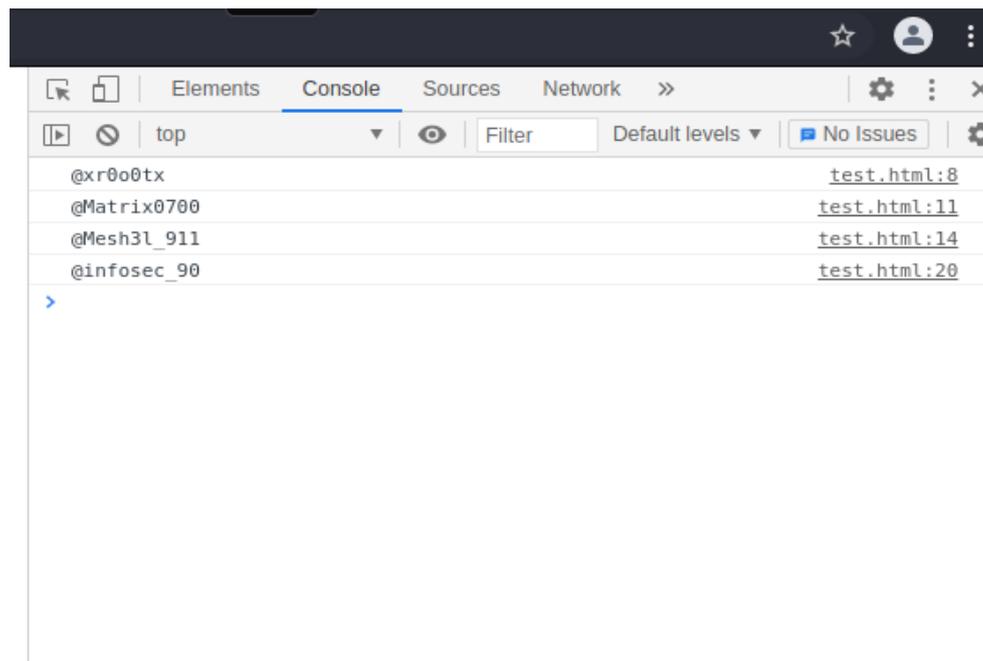
```
File Edit Search Options Help
<script>
var Twttier = "@xr0o0tx"
console.log(Twttier)

var Twttier = "@Matrix0700"
console.log(Twttier)

var Twttier = "@Mesh3l_911"
console.log(Twttier)

var Twttier = "@infosec_90"
console.log(Twttier)
</script>
```

النتيجة النهائية لمتغير Twitter



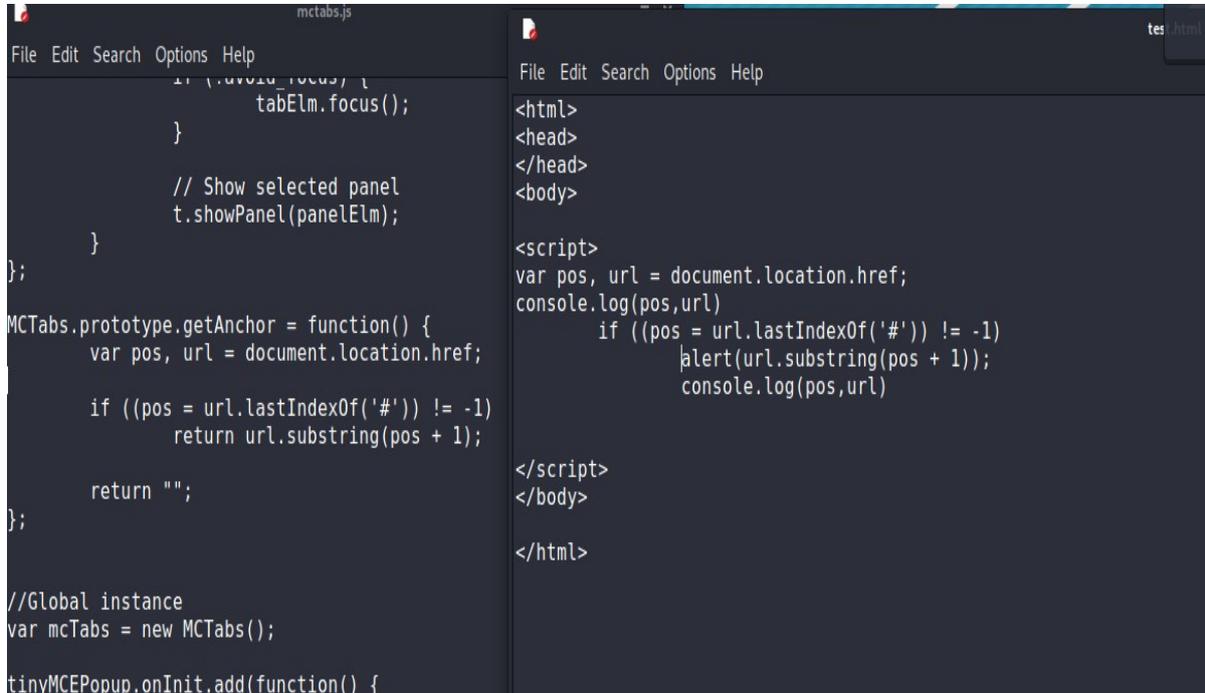
بهذا الطريقة نقدر نتتبع متغير والمراحل الا مختلف فيها وهل يمكن استغلاله او لا هذا Methodology تختصر وقت كثير في تتبع المتغيرات نرجع الان للفنكشن الا تم ايجادها ونعدل الفنكشن اول شي نستخرجها ونعيد كتابتها

شرح بسيط لها

```
MCTabs.prototype.getAnchor = function() {
    var pos, url = document.location.href;
    # pos + url تخزين فيه العنوان
    # هنا شرط ان يكون في العنوان علامة # ليتم تخزينها

    if ((pos = url.lastIndexOf('#')) != -1)
        return url.substring(pos + 1);
    # هنا يتم تخزين اي نص او عنوان بعد علامة # الموجودة في الرابط
    return "";
};
```

الان نستخرجها ونعيد كتابتها



```
mctabs.js
File Edit Search Options Help
    tabElm.focus();
}
// Show selected panel
t.showPanel(panelElm);
};
MCTabs.prototype.getAnchor = function() {
    var pos, url = document.location.href;
    if ((pos = url.lastIndexOf('#')) != -1)
        return url.substring(pos + 1);
    return "";
};
//Global instance
var mcTabs = new MCTabs();
tinyMCEPopup.onInit.add(function() {
test.html
File Edit Search Options Help
<html>
<head>
</head>
<body>
<script>
var pos, url = document.location.href;
console.log(pos,url)
    if ((pos = url.lastIndexOf('#')) != -1)
        alert(url.substring(pos + 1));
        console.log(pos,url)
</script>
</body>
</html>
```

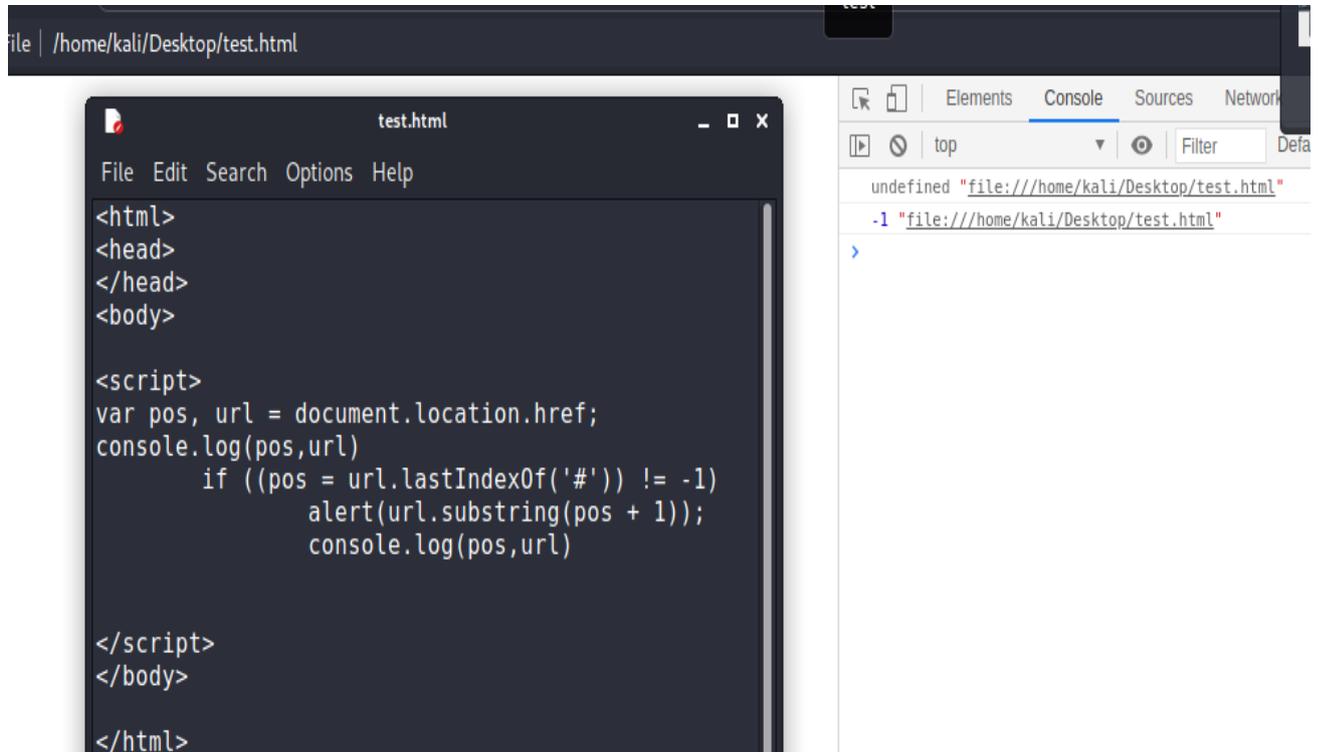
شرح لاعادة كتابة الفنكشن

```
*test.html
File Edit Search Options Help
</head>
<body>
<script>
var pos, url = document.location.href;
  console.log(pos,url)
                                #هنا نكتب امر طباعة
                                # للمتغيرين pos+url
  if ((pos = url.lastIndexOf('#')) != -1)
    alert(url.substring(pos + 1));
    # هنا نغير كلمة return الى alert ما يحتاج حفظها لانها خارج الفنكشن
    console.log(pos,url)
                                # اعادة طباعة للمتغيرين لمعرفة هل تم التنفيذ ام لا
</script>
</body>
```

الان نشغل الكود الجديد ونشوف النتيجة

ملاحظة جميع الاكواد والمرفقات في هذه الورقة مرفقة في هذا الرابط

<https://github.com/Ph33rr/attack/tree/main/Javascript>



بعد تشغيل الملف امر الطباعة اشتغل لكن التنبيه **alert** الا تم إعادة كتابته لم يعمل ؟

نشوف امر الطباعة للمتغيرين (**console.log(pos,url)**)

في اول سطر تم طباعة

مسار الملف + **undefined**

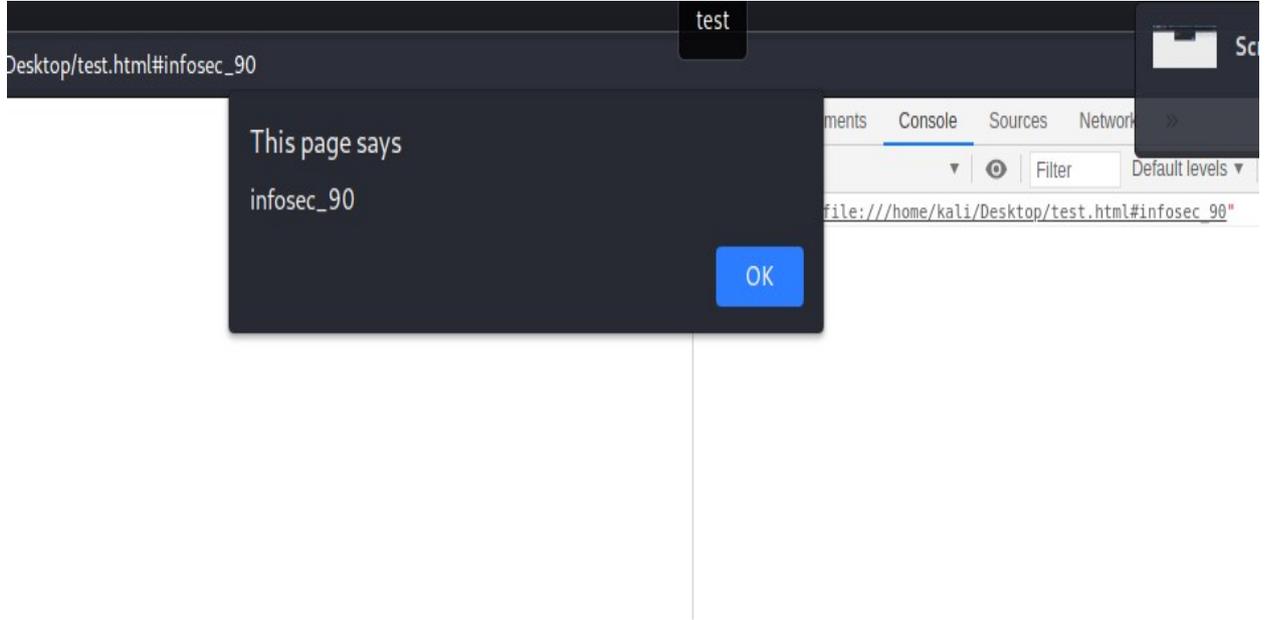
وفي السطر الثاني تم طباعة

مسار الملف + **-1**

التنبيه **alert** داخل شرط **if** الشرط يقول ان اذ المتغير **poc = -1**

التنبيه ما راح يشتغل

طيب كيف ننفذ الشرط يا أبو داحم لاجل كودنا يشتغل بسيطة موضح ان الرابط لازم يحتوي على **#** الان نجرب



بهذا الطريقة عرفنا ننفذ شرط الفنكشن

نفهم الان ان هالفنكشن تخزن أي قيمة بعد # تعمل لها return داخل الفنكشن وتخزنها . هل انتهينا اكيد لا ضروري بعدها نرجع نتتبع كامل الفنكشن واين يتم طلبها داخل السكربت لاجل استغلالها

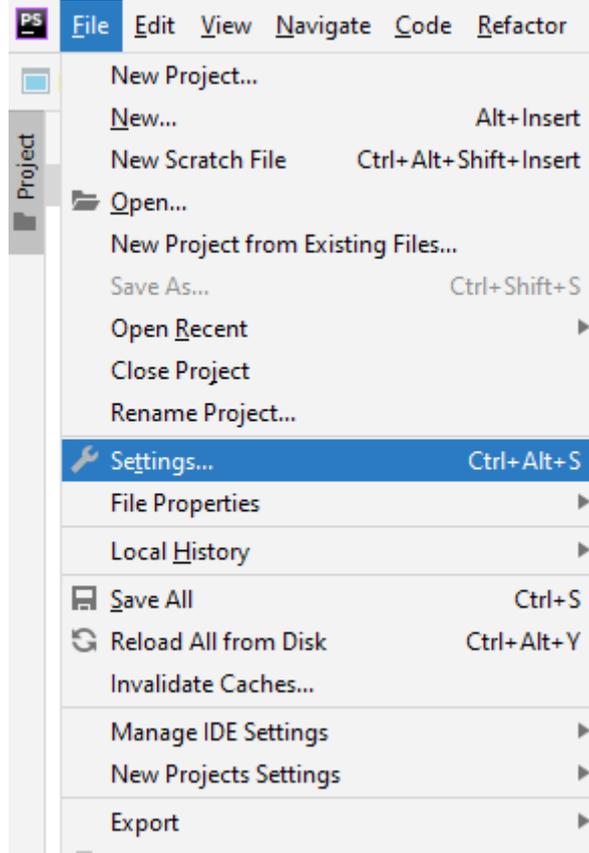
أخيرا لاتنسى اهمية regex في عمليات البحث اليدوية عن الثغرات يتم استخدام هذه الطريقة في حالة عدم ايجاد

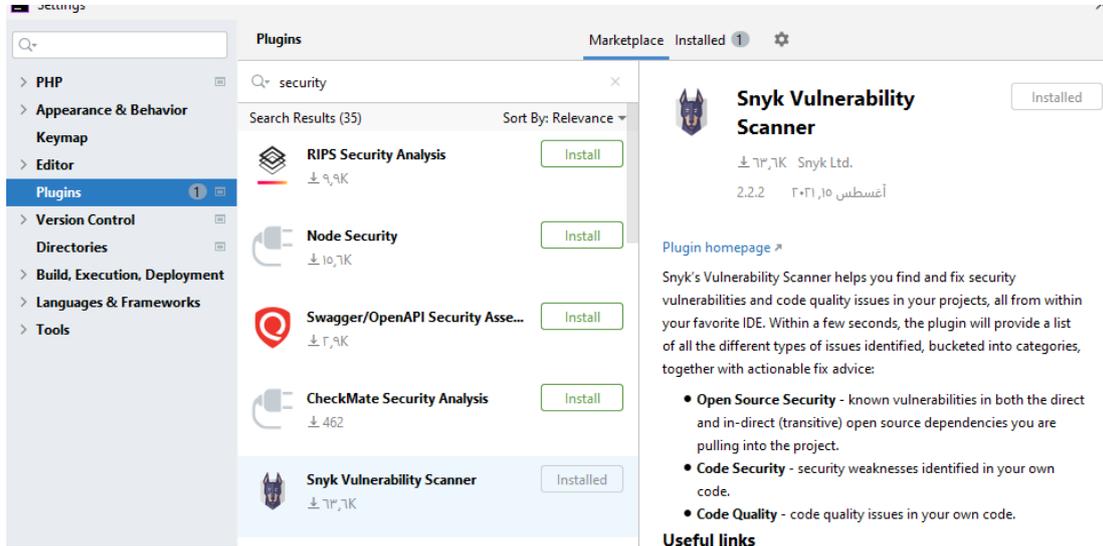
اي ثغرة عندها يتم استخدام grep والبحث عن كل دالة على حدة

لكل شخص اداعة او سكربت خاص فيه لاكتشاف هذه الثغرات بطريقته الخاصة

لنتحدث الان عن الطريقة الاسرع والاسهل وتناسب المستوى المتوسط ومطور المواقع

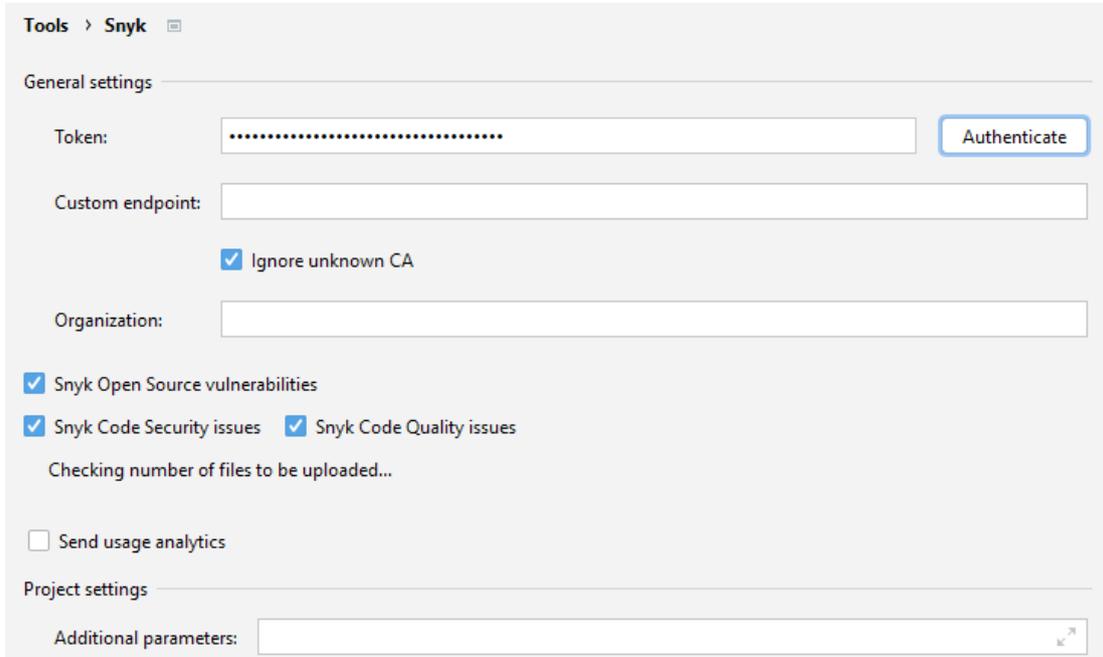
في هذا الشرح سيتم استخدام محرر **PhpStorm** ولكن معظم الإضافات ايضا تتواجد في محررات برمجية آخر توجد عدة اضافات أخرى غير التي تم شرحها لكم حرية اكتشاف هذه المتعة لنتابع الان طريقة تثبيت الإضافة





نُثِبَت الاضافة

الان نعمل كونفج للاضافة أولا بالتسجيل في الموقع عن طريق النقر على **Auth** او اذا كان عندك حساب



بعدها من الحساب نعدل بعض الاعدادات اذا وجدت خطأ عند الفحص تفعل اللغات الا تحتاجها في الفحص

General

Integrations

Languages

Snyk Code

Infrastructure as code

Language settings

NAME	Edit settings
JavaScript	Edit settings
Python	Edit settings
.NET	Edit settings
PHP	Edit settings

الآن نعمل فحص للملفات الجافا سكريبت طبعاً عن طريق فتح نفس المجلد في

phpstrom

The screenshot shows the Snyk IDE interface. On the left, a sidebar displays a list of issues: 'Open Source Security (error)', 'Code Security - 2 vulnerabilities: 2 high', 'line 279: Hardcoded Secret', and 'line 2563: Server-Side Request Forgery (SSRF)'. The 'Server-Side Request Forgery (SSRF)' issue is highlighted in blue. On the right, a detailed view of this issue is shown, including the title 'Server-Side Request Forgery (SSRF)', the severity 'Vulnerability | CWE-918', and a description: 'Unsanitized input from data from a remote resource flows into request, where it is used to result in a Server-Side Request Forgery vulnerability.' Below the description, a 'Data Flow - 7 steps' section shows a code snippet: '1 editors.default.js:2563 | fm.request({', '2 editors.default.js:2571 | ...done(function(data) {'.

References :

<https://github.com/BlackFan/client-side-prototype-pollution>
<https://rules.sonarsource.com/javascript/RSPEC-5696?search=xss>