

Web Application Penetration Testing



Fourth Note

By TheSecDude

Web Application Penetration Testing جلسه چهارم

تا اینجا ما مقدمات مربوط به تست نفوذ وب اپلیکیشن را توضیح دادیم و از این جلسه به بعد میریم یک قدم جلوتر . ما باید یاد بگیریم که چطوری اطلاعات یک تارگت رو جمع اوری کنیم . چه اطلاعاتی از یک تارگت برای تست نفوذ اهمیت دارد ؟ چه چیزهایی میتوانه به ما کمک کنه ؟ این کار رو بهش میگن Reconnaissance یا جمع اوری اطلاعات . عمل Reconnaissance دو حالت دارد :

- 1. Active Reconnaissance
- 2. Passive Reconnaissance

در این نوع جمع اوری اطلاعات ما با تارگتمنون تعامل برقرار میکنیم . برای تارگت پکت ارسال میکنیم و جوابهای تارگت رو دریافت میکنیم و نسبت به رفتار تارگت و جوابهایی که به ما میده اطلاعاتی رو ازش جمع اوری میکنیم . مثلا پورت اسکن یک تارگت از نوع جمع اوری اطلاعات به صورت اکتیو هست چرا که ما پکت هایی رو بر روی پورت های مختلف به تارگت ارسال میکنیم و جواب تارگت به ما میگه که ایا اون پورت باز هست و یا که بسته هست و تارگت پکت های ما رو دریافت میکنه و اونها رو لاغ میکنیم و یا مثلا توی یک Domain سعی میکنیم تمام لینک ها و End Point های تارگت رو بدست بیاریم و خب به تمام لینکهای داخل وب اپلیکیشن یک درخواست HTTP ارسال میکنیم و نسبت به جواب تعیین میکنیم که ایا اون End Point وجود داره یا نه . در این نوع جمع اوری اطلاعات مهاجم به خاطر تعامل مستقیم با تارگت قابل شناسایی است چرا که تارگت Request های مهاجم رو لاغ میکنه .

در این نوع جمع اوری اطلاعات ما بدون تعامل با تارگت سعی میکنیم اطلاعات رو جمع اوری کنیم . در این روش از اطلاعات موجود در اینترنت استفاده نمیشود . از مهم ترین متدهایی که در این نوع روش استفاده میشه OSINT هست که مخفف Open Source Intelligence میباشد . در این روش از منابع رایگان و حتی غیر رایگان داخل اینترنت بهره میگیریم . مثلا از سایت میتوانیم اطلاعات ثبت کننده یک Domain رو بدست بیاریم یا اینکه از طریق سایت هایی مثل <https://ripe.net> اطلاعات SubDomain های یک تارگت رو بدست بیاریم یا اینکه از گوگل دورک ها استفاده کنیم . چنین منابعی در اینترنت بسیار زیاد هست . یک فریمورک به نام OSINT Framework به ادرس <https://osintframework.com> وجود دارد که تا حد ممکن این منابع رو در اختیار شما قرار میده . OSINT کردن یک هنر است و خب هرجی خلاقیت شما بیشتر باشه شما اطلاعات بیشتری بدست خواهید اورد و خب این افزایش خلاقیت فقط و فقط با تمرین حاصل میشه و خب قاعدها به میزان توانایی هوشی مهاجم بستگی داره .

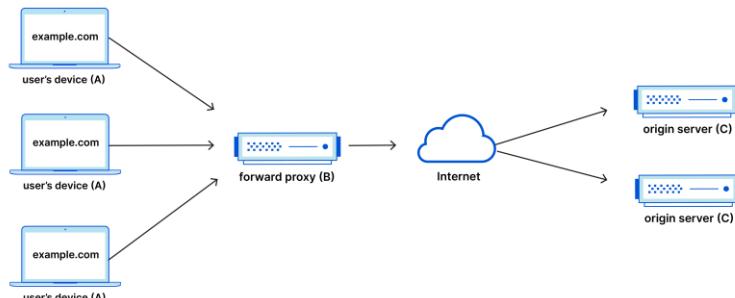
به نظر من مهمترین بخش مربوط به تست نفوذ و کلا هک کردن یک تارگت Reconnaissance است و خب طولانی ترین بخش هم محسوب میشه و هرچه شما به عنوان یک هکر و مهاجم اطلاعات بیشتر و صحیحتری از یک تارگت بدست بیارید حملات شما و نفوذ شما موفقیت امیز تر خواهد بود . دقت کنید که به جای اینکه عده و وقت خود را به صورت کورکرانه به حمله کردن بگذرانید بهتر است که وقت بیشتری بدست خواهید اورد و خب این افزایش خلاقیت فقط و فقط با تمرین حاصل میشه و خب قاعدها به میزان توانایی هوشی مهاجم بستگی داره .

قبل از اینکه عمیق تر وارد مبحث Reconnaissance بشیم باید از اصطلاحاتی اطلاعاتی داشته باشیم . این اصطلاحات رو به وفور در وب اپلیکیشن ها میبینید و باشیستی از کاربرد انها اگاهی داشته باشد . در ادامه لیستی از این اصطلاحات رو توضیح میدهیم .

کامپیوتری . پروکسی درواقع یک واسط محسوب میشه . در مسائل سیاسی و نظامی گرفته تا مفاهیم Forward Proxy یا Proxy چیست ؟ کلمه Proxy رو خیلی جاها میشنویم . از مسائل سیاسی و نظامی گرفته تا مفاهیم کامپیوتری . پروکسی درواقع یک واسط محسوب میشه . در مسائل سیاسی و نظامی به گروهی مثلا A که به نمایندگی از گروهی دیگر مثلا B یک عملیات نظامی رو انجام میده گروه A رو پروکسی گروه B می نامند که مثالش هم زیاده . در مفاهیم کامپیوتری یک پروکسی سرور چیزیست که مابین یک کلاینت و یک Server قرار میگیره و کلاینت درخواست گرفتن Resource از سرور رو به پروکسی سرور میفرسته و پروکسی سرور از طرف خودش درخواست دریافت Resource رو به سرور ان ارسال میکنه و Resource رو از سرور میگیره و به کلاینت میده . اگه بخواه که مثلا بزنم ازش ، توی اینترنت کشور ما که کمتر منبعی رو میبینید که توسط فیلترینگ داخلی بن نشده و یا توسط

تحریمهای خارجی مسدود نشده ما زیاد از این پروکسی ها استفاده میکنیم . مثلا تلگرام یکی از پرطرفدار ترین اپلیکیشن های موبالیست که تقریبا همه دارن و استفاده میکنن ولی توی کشور ما فیلتره و خب ما از پروکسی جهت اتصال بهش میگیریم و میایم و از یک پروکسی که فیلتر نیست و تلگرام هم برای اون فیلتر نیست به عنوان واسط خدمون و سرور های تلگرام استفاده میکنیم و این پروکسی درخواست های ما رو از ما میگیره و به تلگرام ارسال میکنه و تلگرام پاسخ میده و پروکسی پاسخ رو میگیره و به ما میده و خب ما فیلترینگ رو دور میزنیم و این یکی از قابلیت های پروکسی هاست . به طور کلی هر چیزی که مابین یک کلاینت و یک سرور قرار بگیره و درخواست های کلاینت رو بگیره و پاسخ ها رو هم از سرور بگیره و تعامل ما بین سرور و کلاینت را برقرار کنه پروکسی میگن .

Forward Proxy Flow



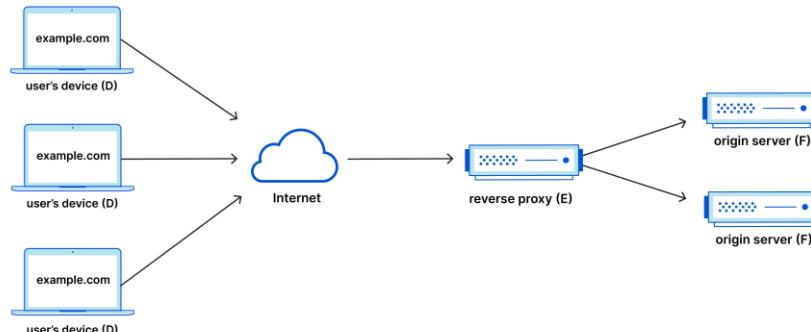
ز

از دلایل از استفاده از Forward Proxy ها میتوانیم به لیست زیر اشاره کنیم :

1. گذر از محدودیت های ایجاد شده توسط دولتها
2. جلوگیری از دسترسی به برخی محتواهای خاص
3. جهت مراقبت از هویت انلاین کلاینت و جلوگیری از شناسایی شدن

Reverse Proxy چیست ؟ یکی Reverse Proxy های کلاینتها رو به جای سرور دریافت میکنه و با Forward Proxy ها مقاومت است چرا که اونها در جلوی کلاینت ها قرار میگرفتند . در حالت Reverse Proxy دارند تمامی کلاینتها به وب سرور ها به Reverse Proxy داده میشه و Reverse Proxy این درخواست ها رو به Server ها میده و سرور ها پاسخ هر درخواست را اماده میکنند و به Reverse Proxy میدن و Reverse Proxy اون پاسخ ها رو به تک کلاینت ها ارسال میکنه . تقاؤت ما بین Forward Proxy و Reverse Proxy یک تقاؤت کم ولی مهم است . Forward Proxy ها جلوی کلاینت های میشین و ضمانت میکن که هیچ سروری با انها به صورت مستقیم در تعامل نباشد و در حالی که Reverse Proxy ها در جلوی سرور های قرار میگیرند و ضمانت میکن که هیچ کلاینتی با سرور ها به صورت مستقیم در تعامل قرار نگیرد . تصویر زیر مفهوم Reverse Proxy رو روشن تر میکنه :

Reverse Proxy Flow



تمام درخواست های کلاینت های D در تصویر بالا به Reverse Proxy (E) داده میشه و سپس این تمام اونها رو به Origin Server های F میده . وب خب در تصویر بالا میتوان حکم Load Balancer را داشته باشن . اما دلایل استفاده از Reverse Proxy ها چیست ؟ چرا باید از اونها در وب اپلیکیشن ها استفاده کرد ؟

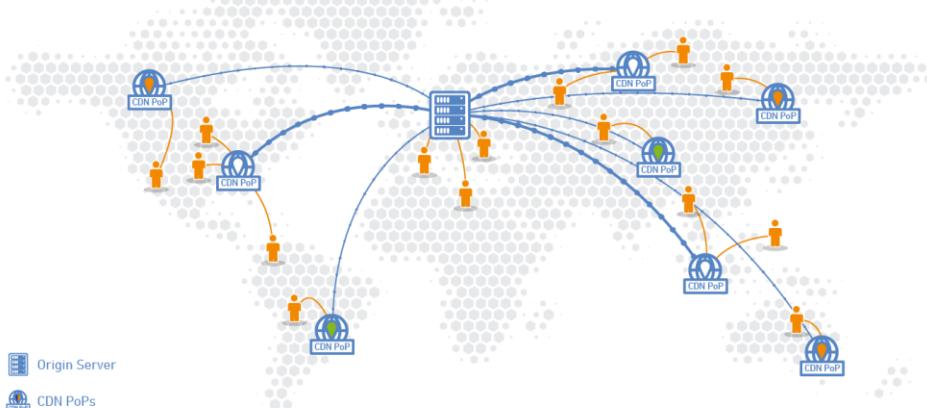
۱. Load Balancing: یک وبسایت معروف که میلیونها کاربر روزانه دارد ممکن است که همه ترافیک ورودی را توسط یک Origin Server مدیریت کند. به جای این کار میان و این وبسایت را مابین مجموعه‌ای از سرورها توزیع میکند و همه این سرورها Request های یک وبسایت را مدیریت خواهد کرد. برای این کار میان و از یک Reverse Proxy به عنوان Load Balancer استفاده میکند و این Reverse Proxy میاد و ترافیک ورودی به وبسایت را مابین مجموعه‌ای از سرورها تقسیم میکند.

و جلوی بیش از حد استفاده شدن از یک سرور خاص را میگیرد به عبارتی دیگه میاد و Request های جدید را به سرورهای مختلف میده تا سرورها بیش از حد Overloaded نشون و کندی برای وبسایت ایجاد نشه. حتی اگر یک سرور هم به مشکل بخورد Load Balancer تشخیص میده و درخواست ها رو به اون ارسال نمیکنه و سعی میکنه مابین سرورهای باقی مانده تقسیم کنه. این کار خیلی به وبسایت ها کمک میکنه تا سرعت بیشتری برای دادن پاسخ به کاربران داشته باشدند.

۲. Protection from attacks: با استفاده از یک Reverse Proxy یک وبسایت یا یک سرویس دیگر نیازی نداره که IP Address حقیقی Origin Server های خودش رو فاش کنه. این فاش نشدن IP Address ها نفوذ یک نفوذگر به یک سرور را بسیار سخت تر میکنه و همچنین اون سرور رو در مقابل حملات DDoS محافظت خواهد کرد. با این کار هکرها فقط میتوان Reverse Proxy رو مورد حمله قرار بدم مثلاً CDN های Cloudflare را که اونها هم نسبتاً امنیت بالاتری دارند و نفوذ بهشون به خاطر تیم قدرتمندی که پشتیشون هست تقریباً بسیار سخت خواهد بود.

۳. Global Server Load Balancing (GSLB): همانطور که مشخص است این کاربرد هم یک نوع Load Balancing است اما به صورت جهانی. در واقع چنین است که میان و یک وبسایت را مابین چندین سرور در جاهای مختلف جهان توزیع میکند و هر کاربر جهت دسترسی به سرورهای اصلی به Reverse Proxy نزدیک به خودش متصل میشود و درخواست های هر کاربر بسته به اینکه در کجای جهان است به Reverse Proxy نزدیک به اون ارسال خواهد شد و این موجب میشه که سرعت دسترسی کاربران به Origin Server(s) بسیار بیشتر بشه.

۴. Caching: یک Reverse Proxy همچنین میتوانه محتوا را Cache و پاسخ های کاربران را سریع تر ارسال کنه. مثلاً اگه یک کاربر توی پاریس به یک Reverse-Proxied وبسایت دسترسی پیدا کنه که این Origin Server(s) اون وبسایت توی Los Angeles هست کاربر ممکنه که در واقع به یک Reverse Proxy Server محلی داخل Paris متصل شده باشه که این Reverse Proxy با Origin Server(s) داخل LA ارتباط دارد. این Reverse Proxy میتوانه پاسخها رو Cache یا به صورت موقت ذخیره کنه. کاربران دیگری که دوباره از پاریس به این Reverse Proxy Server جهت دسترسی به وبسایت اصلی درخواست رو ارسال میکنند در واقع نسخه Cache شده پاسخ را خواهند گرفت و Reverse Proxy Server دیگه برای درخواست های تکراری که پاسخ آنها Cache شده به این Origin Server(s) درخواستی ارسال نخواهد کرد. این موجب میشه که کارابی وبسایت بسیار سریعتر بشه و کاربران پاسخهای خودشون رو سریعتر دریافت کنند. اما توی این مبحث این سوالات هم بیش میاد که آیا باید هر چیزی رو Cache کرد؟ مثلاً آیا باید پنل کاربری یک کاربر رو Cache کرد و به کاربران دیگه هم اون رو ارسال کرد؟ خب نه قاعتنا و توی چنین مواردی میان و پیکربندی هایی رو اعمال میکنن.



۵. SSL Encryption: Encrypt کردن و Decrypt SSL یا TLS ارتباط های هر کلاینت از لحاظ ریاضیاتی منابع Origin Server را مصرف میکنه. یک Reverse Proxy را میتوان پیکربندی کرد که تمام درخواست های ورودی را واسه سرور Decrypt کنه و تمام جواب های خروجی رو که قرار است به کلاینت ها برسه Encrypt کنه و این کار موجب آزاد شدن منابع زیادی برای Origin Server خواهد شد.

اما چطوری باید یک Reverse Proxy Server را پیاده سازی کرد؟ برخی از کمپانی ها Reverse Proxy های خودشون رو میسازند و این کار نیازمند مهندسین نرم افزار و سخت افزار ماهری است و همچنین سخت افزار فیزیکی نسبتاً خوبی را میخواهد. راحت ترین راه برای پیاده سازی Reverse Proxy استفاده از CDN Provider های موجود است. مثلاً Cloudflare CDN تمام مواردی را که در بالا گفتم

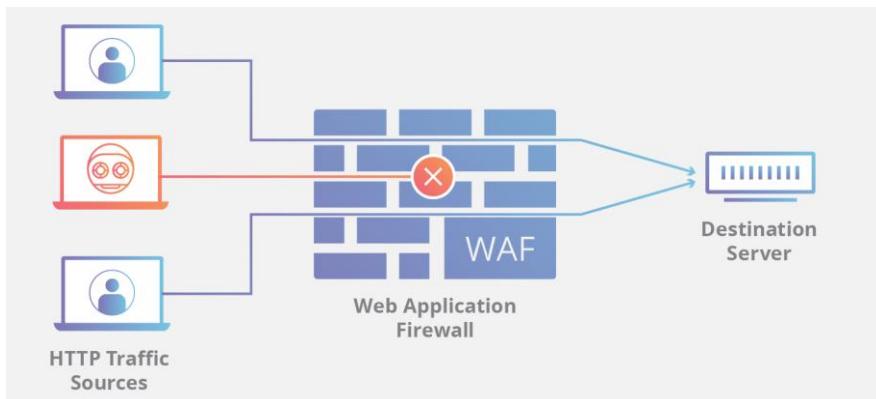
برای کلاینت های خودش فراهم میکند و هم نسخه رایگان داره و هم نسخه غیر رایگان . همچنین میتوانید از ArvanCloud که یک CDN داخلى است هم استفاده کنید ولی نمیتوانید ازش انتظار امکانات Cloudflare و ... رو داشته باشید .

بسیار خوب تمام چیز هایی که نیاز بود درباره Reverse Proxy و تارگت است . ما باید بدانیم که آیا تارگت از در حین Reconnaissance کردن یک تارگت بفهمیم همین استفاده از Reverse Proxy توسط تارگت است . ما باید بدانیم که آیا تارگت از Reverse Proxy استفاده میکنه یا نه ؟ آخه اهمیت داره چرا که اگه چنین باشه ما به عنوان یک مهاجم باید تمام تلاشمون رو بکنیم که IP Address پشت CDN تارگت رو بدست بیاریم و گرنه دستمون و اسه نفوذ زیاد باز نخواهد بود . پسدا کردن IP Address پشت ابر یا همون پشت CDN هم روش های خودش رو داره و خب در آینده در موردش حرف خواهیم زد . حالا اینکه چطوری بدانیم یک تارگت داره از Reverse Proxy استفاده میکنه یا نه هم قدم اوله که راههای مختلفی داره و شاید هم گاهی اوقات امکانش وجود نداشته باشه . مثلاً یکی از راهها وجود برخی از Response HTTP Headers است . با این هم بیشتر ور خواهیم رفت .

گفتم که یکی از کاربردهای Reverse Proxy ها اینه که Incoming Load Balancing را انجام بدن و ترافیک Load Balancing را اینه که Response کاربری را در پردازش تمام Request هاروی یک سرور نیست و اینطوری سرورها با سرعت بیشتر و مشکل کمتری Response کاربری را رو میفرستند . اما این کار میتوانه در صورتی که به درستی پیکربندی نشده باشه مشکلاتی رو پیدا بیاره مثلاً فرض کنید سه تا سرور وجود داره که هر سه به یک Reverse Proxy متصل هستند و این Reverse Proxy عمل Load Balancing را انجام میده . حال اگر بر روی یکی از این سرورها یک Session برای ایجاد بشه باشیستی بر روی دو سرور دیگه هم همین ایجاد بشه و گرنه Request های کاربر اگر به سرور هایی ارسال بشه که Session کاربر رو ندارند کاربر به مشکل خواهد خورد و یا حتی ممکن است سرورها و Reverse Proxy به درستی پیکربندی نشود و یک صفحه رو بعد از چندبار درخواست به یک کاربر Unauthorized Misconfiguration گزارش بشه .

بیشتر توصیه میشه که Reverse Proxy خود را از CDN های مختلفی مثل Cloudflare و ... بگیرید ولی خب اگه بخواهد خودتون چنین کاری رو انجام بدهید میتوانید از وب سرور هایی مثل Nginx استفاده کنید و کافیه که پیکربندی های مربوطه رو بر روی اون پیاده سازی نمایید .

WAF یا Web Application Firewall چیست ؟ یک WAF یا یک Web Application Firewall Monitoring Filtering ترافیک بین یک وب اپلیکیشن و اینترنت از اون وب اپلیکیشن مراقبت میکند . معمولاً WAF ها وب اپلیکیشن ها رو در مقابل حملاتی مثل Cross-Site Forgery, Cross-Site-Scripting (XSS), File Inclusion, SQL Injection , ... حفاظت می کند . WAF یک محافظ پروتکل های لایه 7 در OSI که همون لایه Application هست می باشد و طوری طراحی نشده که در مقابل همه حملات دفاع رو انجام بده و درواقع WAF ها جزئی از ابزارهایی هست که دامنه حملات یک وب اپلیکیشن رو کاهش میده . اگر یک WAF رو پیاده سازی کنیم، این WAF در جلوی Web Application قرار میگیره و به عنوان یک سپر مابین وب اپلیکیشن و اینترنت عمل خواهد کرد . اگهه یادتون باشه که قطعاً باید یادتون باشه، Proxy ها از هویت یک کلاینت محافظت میکنند و مابین Client و اینترنت قرار میگیره ولی WAF ها یک نوع Reverse Proxy هستند و مابین Origin Server ها و اینترنت قرار میگیرند و ترافیک ورودی از طرف کاربران رو بررسی میکنند و موارد مشکوک رو شناسایی و Filter میکنند . یک WAF از طریق مجموعه ای از قوانین عمل میکنند که به آنها Policy گفته میشود و میتونم بگم که WAF حفرات امنیتی رو برطرف نمیکنه و فقط با فیلتر کردن ترافیک های مشکوک (حاوی Payload خاص) از Exploit شدن اسیب پذیری هایی که ممکن است وجود داشته باشد جلوگیری میکند .



اگه بخواهیم مثالی بزنم از عملکرد WAF ها میتونم بگم که مثلاً میان توی یک Request در قسمت Path علامت ' وجود داشت بیا و این علامت رو حذف کن و یا بیا و اون Request را Drop کن و یا میگن که اگه توی یک Request در داخل Path از عبارت <script> استفاده شده بود بیا و این عبارت رو تبدیل کن به HTML Entities و خب خطرش رو رفع کن . از این جور کارا که با کمی بازی کردن با Payload میشه گاهی اوقات دور زد و Payload رو اجرا کرد . این قسمت نیاز به خلاقیت دارد .

راستی اینو هم بگم که، گفتیم WAF ها تضمین صد درصدی برای محافظت در مقابل همه حملات نمیده و حملاتی مثل IDOR, Business Logic و ... رو اصلاً نمیتوانه تشخیص بدنه چه برسه به اینکه Filter کنه . ما دو نوع WAF داریم که عبارت اند از :

1. Blacklist WAFs

2. Allowlist WAFs (whitelist)

یک Blacklist WAF در واقعی فایروالی است که بر اساس Negative Security Model یا یک Blacklist کار میکنه و در خودش لیست داره شامل حملات مشخص (Payload های خاص) و درواقع هر وقت که اون حملات رو دریک ترافیک شناسایی کرد اون ترافیک رو Drop میکنه و اجازه رسیدن اون به وب سرور رو نمیده . معمولاً با این نوع WAF ها دیدار خواهیم کرد چون بر روی عموم وب اپلیکیشن ها چنین WAF هایی وجود دارد . در مقابل یک Allowlist WAF، فایروالی است که بر اساس Positive Security Model یا یک Allowlist عمل خواهد کرد . این نوع فایروالها فقط اجازه میدهند ترافیک هایی به وب سرور برسد که در Allowlist آنها تعریف شده باشد و در صورتی که یک ترافیک در Allowlist تعریف نشده باشد Drop خواهد شد . Allowlist WAF ها و Blacklist WAF ها هر دو دارای معایب و مزایایی هستند و به همین علت هست که بسیاری از WAF ها ترکیبی از هر دوی آنهاست . علاوه بر دو نوع WAF بالا که خدمتتون گفتم که به خاطر نحوه عملکرد WAF ها این تقسیم بندی رو کرده بودند ما از لحاظ پیاده سازی هم تقسیماتی برای WAF ها داریم که فقط نام میرم و توضیح نمیدم حقیقتاً و لینک میدم هر کی خواست بره بخونه :

1. Network-Based WAFs

2. Host-Based WAFs

3. Cloud-Based WAFs

اگه میخواید بدونید اینها به چه شکل هستند میتوانید به لینک <https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf> مراجعه کنید .

درنهایت هم بگم که WAF ها هم میتوانن به شکل نرم افزاری باشند و هم میتوانن به شکل سخت افزاری ارائه شوند و خب بستگی به نیاز ما داره که کدام رو انتخاب کنیم .

بسیار خوب، فهمیدیم که WAF ها چی هستند و چطوری کار میکنند . اما ما به عنوان یک هکر چیزی که نیاز داریم اینه که تشخیص بدیم آیا یک وب اپلیکیشن از WAF استفاده میکنه یا نه ؟ یکی از ابزار هایی که واسه این کار زیاد استفاده میشه و توی کالی لینوکس هم به صورت پیش فرض نصب هست . این ابزار به سه شکل عمل میکنه که به عبارت زیر است :

1. میاد و چند HTTP Request خیلی ساده رو میفرسته به تارگت و Response رو میگیره و اون رو آنالیز میکنه و هدرهای داخلش

رو برسی میکنه و با لیستی از Firewall ها که توی خودش داره قیاس میکنه . اگه شbahati مابین لیست خودش و Response دریافتی بود مینویسه که یک فایروال با نام فلان از این وب اپلیکیشن محافظت میکنه :

```
(kali㉿kali)-[~]
$ wafw00f https://...
[+] Checking https://...
[+] The site https://... is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
```

میبینید که توی مثال بالا تشخیص داد که این وبسایت پشت Cloudflare WAF هست .

2. در صورتی که مرحله اول موفقیت آمیز نبود میاد و تعدادی HTTP Request مشکوک و حاوی Payload را در لود رو میفرسته به تارگت و Response رو چک میکنه و وجود WAF یا عدم وجودش رو تشخیص میده .

3. در صورت موفقیت آمیز نبودن هر دو مرحله اول میاد و دوباره برسی میکنه و درنهایت جوابش رو برآتون مینویسه .

اگر هم بخوایم بینیم Firewall هایی رو میتوانه تشخیص بدنه میتوانید از دستور زیر استفاده کنید تا لیستی از Firewall های قابل تشخیص wafw00f رو ببینید و لیست نسبتاً بلندی هست و در تصویر زیر من فقط قسمتیش رو نشون دادم :

WAF Name	Manufacturer
ACE XML Gateway	Cisco
aeSecure	aeSecure
AireeCDN	Airee
Airlock	Phion/Ergon
Alert Logic	Alert Logic
AliYunDun	Alibaba Cloud Computing
Anquanbao	Anquanbao
AnYu	AnYu Technologies
Approach	Approach
AppWall	Radware
Armor Defense	Armor
ArvanCloud	ArvanCloud
ASP.NET Generic	Microsoft
ASPA Firewall	ASPA Engineering Co.
Astra	Czar Securities
AWS Elastic Load Balancer	Amazon
AzionCDN	AzionCDN
Azure Front Door	Microsoft
Barikode	Ethic Ninja
Barracuda	Barracuda Networks
Bekchy	Faydata Technologies Inc.

راستی لینک گیتهاب این برنامه هم <https://github.com/EnableSecurity/wafw00f> هست . میتوانید بزید و Clone کنید و اگه پایتون بلدید باهاش ور بزید و از سازوکارش بیشتر سر در بیارید .

ابزار whatwaf ؟ این هم یکی دیگه از ابزارهایی هست که میتوانیم استفاده کنیم و سعی کنیم باهاشون Firewall ها را تشخیص بدیم . این ابزار به صورت پیش فرض روی کالی نصب نیست و میتوانید از گیتهابش اون رو دریافت کنید .

<https://github.com/Ekultek/WhatWaf>

این ابزار نسبتاً پیشرفته تر از wafw00f هست چرا که امکانات بیشتری را در اختیار ما قرار میده . از کارهایی که میشه با این ابزار انجام داد، مثلاً میتوانید بیاین و یک Payload خاص را بسیز بذید تا برآتون روی وب اپلیکیشن اجرا کنه و نتیجه رو بررسی کنه و تشخیص بده که ایا WAF داره یا نه و یا مثلاً میتوانید بیاید و یک User-Agent خاص رو توی Request ها بنویسید و Response های دریافتی رو بررسی کنید . اما به نظرم از هردوی این ابزارها استفاده کردن میتوانه نتیجه بهتری را در اختیار ما قرار بده .

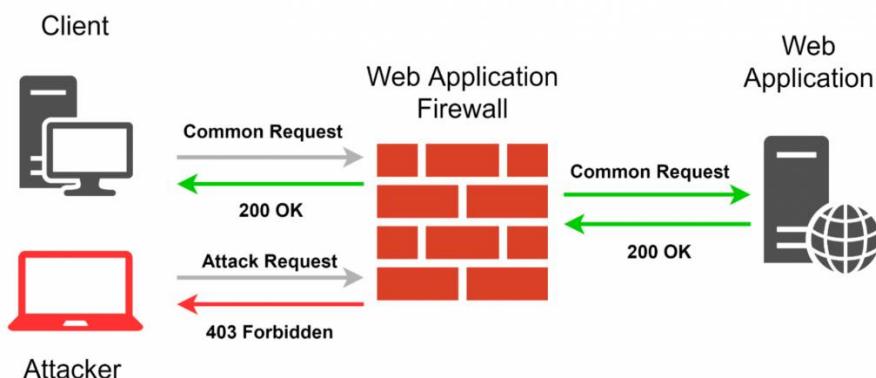
اما اینکه چطوری این برنامه ها کار میکنن یه طورایی اهمیت داره ولی خب لازم نیست فعلاً بدونیم، روشهای تشخیص هر فایروال میتوانه متفاوت باشه مثلاً یک فایروال یک Header خاص رو توی Response بر میگردونه و یک فایروال دیگه یک Header خاص دیگه رو بر میگردونه، این ابزارها ساخته شدن تا اینها رو تشخیص بدن . همه چیز بر اساس رفتار وبسایت نسبت به Request های ما به اون وب سرور تعیین میشه . در واقع یک Active Reconnaissance محسوب میشه .

تشخیص فایروال با استفاده از nmap ؟ بله nmap هم اسکریپت‌هایی جهت تشخیص فایروال داره و میشه ازش استفاده کرد . نام این اسکریپت

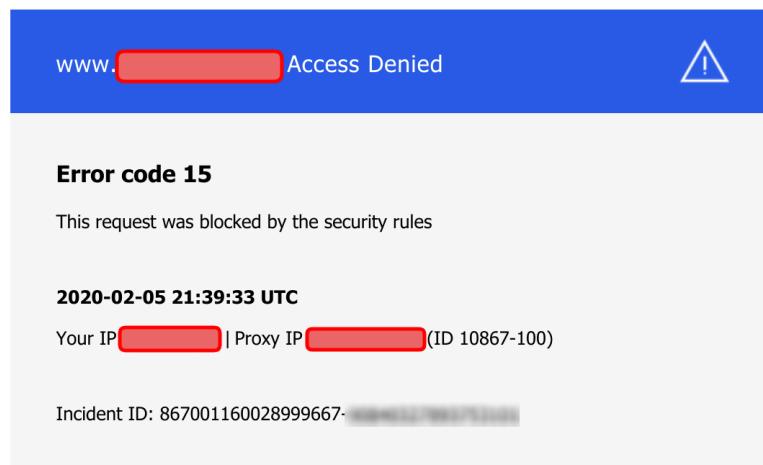
است و طریقه استفاده ازش به شکل زیر می باشد :

```
kali@kali:~$ sudo nmap --script=http-waf-fingerprint targetweb.com
```

اما چطوری به صورت دستی WAF رو تشخیص بدیم ؟ راه تشخیص یک WAF به صورت دستی بستگی به خلاقيت مهاجم داره . یکی از راهها ارسال یک HEAD Request به وب اپليکيشن هست که در برخی موارد نام WAF رو در مولفه Server پاسخ، برای ما ارسال خواهد کرد و روش دیگه هم اينه که بيايم و یک Request مشکوك و حاوی Payload به وب اپليکيشن بفرستيم و خب درصورتی که WAF داشته باشه قاعتنا خطايي رو برای ما ارسال ميکنه و از طریق این خطا میشه فهمید که WAF وجود داره یا نه .



این خطاهارو میشه توی صفحات وب هم دید . یعنی یک صفحه رو بازکنیم که اجازه نداریم و خب درصورتی WAF داشته باشه و صفحه خطای WAF رو Customize نکرده باشن، میشه از طریق اون صفحه وجود WAF و نوع ان را تشخیص داد .



مثلًا صفحه بالا برای فایروال Imperva هست و خب هر فایروال صفحه خودش رو داره مثلًا صفحه زیر هم واسه فایروال Fortiweb هست :



یکی دیگه از روش‌های تشخیص فایروال ها این هست که برخی از اونها توی کوکی ها یک مورد رو اضافه میکنند. مثلا Fortiweb میاد و یک کوکی به نام cookiesession رو اضافه میکنه.

Request Cookies		<input type="checkbox"/> show filtered out request cookies								
Name	Value	Do...	Path	Expir...	Size	HttpOnly	Se...	S...	Pri	
cookiesession1	253EA5E9N5A2PBFLCTCP6HK1B...	ho...	/	Sessi...	46	✓				Me

خلاصه اینکه تشخیص Firewall مهم است و خب به صورت دستی روش‌های زیادی وجود داره و برخی از روشها مختص یک فایروال خاص هست. نظر من اینه که ابتدا از طریق ابزارها سعی کنید که فایروال رو تشخیص بدید و درصورتی که تشخیص درست نبود سعی کنید به صورت دستی این کار رو انجام بدید. معمولاً ابزار هایی که واسه این کار هست میتونه کمک کننده باشه.

اگر لازم بیدی که بیشتر بدونی به لینک <https://geekflare.com/find-which-waf-is-protecting-a-website> مراجعه کنید که توضیحات کامل تری داده.

نکته قابل توجه این هست که اگر یک WAF جلوی یک وب اپلیکیشن باشه کار مانیتورینگ رو هم انجام میده و درخواست هایی که به وب اپلیکیشن زده میشه رو لاغ میکنه. مثلا اگر یک مهاجم بیاد و سعی کنه از Payload های SQL Injection استفاده کنه و روی وب اپلیکیشن تست بزنه WAF اونها رو لاغ میکنه که مثلا یک کاربر با IP Address فلان سعی کرده که یک Payload رو روی وب اپلیکیشن اجرا کنه و مثلا WAF درخواستش رو DROP کرده. پس مواظب این موارد باشیم و روی جاهایی که اجازه نداریم هر Payload رو اجرا نکنیم چرا که میان و میگیرنمون و بگا میریم.

و نکته اخر اینکه WAF ها هموطنطور که گفتم از طریق Rule ها کار میکنن و بر طبق این Rule ها Payload ها رو تشخیص میدن. به همین دلیل WAF ها راه حل صدرصدی واسه امنیت وبسایت ها نیست و ممکنه Payload های یک حفره امنیتی رو اصن توی Rule هاش نداشته باشه و همچنین ممکنه که از طریق Rule هایی بشه Payload رو Bypass کرد و خب این بستگی به WAF داره.

CDN چیست؟ Content Distribution Network یا CDN یکی از مفاهیم مهم توی حوزه وب اپلیکیشن هاست و ما باید بفهمیم که چیست و چرا و چگونه استفاده میشه. CDN مجموعه ای از سرور های توزیع شده در سطح جهان هست که محتوای یک وب اپلیکیشن رو Cache میکند و هر کاربر زمانی که میخواهد به یک وب اپلیکیشن درخواست بده به جای اینکار میدار و به اون وب اپلیکیشن که در نزدیکترین فاصله هست درخواست میده. یک CDN موجب میشه که Asset هایی مثل صفحات HTML، فایلهای CSS، فایلهای Javascript، تصاویر و ویدیو ها که برای بارگذاری صفحات وب لازم هستند با سرعت بیشتری برای کاربر ارسال شوند.

کمپانی هایی هستند که CDN ها رو کلاینت های خودشون فراهم میکنند و به اونها CDN Provider میگن و تعداد CDN ها و Provider ها و استفاده از CDN روز به روز در حال گسترش هست و سایتها هایی که کاربران زیادی دارند مثل Facebook, Netflix, Amazon و ... از CDN ها جهت کاهش ترافیک روی وب سرور های اصلی و افزایش سرعت دسترسی کاربرانشون استفاده میکنند.

فهرست زیر تعدادی از معروف ترین CDN Provider های حال حاضر رو داره میگه:

1. Cloudflare
2. Akamai
3. Fastly
4. Amazon CloudFront
5. CacheFly
6. Edgio
7. ...

دوتا از کاربرد های CDN ها رو گفتیم که اولیش Cache کردن محتوا و دومیش کاهش ترافیک روی Origin Server ها بود و یه کاربرد دیگه هم داره که اون فراهم کردن برخی موارد امنیتی برای وب اپلیکیشن هست. CDN هایی که خوب Configure شده باشند میتونن جلوی برخی حملات مثل DDoS رو بگیرن.

اما چطوری CDN ها کار میکنند؟ CDN ها شبکه ای از سرور های لینک شده به هم هستند که با هدف تحویل دادهها به کاربران با سرعت بیشتر، هزینه کمتر، اعتماد بالاتر و امنیت بیشتر وجود دارند. اگر یک وب اپلیکیشن از CDN ها جهت دسترسی کاربران به محتوا استفاده نکند همه درخواست های تمام کاربران در تمام دنیا به صورت مستقیم به Origin Server های اون وب اپلیکیشن ارسال میشه. توی تصویر زیر میبینید که چطوریه:

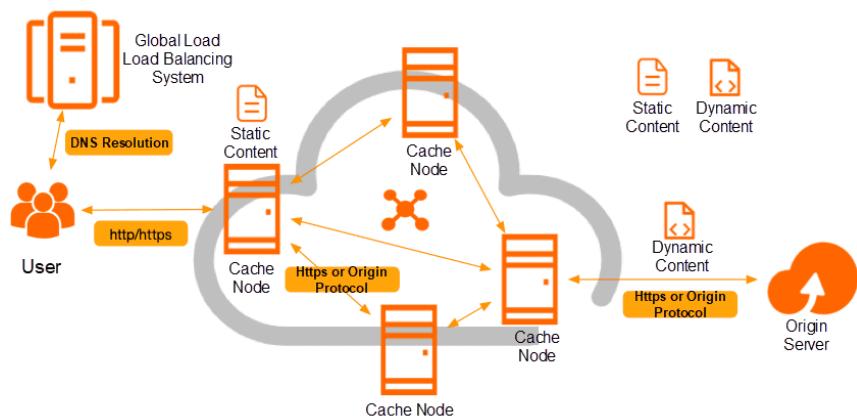


این موجب افزایش سطح ترافیک روی Origin Server های وب اپلیکیشن میشه و خوب سرعت دسترسی کاربران قاعدها با افزایش تعداد اونها به محتوا، کاهش پیدا میکنه و همچنین IP Address سرورها به صورت مستقیم در دسترس همه کاربران است و این خودش موجب میشه که امنیت وب اپلیکیشن بیشتر در خطر قرار بگیره . ولی زمانی که میان و از CDN ها استفاده میکنند، دسترسی کاربران به وب اپلیکیشن به شکل زیر میشه :



هر کاربر بسته به موقعیت جغرافیایی، به نزدیک ترین CDN از اون وب اپلیکیشن دسترسی پیدا میکنه و اون CDN حالا یا به یک NCDN نزدیک به وب اپلیکیشن و یا هم به Origin Server ها درخواست کاربر را ارسال میکنه و اینطوری میشه که ترافیک روی Origin Server کاهش پیدا میکنه و سرعت دسترسی کاربران به محتوا نیز افزایش می یابد . اما وجود CDN ها خودش موجب افزایش Surface Attack شد . یعنی بوجود امدن اونها مزیت هایی داشت ولی خوب معایبی را نیز بوجود اورد . مثلًا همین قاعده Cache شدن دادهها موجب شد که Cache Poisoning روی CDN ها انجام بشه و خوب مهاجم دادههایی را به کاربران متصل به CDN بفرسته که الوده باشه و همچنین ممکنه که NCDN بیاد و محتوایی رو Cache کنه که نباید Cache بشه و این Cache شدن موجب دسترسی افراد Unauthorized به محتوا بشه، مثلًا اگه CDN بیاد و Panel یک کاربر رو Cache کنه و این محتوا به کاربران دیگه هم ارسال بشه مشکل زا میشه دیگه، مگه نه ؟ از این نوع حملات روی CDN ها انجام میشه و اگه CDN ها امنیت خوبی نداشته باشند میتوونن موجبات نفوذ به یک وب اپلیکیشن و همچنین دسترسی به محتوا را به یک مهاجم بدن . پس با مراقبت و اگاهی بیشتر استفاده بشن بهتره .
به طور کلی HTTPS Workflow مربوط به CDN ها به شکل زیر است :

CDN - HTTPS Workflow



تصویر بالا میگه که دو نوع محتوا داریم که عبارت اند از :

1. Static Contents

2. Dynamic Contents

همون **CDN** ها همان هستند که به صورت پراکنده در سطح اینترنت در دنیا وجود دارند و به هم متصل اند و خوب برخی از اونها هم به **Origin Server** ها متصل هستند و دادهها را از اونها میخونن. کاربران هم به این **Cache Node** ها دسترسی خواهند داشت و درخواست های خودشون رو به اونها ارسال میکنند که درنهایت **Origin Server** ها از **Static Content** ها خونده میشن ولی **Dynamic Content** ها به صورت **Cache Node** شده در **Cache Node** Content به کاربران ارسال میشن.

نکته اخیر اینکه **CDN** ها یک نوع **Reverse Proxy** محسوب میشن و وقتی در جلوی **Origin Server** ها قرار میگیرند دقیقا به مانند یک **HTTP Server** عمل میکنند و همچنین **IP Address** سرور های اصلی رو از دید کاربران مخفی میکنند و کاربر فقط **IP Address** مربوط به **Cache Node** ها را رو خواهد داشت و زمانی که **DNS Resolution** رخ میده هم کاربر به **IP Address** مربوط به **Cache Node** اینکه **Origin Server** ها که جلوی **Origin Server** ها هستند این میگن و به اصطلاح میگن که **Origin Server** ها رو بست بیار و مستقیما با اونها در ارتباط باشه تا بتونه **Payload** های خودش رو بدون خر مگس معرفه روی سرور ها اجرا کنه.

بسیار خوب مفاهیم شبکه ما اینجا تومم شد و رسما میریم سمت **Reconnaissance** کردن.

نکته مهمی که جنبه حقوقی و قضایی داره اینه که زمانی که شما دارید روی یک وب اپلیکیشن **Pentest** انجام میدی باید این رو هم بدونید که اگه اون وب اپلیکیشن روی یک هاست اشتراکی قرار داشته باشه شما باید به اون هاستینگ اطلاع بدید، چرا که با **Pentest** شما در واقع تمام وب اپلیکیشن هایی که روی اون هاست اشتراکی هست هم مورد حمله محسوب میشن. اینو باید یادتون باشه. باید این موضوع رو در حین **Reconnaissance** بفهمید و ختم به خیرش کنید.

یکی از مواردی که زیاد بهش توجه نمیشه توى **Reconnaissance** اینه که **Static File** هایی وجود داره روی وب اپلیکیشن است. میتوانید از طریق اونها **MetaData** داخلشون رو بخونید و ممکن است که اطلاعاتی از **Owner** توشن باشه و یا حتی میتوانید در گاهی موارد **Static File** هایی رو پیدا کنید که اطلاعات حساسی رو فاش میکنه و حکم **Information Disclosure** داره.

توی Reconnaissance یک وب اپلیکیشن شما میتوانید با OSINT کردن اطلاعاتی از Owner دامنه رو پیدا کنید و سعی کنید اطلاعات رو از طریق OSINT بدست بیارید .

یا هم میتوانید از طریق Script های مختلفی که مینویسید سعی کنید Users Enumeration را انجام بدهید و کاربران یک سامانه رو بدست بیارید . کلا هر کاری که اطلاعاتی به شما بده، میتوانه به شما توی نفوذ کردن کمک کند و این بدست اوردن اطلاعات بستگی به سطح داشت شما از مباحثی مثل OSINT و همچنین خلاقیت شما داره .

توی ابتدای این جزو در مرور دنیا از این جزو Reconnaissance توضیحاتی دادیم و دیگه نمیخوایم با تکرار مکرات به تعداد صفحات این جزو اضافه کنیم .

انواع Scope های Reconnaissance در Bug Bounty ؟ در باگ بونتی، ما سه Scope مربوط به Reconnaissance داریم که مشخص خواهد شد و تعیین میکنن که یک Hunter در چه سطحی باید Reconnaissance انجام بده و در چه سطوحی اجازه ندارد . این نوعها به شرح زیرند :

1. زمانی میگن Small Scope ما هست که یک یا چند Domain IP Address یا SubDomain خاص رو جهت Reconnaissance مجاز کرده باشند. مثلا :

- Target.com, support.target.com, api.target.com
- 10.0.2.10, 10.0.2.11, 10.0.0.15

2. زمانی که یک Domain و Subdomain های اون و یا IP Address خاصی با اون (CIDR) رو جهت Reconnaissance و تست نفوذ مجاز کرده باشند میگن که Medium Scope است. در این نوع Scope مهاجم باید بتوانه که تمام Subdomain های موجود را کشف کنه که بهش Subdomain Discovery میگن و ابزارهایی هم واسش هست. در برخی موارد حتی کمپانی که Bug Bounty داده خودش هم اطلاع نداره که واقعاً چندتا Subdomain داره و هرچه شما بیشتر بتونی پیدا کنی احتمال پیدا کردن باگ هم بیشتر چون گاهی اوقات برخی Subdomain های زمانی بوجود اومدن و اسیب پذیرند و وجود دارند و حذف نشده اند.

3. زمانی هم که یک CIDR داده باشن میتوانید از طریق NMAP اقدام به کشف Host هایی کنید که Live هستند و اونها رو تست کنید . مثلاً موارد زیر جزو Medium Scope محسوب میشوند :

- *.target.com
- 192.168.1.1/24

4. زمانی هم هست که یک سازمان میاد و اعلام میکنه تمام وبسایت هایی که به نام من هست، تماماً جزو باگ بونتی هست و میتوانید روی تمام اونها تست نفوذ انجام بدهید و یا هم میاد و یک CIDR با رینج نسبتاً زیادی رو قرار میده که باید همه Host های Live اون رو بدست بیارید و میتوانید روی همشون تست نفوذ انجام بدهید . مثلا :

- All related websites to the company
- 192.168.0.1/16

5. مهم هست که Scope رو تشخیص بدم و شروع نکنیم به Recon کردن مواردی که خارج از Scope هستند چرا که هیچ باگ بونتی به شما تعلق نخواهد گرفت حتی زمانی که باگ مهمی رو کشف کرده باشید .

اطلاعاتی که مهم هست در حین Reconnaissance به اونها دست پیدا کنیم به شرح زیرند :
1. Discover Real IP Addresses : اینکه شما باید تمام تلاشتون رو بکنید و IP Address های واقعی (نه پشت Origin) Server های مربوط به سازمان رو بدست بیارید . پیدا کردن این IP Address ها موجب میشه که دست ما برای کشف اسیب پذیری ها باز تر باشه و ما بتونیم اسیب پذیری های بیشتری رو کشف کنیم .

Enumerate IP Range .2 : این مورد ممکن است که بیشتر توی شبکه ها وجود داشته باشه ولی خب گاهی هم توی باگ بونتی های وب اپلیکیشن ها میتوانه اطلاعاتی رو به ما بده . باید Alive IP های IP Address داخل یک Range را Enumerate کنیم . میتوانیم از طریق NMAP و ابزار های مشابه این کار رو انجام بدیم .

اما این کار چه مزیتی میتوانه واسه ما داشته باشه ؟ خب ما میتوانیم بیایم و IP Address Range یک سازمان که قراره Pentest یا Bug Bounty کنیم رو بدست بیاریم و میدونیم که IP Address هاست و ما دسترسی نداریم بهش . پس میایم این رینج IP Address مربوط به سازمان رو بررسی میکنیم و شاید IP Address وеб اپلیکیشن تارگت ما یکی از همین IP Address ها باشه که پشت ابر قرار داره . این یکی از مزایایی هست که این کار داره .

Enumerate Public Servers .3

Enumerate Public Databases .4 : منظور همون Database هایی هست که NoSQL هستند و میشه بدون مجوز و به صورت Unauthenticated بهش متصل شدن و اطلاعات رو Fetch کرد . خیلی از اطلاعاتی که Leak شدن از طریق همین دیتابیس های NoSQL بدون نیاز به احرار هویت بودند که مهاجمین توئنستن اطلاعات Extract کنن . در انتهای جزوی بیشتر درمودشون حرف زدیم .

Enumerate Services .5 : در انتهای جزوی توضیح داده شده اند .

Enumerate Sensitive Data Leakages .6 : در اینکار ما میایم و سعی میکنیم عموم Static File ها رو بدست بیاریم و ممکن هست که جایی به یک فایل برخورد کنیم که اطلاعاتی حیاتی توش باشه و مثلًا Credential Extract یا Excel پیدا کنیم . از این موارد بوده قبله .

خب ما تا اینجا از دو اصطلاح استفاده کردیم که بهتره تفاوت های اونها رو بدونیم چرا که متفاوت هستند . **Bug** و **Bounty** این دو تا اصطلاح تفاوت هایی دارند .

1. توی Bug Bounty شما نیاز هست که یک اثر از یک اسیب پذیری رو نشون بدی و اگه نشه چنین کاری کرد یعنی یک اسیب پذیری رو پیدا کرد ولی نشه اون رو Exploit کردن به هیچ عنوان قابل قبول نیست . این مورد توی Policy های HackerOne گفته شده . مثلًا شما یک تارگت دارید که داره از یک نسخه اسیب پذیر Jquery استفاده میکنه، اما شما نمیتوانید از اسیب پذیری اون نسخه بهره کشی کنید و اون رو Exploit کنید چون اصن توی کد تارگت هیچ جایی از اون قسمت اسیب پذیر استفاده نشده و یا استفاده شده ولی خب قابل اکسپلوبیت کردن نیست و یا شما نمیتوانی پس بانتی نمیگیری .

2. در Bug Bounty شما موظف نیستید که همه اسیب پذیری های موجود در یک تارگت رو بدست بیارید و فقط به از ای اسیب پذیری هایی که توئنستید Exploit کنید به شما هزینه ای پرداخت خواهد شد .

3. در Penetration Testing شما باید هر چیزی رو گزارش کنید . ورژن های قدیمی رو توی گزارش ذکر کنی، اسیب پذیری های ورژن های قدیمی رو ذکر کنی، وجود یک اسیب پذیری رو ذکر کنی، عدم وجود اسیب پذیری رو هم باقیستی ذکر کنی و نسبت به Bug Bounty سخت گیری هایی وجود داره و در صورتی که شما چندین اسیب پذیری که وجود داره رو نتوانید کشف کنید و گزارش بدهید برای شما میتوانه مشکلات قضایی ایجاد کنه .

4. در Penetration Testing شما باید اسیب پذیری هایی که پیدا میکنید رو ارزیابی کنید یعنی Vulnerability Assessment را انجام بدهید و شدت یک اسیب پذیری و مشکلاتی رو که میتوانه پدید بیاره در گزارش خود بنویسید .

- نکته ای که هم در Post Exploitation و هم در Bug Bounty نیاز است که رعایت شود این است که Proof Of Concept POC یا استخراج کرد .

Online OSINT Service هایی وجود دارد که جهت جمع اوری اطلاعات می توئنیم از شون استفاده کنیم که حس کردم نیازه لیستی از اونها رو داشته باشیم :

1. Google Dorks (<https://google.com>)

گوگل دورکها که بهشون Google Hacking هم میگن عباراتی هستند که میتوانیم از شون استفاده کنیم تا جستجوی دقیق تری داشته باشیم . از طریق این عبارات میشه جستجو رو فقط به یک سایت معطوف کرد و همچنین میتوانید به صورت ترکیبی از انها استفاده کنید . از کاربرد های اونا میشه به جستجوی یک صفحه خاص در یک وبسایت خاص اشاره کرد و سایت <https://www.exploit-db.com> هم یک دیتابیس از اونها داره که میتوانید از ادرس- <https://exploit-db.com> اونها رو ببینید . لیست زیر صفحه اول این دیتابیس است :

<https://shodan.io> (Also there is a beta <https://beta.shodan.io>) .2

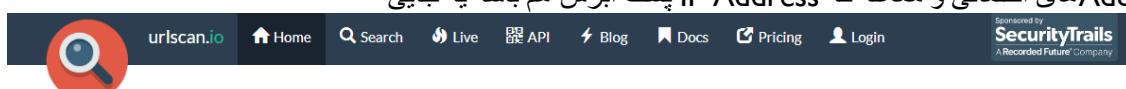
شدن یک موتور جستجو است که میشه از طریق انواع مختلف سرور های متصل به اینترنت رو پیدا کرد . مثلا از طریق shodan میتوانید یک ادرس IP خاص توی یک کشور خاص رو جستجو کنید و اطلاعاتی از اون سرور مثل سرویس های فعال و پورتها باز و اطلاعاتی درمورد سرویس ها بدست بیارید . البته نسخه رایگان Shodan محدودیت دارد و نسخه پولی اون هم 59 دلار در ماه است و البته در یک روز خاص از سال که الان نمیدونم چه روزیه میشه با اشتراک اون رو خرید .



: این وب سایت یک URL میگیره و سعی میکنه اطلاعاتی از اون وبسایت به ما بده، مثلا تکنولوژی

های استفاده شده و همچنین فایل های Static داخل وبسایت و درخواست هایی که در اون Request ارسال شده .

های احتمالی و ممکنه که IP Address پشت ابرش هم باشه یه جایی



urlscan.io

A sandbox for the web

URL to scan

▶ Public Scan

Options

Recent scans

Updates every 10s - Last update: 13:40:51

URL

[otvetkino.ru/user/woupSpoubtets/](#)

Age

7 seconds

Size

5 MB

IPs

413

10



[toycarz4kidz.com/about/](#)

9 seconds

2 MB

101

5



[www.expiredwixdomain.com/?redirectedFor=josebadaro.com](#)

10 seconds

837 KB

79

5



[studynotes365.com/](#)

11 seconds

6 MB

839

20



[bankmuscat.workplace.com/work/landing/input/?next=https%3A%2F%2Fbankmuscat.work...](#)

12 seconds

472 KB

48

3



[www.frontendr.com/frontendr.php](#)

44 seconds

2 MB

102

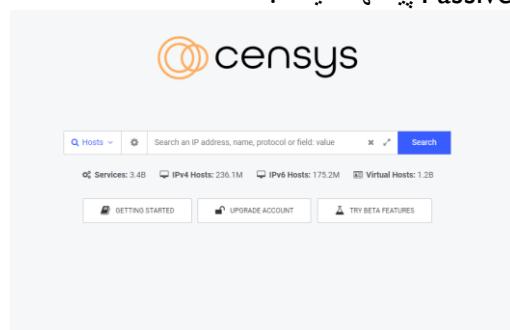
22



: این هم یکی از سایت های خوب جهت پیدا کردن IP Address IP پشت ابر هست . اطلاعات نسبتا

زیادی از وب سرور رو به ما میده و همچنین میتوانیم یک IP Address Range IP خاص رو هم بررسی کنیم و هاست های

اون رو پیدا کنیم و همچنین سرویس های در حال اجرا رو اونها رو ببینیم . نسبتا سایت کاملیه و استفاده ازش به شدت توانی رو پیشنهاد Passive Reconnaissance به صورت .



: یکی از سرویس های خوب جهت دسترسی به اطلاعات Leak شده می باشد . میتوانید یک Domain, CIDR, IP Address, ... را توش جستجو کنید و ببینید که چه اطلاعاتی Leak شده . البته این سرویس پولی

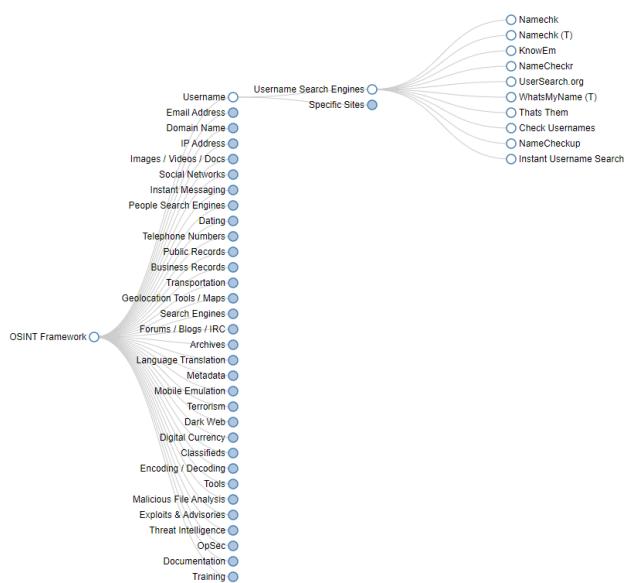
هست و خب تا يه جاهایی ممکنه اطلاعاتی رو به شما بده و همچنین میتوانید ثبت نام کنید و 50 تا Look Up در روز داشته باشید.

Search engine & data archive.

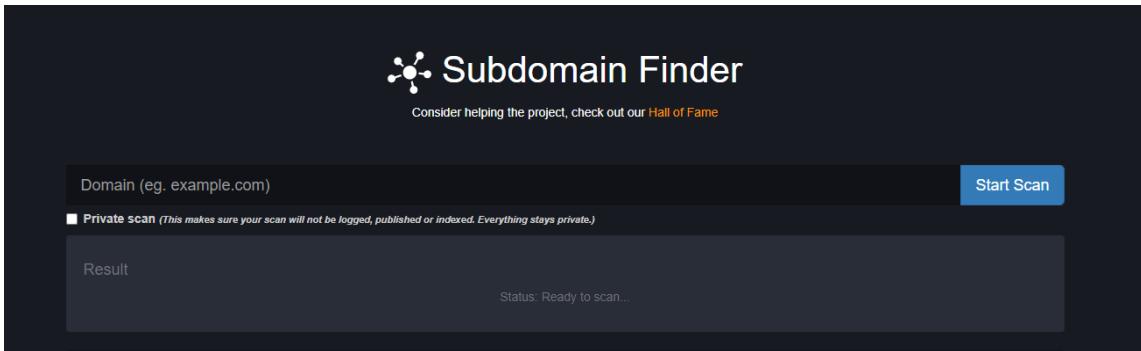


یکی دیگه از وب سایت هایی هست که میتوانیم واسه OSINT ازش استفاده کنیم. این وبسایت، وبسایت ها و ابزار هایی که میتوانه توی OSINT کردن به شما کمک کنه رو دسته بندی کرده و بهتون نشون میده. مثلًا اگه بخوايد یک Username خاص رو OSINT کنید میتوانید توی دسته بندی ها قسمت Username سایتها مربوط به این کار رو ببینید و استفاده کنید.

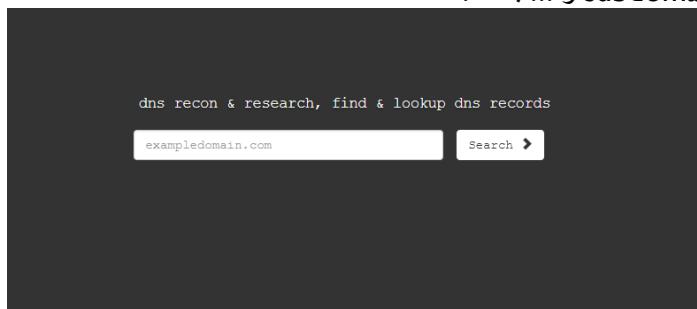
OSINT Framework



این وبسایت به شما توی پیدا کردن Subdomain های یک تارگت کمک میکنه . البته ابزار هایی هم هستند که با پایتون یا Go نوشته شدن، میتوانن توی تکمیل کردن این فرایند بهتون کمک کنن ولی خب به صورت Active Reconnaissance این کار رو میکنن ولی این وبسایت به صورت Passive Reconnaissance برآتون این کار رو انجام میده . همچنین این وبسایت بهتون میگه که ایا اون تارگت شما از Cloudflare استفاده میکنه یا نه و همچنین ادرس IP های Subdomain ها رو هم برآتون درصورتی که بتونه پیدا کنه مینویسه . اما بهتر است که اگه میخوايد Subdomain های یک تارگت رو پیدا کنید از چند ابزار جهت اینکار استفاده کنید و نتیجه شامل اجتماع خروجی همه ابزارها باشه تا نتیجه بهتری بگیرید .



dns recon & research, find & lookup dns : این وبسایت توانی عنوانش زده <https://dnsdumpster.com> .8
یعنی اینکه میاد و درمورد DNS یک تارگت، Reconnaissance میکنه و این یعنی میاد DNS Record اون تارگت رو پیدا میکنه. در نهایت خروجی رو هم میتوانید به شکل گراف ببینید و هم به صورت عادی و جدول . همچنین Subdomain ها رو هم پیدا میکنه و سعی میکنه IP Address های اونها رو هم پیدا کنه . میتونه تکمیل کننده ابزار های دیگه مثل subdomainfinder.c99.nl و ... باشه .



dns recon & research, find & lookup dns records : این هم یک ادرس Domain یا IP از شما میگیره و اطلاعاتی از اون رو برآتون پیدا میکنه .9
متلا Whois دامنه رو نشون میده، اطلاعات مربوط به DNS Record ها رو، میتوانه Ping, Traceroute همچنین اگه IP Address بدید برآتون Nslookup میکنه و اطلاعات مربوط به سرویس های فعال روی سرور رو هم نشون میده .

Free online network tools

Tools

Domain Dossier

Investigate domains and IP addresses. Get registrant information, DNS records, and more—all in one report.

enter a domain or IP address

or [learn about yourself](#)

Domain Check

See if a domain is available for registration.

Email Dossier

Validate and troubleshoot email addresses.

Browser Mirror

See what your browser reveals about you.

Ping

See if a host is reachable.

Traceroute

Trace the network path from this server to another.

NsLookup

Look up various domain resource records with this version of the classic NsLookup utility.

AutoWhois

Get Whois records automatically for domains worldwide.

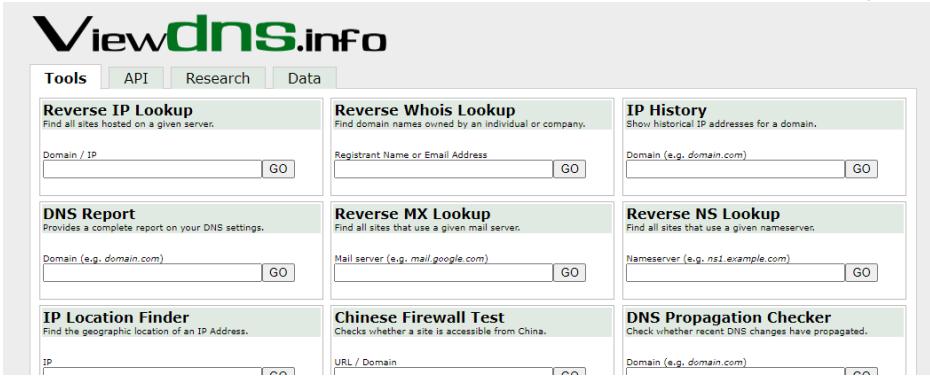
AnalyzePath

Do a simple, graphical traceroute.

IP History, Reverse IP : این وبسایت امکانات خیلی زیادی رو در اختیار ما میدارد . برآمده <https://viewdns.info> . 10

Lookup, Reverse NS Lookup, Get HTTP Headers, DNS Record Lookup, Port Scanner, Ping, Traceroute, Is My Site Down و ... رو انجام میده و یکی از کاربرد های این وبسایت استفاده از IP History است که به ما این امکان رو میدهد که بتوانیم IP Address پشت ابر یک وب اپلیکیشن رو پیدا کنیم . تاریخچه ای رو به ما میده و لیستی از IP Address ها که در اون تاریخها، وب اپلیکیشن تارگت ما داشته و ممکنه یکی از اونا IP Address پشت ابر باشد .

همچنین میشه از Is My Site Down استفاده کرد و بررسی کرد که ایا وب اپلیکیشن Down هست یا صرفاً ما رو بلاک کرده و همچنین میتوانید از گزینه های Chinese Firewall Test, Iran Firewall Test تارگت از فایروالهای چینی و ایرانی گزرنمیکن و در اونجاها در دسترس هست یا نه .



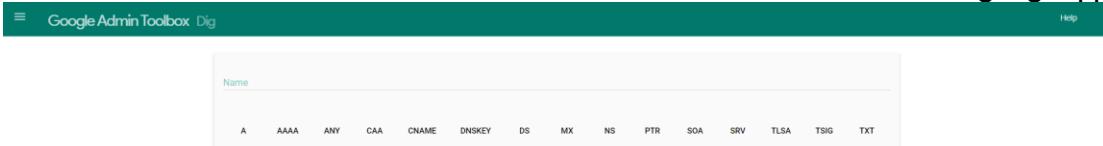
نکته ای که لازمه بدونید اینه که وقتی ما IP Address پشت ابر رو پیدا کردیم میتوانیم بریم توی ادرس IP که یک فایل به نام hosts وجود داره و اونجا دامنه تارگت رو تنظیم کنیم به IP Address پشت ابر که هر وقت دامنه رو باز کردیم سریعاً به CDN پشت ابر درخواست ارسال بشه نه به Provider . مثلاً یه خطی به شکل زیر اضافه میکنیم بهش :

یا هم میتوانید از BurpSuite این تنظیمات رو انجام بدهید، که وقتی میخواهد دامنه رو اجرا کنه به جای ارسال درخواست به IP ادرس CDN Provider، درخواست مستقیماً به IP Address پشت ابر ارسال کنه . جهت انجام این تنظیمات وارد تب Project options بشید و قسمت Hostname Resolution Add کلیک کنید و یک رکورد رو اضافه کنید که شامل IP Address پشت ابر و Hostname Address تارگت خواهد بود .

وقتی یک تارگت پشت ابر باشه، در Request هر Response هست، مولفه هایی وجود داره که نشون دهنده اون هست مثلاً در زمانی که یک تارگت پشت ArvanCloud هست، مولفه هایی در Response وجود داره که با ar- شروع میشوند یا هم توی Response Header ها مولفه Server مقدار ArvanCloud رو داره . بعد از اینکه IP Address پشت ابر رو جهت ارسال درخواست ها تعیین میکنیم، این مولفه ها حذف میشوند یا تغییر مقدار میدند.

دسترسی مستقیم به IP Address پشت ابر موجب میشه WAF موجود در CDN که موجب میشن برخی از Payload ها کار نکنه دیگه وجود نداشته باشن و از آرمون ندن .

11 : این ابزار کوچیک شده ابزار dig توی لینوکس هست که برای ما DNS Record ها رو نشون میده . ابزار dig رو بعداً شاید بررسی کردم و جزوی کردم ولی فعلاً میتوانید از همین استفاده کنید . A, AAAA, CNAME, TXT, NS و ... رو بر میگردونه . <https://toolbox.googleapps.com/apps/dig>



12 : از این ابزار هم میتوانید گاهی اوقات استفاده کنید و شاید بتوانید از طریقش IP Address پشت ابر رو پیدا کنید . بر اساس Certification نتایج رو نشون میده . یعنی میاد و بررسی میکنه که چه سرور هایی وجود داره که SSL Certification اونها با SSL Certification تارگت ما یکی هست و اونها رو نشون میده .

13 : این ابزار برآتون مجموعه ای از امکانات رو فراهم میکنه من جمله WAF Detection, Nmap, Subdomain finder, Ping Tool, Cipherscan, IP ASN Lookup, IP Geo Location <https://nmapmapper.com>

<https://lookup.icann.org> . 14

15 DNS Record های تارگت ما رو نشون میده و میتوانیم ازش برای پیدا کردن Real IP Address پشت ابر استفاده کنیم .

تمام ابزار هایی که در فهرست بالا گفته شدند همگی جهت Passive Reconnaissance استفاده خواهند شد و درواقع با استفاده از آنها مانع تعامل مستقیمی با وب اپلیکیشن تارگت نخواهیم داشت .

ابزار های Binary جمع اوری اطلاعات ؟ ما علاوه بر اون سرویس های انلاین که جهت جمع اوری اطلاعات بهمن ممکن میکردن گزینه های باینری هم داریم، یعنی ابزار هایی که روی سیستم عامل ما اجرا میشوند و میتوانیم از طریقشون اطلاعاتی از یک تارگت بدست بیاریم . تعداد این ابزار های بسیار زیاد است ولی اینجا ما فقط سعی میکنیم برخی از آنها را بیان کنیم . فهرست زیر مجموعه ای از این ابزار ها را نشون میده و توضیحاتی درباره آنها دادیم :

1. (Linux/Unix) dig command : دستور dig یکی از دستوراتی هست که در ترمینال لینوکس به صورت پیشفرض وجود دارد و توی صفحه توضیحات این ابزار نوشته که این ابزار جهت پرس و جو کردن از DNS name server ها می باشد . dig یک lookup انجام میده یعنی میاد و یک Name Server به DNS Query میکنیم ارسال میکنه و در نهایت پاسخ را برای ما نشون میده . dig یک ابزار تحت ترمینال هست و حالت GUI نداره و قاعدها دارای مجموعه ای از Option هاست که میتوانید آنها را استفاده کنیم . اگه ما یک دستور رو وارد کنیم ولی مشخص نکنیم که از چه DNS Name Server پرس و جو کنیم به صورت پیشفرض میره توی /etc/resolv.conf و از داخل این فایل DNS Name Server های پیشفرض رو انتخاب میکنه و استفاده میکنه . فایل resolv.conf توی لینوکس مشخص کننده Name Server های پیشفرض سیستم عامل هست .



```
kali㉿kali:~$ dig -h
Usage: dig [global-server] [domain] [q-type] [q-class] [q-opt]
          {global-d-opt} host {@local-server} {local-d-opt}
          [ host {@local-server} {local-d-opt} [...]]
Where: domain   is in the Domain Name System
        q-class  is one of (in,hs,ch,...) [default: in]
        q-type   is one of (a,any,mx,ns,soa,hinfo,axfr,txt,...) [default:a]
                  (Use ixfr=version for type ixfr)
        q-opt    is one of:
                  -4           (use IPv4 query transport only)
                  -6           (use IPv6 query transport only)
```

مثالاًگه بخواهیم یک DNS Query بزنم به google.com های IP Address ها و ازشون Name Server رو بپرسم کافیه که به شکل زیر عمل کنم :



```
kali㉿kali:~$ dig google.com
; <>> DiG 9.19.17-1-Debian <>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 19534
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; ANSWER SECTION:
;google.com. IN A 216.239.38.120
;; AUTHORITY SECTION:
;; ADDITIONAL SECTION:
;; SERVER: 4.2.2.4#53(4.2.2.4) (UDP)
;; WHEN: Tue Nov 28 07:06:20 EST 2023
;; MSG SIZE rcvd: 44
```

میتوانید از طریق dig +short به dig بگید که فقط ادرس IP های برگشته را بهم نشون بده و نمیخواهیم جزئیات پاسخ را ببینم :

```
kali㉿kali:~$ dig google.com +short
216.239.38.120
```

dig به صورت پیش فرض از /etc/resolv.conf هست جهت ارسال DNS Query و گرفتن پاسخ استفاده میکنه، ولی ما هم میتوانید از طریق @x.y.w.z یک Name Server خاص رو بهش بدم و بگیم که از این Name Server جهت پرس و جو استفاده کن :

```
kali@kali:~$ dig cloudflare.com @4.2.2.4 +short
104.16.132.229
104.16.133.229
```

توی پاسخ هایی که یک DNS Query بر میگردونه اطلاعاتی با انواع مختلف وجود داره که بهشون میگن Record، مثل Record A، AAAA، CNAME، ... فقط dig بخواهد که فقط جواب رو در قالب یکی از این Record ها برگردونه مثل NS Record ها رو نشون بده و کافیه که نوع پاسخی که میخواهد رو جلوی dig بنویسید . مثلا در مثال زیر من نوشتم که یعنی فقط NS Record ها رو بده :

```
kali@kali:~$ dig NS cloudflare.com
; <>> DiG 9.19.17-1-Debian <>> NS cloudflare.com
;; global options: +cmd
;; Got answer:
;; ->>>HEADER<<- opcode: QUERY, status: NOERROR, id: 52501
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 15

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;cloudflare.com.           IN      NS
;; ANSWER SECTION:
cloudflare.com.        12891   IN      NS      ns6.cloudflare.com.
cloudflare.com.        12891   IN      NS      ns7.cloudflare.com.
cloudflare.com.        12891   IN      NS      ns3.cloudflare.com.
cloudflare.com.        12891   IN      NS      ns4.cloudflare.com.
cloudflare.com.        12891   IN      NS      ns5.cloudflare.com.

;; ADDITIONAL SECTION:
ns3.cloudflare.com.    79920   IN      A       162.159.7.226
ns3.cloudflare.com.    79920   IN      A       162.159.0.33
ns3.cloudflare.com.    85719   IN      AAAA    2400:cb00:2049:1::a29f:21
ns3.cloudflare.com.    85719   IN      AAAA    2400:cb00:2049:1::a29f:7e2
ns4.cloudflare.com.    86025   IN      A       162.159.8.55
ns4.cloudflare.com.    86025   IN      A       162.159.1.33
ns4.cloudflare.com.    81688   IN      AAAA    2400:cb00:2049:1::a29f:837
ns4.cloudflare.com.    81688   IN      AAAA    2400:cb00:2049:1::a29f:121
ns5.cloudflare.com.    84516   IN      A       162.159.2.9
ns5.cloudflare.com.    84516   IN      A       162.159.9.55
ns5.cloudflare.com.    85469   IN      AAAA    2400:cb00:2049:1::a29f:937
ns5.cloudflare.com.    85469   IN      AAAA    2400:cb00:2049:1::a29f:209
ns6.cloudflare.com.    84516   IN      A       162.159.5.6
ns6.cloudflare.com.    84516   IN      A       162.159.3.11

;; Query time: 159 msec
;; SERVER: 4.2.2.4#53(4.2.2.4) (UDP)
;; WHEN: Tue Nov 28 07:11:12 EST 2023
;; MSG SIZE  rcvd: 429
```

whois command (Linux/Unix/Win32) : این دستور هم توی لینوکس وجود داره و هم توی یونیکس ها و ویندوز ها . وقتی یک دامنه ثبت میشه اطلاعات Owner اون دامنه توسط یک دیتابیس جهانی ذخیره میشه و زمانی که شما بباید و از اون دامنه whois بگیرید میتوانید به اطلاعات Owner دسترسی پیدا کنید، ولی در برخی اوقات Owner میتوانه اطلاعات خودش رو مخفی کنه . در واقع whois یک Internet Protocol هست که این کار رو انجام میده . اطلاعاتی که میتوانید از طریق این ابزار بدست بیارید، نام و نام خانوادگی، ایمیل، شماره تماس، ادرس و ... Owner هست و همچنین اطلاعاتی هم از خود Domain مثل Name Server ها، تاریخ انقضا، تاریخ ثبت و ...

```
kali㉿kali:~$ whois google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851750
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/

```

3. nslookup command (Linux/Unix/Win32) : این ابزار هم توی لینوکس، یونیکس و هم ویندوز وجود دارد . میتوانید از شاستفاده کنید که ادرس IP متناظر با یک Domain را بدست بیارید و در واقع این ابزار یک DNS Query به Name Server ها ارسال میکنه و جواب رو به شما نشون میده :



میتوانید به شکل زیر هم ازش استفاده کنید :



4. nmap : ابزار بسیار قدرتمند nmap رو دیگه تقریبا همه میشناسن. این ابزار میتونه برای شما خیلی خیلی از کارا رو انجام بده . مثلًا میتوانه به سادگی با ارسال پکت هایی پورتهای باز یک تارگت رو نشون بده و همچنین میتوانه اطلاعاتی از اون پورت های باز رو جمع اوری کنه . این ابزار سوئیچ های بسیار زیادی داره و خودش اندازه یک کتاب زمان میبره که بخوایم توضیح بدیم در مروره همهشون . به صورت پیش فرض روی کالی لینوکس نصب هست و اگه هم نبود کافیه که از طریق apt اون رو به سادگی نصب کنید و میتوانید nmap رو توی ویندوز و سیستم عاملهای دیگه هم نصب کنید و تقریبا همه جا هست . علاوه بر چک کردن پورتهای باز از طریق چندین متد مختلف که پشتیبانی میکنه، این ابزار از طریق Script هایی که داره هم به شما کمک میکنه اسیب پذیری هایی رو از تارگت هایی کشف کنید و در برخی از Script ها این امکان رو دارید که حملاتی رو انجام بدهید . خب هرچی از قابلیت های این ابزار بگم بازم کم گفتم و بهتره که به Documentation های رسمی اون مراجعه کنید و به نظر من که قویتری Scanner شبکه های کامپیوتری همین ایشون هستن و هر کی بلد نیس نصف عمرش بر فناست .



دقشود که nmap به صورت Active عمل Reconnaissance را انجام میده و قطعا وقتی شما یک تارگت رو Scan میکنید لاگهایی از با IP Address شما در تارگت انداخته میشه و برای اینکه بتوانید خودتون رو مخفی کنید میتوانید از پراکسی ها، VPN ها و ... استفاده کنید . ابزار هایی مثل proxychains توی لینوکس میتوانه عمل پراکسی کردن پکت ها رو برآتون انجام بده و توی ویندوز هم proxyfire همین کار رو میکنه .

.5 باشد . این ابزار سریعتر از nmap عمل میکنه و توسط زبان برنامه نویسی Rust توسعه داده شده است . میتوانید از لینکی که گذاشتم کدهای اون رو ببینید و اون رو دانلود کنید .

```
RustScan on master [?] is ⚡vrustscan:1.10.0 via R v1.46.0
→ rustscan google.com --no-nmap --ulimit 10000
[0] { [ ] } { [ ] } { [ ] } { [ ] } { [ ] } { [ ] } { [ ] } { [ ] } { [ ] }
Faster Nmap scanning with Rust.

: https://discord.gg/GFrQsGy :
: https://github.com/RustScan/RustScan :
@https://admin.tryhackme.com

[+] The config file is expected to be at "/home/bee/.rustscan.toml"
[-] Automatically increasing ulimit value to 10000.
Open 216.58.204.14:80
```

قاعدها نیومدیم اینجا که تمام امکانات اسکنر ها رو بررسی کنیم ولی اگه خواستید درمورد این اسکنر اطلاعات کاملی داشته باشید میتوانید به صفحه [Wiki](https://github.com/RustScan/RustScan/wiki) اون به ادرس <https://github.com/RustScan/RustScan/wiki> توری گیت هاب مراجعه کنید . اینو هم بگم که این اسکنر از nmap هم استفاده میکنه و میتوانید هم بهش بگید نکنه .

.6 Bug Hunter : این ابزار توسط تامنانام معروف که یکی از بزرگترین <https://github.com/tomnomnom/waybackurls> های فعلی هست با زبان Go نوشته شده (ایشون ابزار زیاد نوشته که تقریبا خیلی استفاده میکند پیشنهاد میشه که بربد توری گیتها بشیش یه نگاهی بندازیید) . waybackurls میره از توری <https://web.archive.org> که ارشیو کلی اینترنت هست و یکی از بزرگترین دیتابیس های حال حاضر محسوب میشه ، URL های قدیمی یک وبسایت رو میکشه بیرون . WayBack Machine میگیره و تغیرات اون وبسایت SnapShot میگیره و تغیرات اون وبسایت نسبت به دفعه قبلی رو نشون میده . گاهی اوقات میتوانید ازش استفاده کنید تا URL های قدیمی که الان دیگه توری وبسایت نیست و قبلابوده و هنوز فعال هستن رو پیدا کنید و ممکن است توری اون URL بتوانید اسیب پذیری کشف کنید .

```
File Actions Edit View Help
(kali㉿kali)-[~]
$ waybackurls -h
Usage of waybackurls:
-dates
    show date of fetch in the first column
-get-versions
    list URLs for crawled versions of input URL(s)
-no-subs
    don't include subdomains of the target domain
(kali㉿kali)-[~]
```

اگه بخواید هم ازش استفاده کنید کافیه که Domain رو بهش بدد :

```
(kali㉿kali)-[~]
$ waybackurls tesla.com
https://www.tesla.com/
https://www.tesla.com/%20%20
https://www.tesla.com/?utm_source=Instagram&utm_medium=social
https://www.tesla.com/?utm_campaign=NA_HQ_fireBlog_NC01&utm_medium=email&utm_source=Eloqua&elq=15489ba23f184578b5a89c500d983a12&elqCampaignId=497
https://www.tesla.com/?redirect=no
https://www.tesla.com/about
https://www.tesla.com/about/blog/11-questions-odyssey-pioneers-driver-luke-mcclellan?redirect=no
https://www.tesla.com/about/executives
https://www.tesla.com/about/executives/elonmusk
https://www.tesla.com/about/legal
https://www.tesla.com/about/legal?utm_campaign=NA_HQ_fireBlog_NC01
https://www.tesla.com/about/press
https://www.tesla.com/about/press/press-mentions?page=12&redirect=no
https://www.tesla.com/about/press/press-mentions?page=24
https://www.tesla.com/about/press/releases/clone-tesla-motors-model-s-makes-its-asian-debut-hong-kong?page=3&redirect=no
https://www.tesla.com/about/press/releases/elon-musk-named-innovator-year-technology-wsj-magazine
https://www.tesla.com/about/press/releases/HongKong%40teslamotors.com?page=1
https://www.tesla.com/about/press/releases/HongKong%40teslamotors.com?page=6
https://www.tesla.com/about/press/releases/panasonic-enters-supply-agreement-tesla-motors-supply-automotivegrade-battery-c?redirect=no
https://www.tesla.com/about/press/releases/panasonic-presents-first-electric-vehicle-battery-tesla?redirect=no
```

.7 : این ابزار یک لیست از URL هارو میگیره و یک Request بپشنون میزنه و برگشتی رو نشون میده . مثلا زمانی که شما یک لیست از URL هارو دارید و میخواید ببینید کدومشون 200 بر میگردونه و کدومشون 404 و ... راحت میتوانید از این ابزار که با زبان برنامه نویسی Go نوشته شده استفاده کنید . این پارو که این ابزار رو نوشته هم اگه دوست داشتید دنبال کنید ، hakluke چندین ابزار خوب نوشته مثل hakrawler که واسه پیدا کردن لینکهای داخل یک سایت استفاده میشه و خیلی کاربردیه .

.8 : این ابزار با استفاده از PowerShell نوشته شده و کارش <https://github.com/dafthack/PowerMeta> هست و فایل‌های یک تارگت که به صورت عمومی وجود دارند رو توی گوگل و بینگ با استفاده از دورکها و ...

جستجو میکن و متادیتا های او نها رو بیرون میکشه که میتوانه منجر به پیدا شدن Username ها، دامنه ها، اسم نرم افزارها و نام کامپیوتر های اون تارگت بشه .

.9 : این ابزار هم جهت استخراج MetaData فایلها استفاده میشه . یه کم کار باهاش سخنه چون هم میگردد انلاین یه کارابی رو انجام میده مثل ماتادیتا ها رو جستجو میکنه و ... و هم نیاز به SQL Server داره و فقط هم تحت ویندوز هست . نصبش یه کم دردرس سازه ولی میشه باهاش کنار اومد . ابزار قدرتمندیه .

.10 : این ابزار مثل اینکه خیلی معروفه توی گیت هاب و واسه Web path یعنی Fuzzing discovery استفاده میشه . حالا اینکه Fuzzing چیه رو بعدا بحث میکنیم . توسط پایتون نوشته شده و مت اینکه قویه

.11 : این ابزار توسط زبان برنامه نویسی Go نوشته شده و کارش Fuzzing هست مثل dirsearch . خب این سtarه های زیادی توی گیت هاب داره و مت اینکه قویه . حالا اینکه چطوری از این برنامه استفاده کنیم کافیه که از گیت هاب اونو دانلود کنید و نسخه های ویندوزی و لینوکسی و ... داره و روی کالی لینوکس هم به صورت پیش فرض نصبه . باید به این ابزار یک Wordlist شامل تمامی کلماتی که میخواهد تست کنید بدید، یک URL و هرجایی که خواستید کلمات داخل Fuzzing را بگیره کلمه FUZZ رو مینویسید .

```

(kali㉿kali)-[~/tmp]
$ ffuf -w wordlist.txt -u https://meggieschneider.com/FUZZ View Help
[+] [FFUF v2.1.0-dev] [Parallel Threads: 40] [Running: 10/10] [Total: 10] [Duration: 0:00:01] [Errors: 0]
[+] [Status: 301, Size: 242, Words: 14, Lines: 8, Duration: 479ms]
[+] [URL | https://meggieschneider.com/admin]
[+] [-> | https://meggieschneider.com/admin/]
[+] [* FUZZ: admin]

:: Method      : GET
:: URL         : https://meggieschneider.com/FUZZ
:: Wordlist    : FUZZ: /home/kali/tmp/wordlist.txt
:: Follow redirects: false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500

admin
[Status: 301, Size: 242, Words: 14, Lines: 8, Duration: 479ms]
.. Progress: [10/10] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:01] :: Errors: 0 ::


```

خب میبینید که admin رو که توی wordlist.txt بود با Status Code 301 نشون داد و بقیه که نشون نداد یعنی 404 دادن . میتونید -v رو هم بزنید که بیشتر در مورد چیزهایی که پیدا میکنه اطلاعات بد . مثلا توی مورد زیر گفت که 301 داده و ریدایرکت شده به یه جایی دیگه :

```

[Status: 301, Size: 242, Words: 14, Lines: 8, Duration: 346ms]
| URL | https://meggieschneider.com/admin
| --> | https://meggieschneider.com/admin/
| * FUZZ: admin

Method      : GET
URL         : https://meggieschneider.com/FUZZ
Wordlist    : FUZZ: /home/kali/tmp/wordlist.txt

```

یکی از نقاط قوت این ابزار اینه که سرعت خیلی زیادی داره و این به نظرم خودش کافیه که بیشتر با این ابزار کار کنیم . ولی خب در کل توی ابزارها پیشنهاد اینه که با هر کوم راحت ترین کار کنین، زیاد درگیر نشین چون اینقدر ابزارها زیادن که همشون رو نمیشه تست کرد .

.12 : این پروژه واسه OWASP هست و Network Mapping میکنه من فارسیش نمیدونم چطوری بگم ولی خب این کارا رو میکنه و خیلی کارای کاری که در ادامه بیشتر در موردش نوشتیم .

.13 : این ابزار هم ابزار قدرتمندی هست و واسه OSINT استفاده میشه و به صورت Passive Recon میکنه . اگه بخوايد مثلا توی Shodan.io و Censys.io و ... هم برآتون جستجو کنه باید API اونها رو برآش پیکربندی کنید، وگرنم به صورت عادی توی موتور های جستجو برآتون اطلاعات جمع اوری میکنه . اطلاعاتی مثل Names, Emails, IPs, subdomains, URLs برآتون جمع اوری خواهد کرد .

.14 : این ابزار زیاد ممکنه توی وب استفاده نشه، البته مت اینکه توی-Bug Bounty استفاده میشه ولی خب کارش اینه که DNS Enumeration بکنه و به تعداد زیاد DNS Query بفرسته و کارای اینطوری، منم زیاد نمیدونم فقط یه جستجو کردم و اینا رو نوشتیم .

15. **Passive Reconnaissance** : این Tool واسه پیدا کردن Subdomain ها استفاده میشه و به صورت <https://github.com/aboul3la/Sublist3r> باید و برآتون تست کنه . ابزار قوی هست .
16. **Brute-Force** : این ابزار گرچه به قول صفحه گیتها بش به صورت نسبتا Passive ساب دومین ها رو پیدا میکنه ولی به صورت Brute-Force هم این کار رو انجام میده . یه طور ایی Sublist3r توی حالت Brute-Force از این ابزار استفاده میکه .
17. **Virtual Host Fuzzing** : این ابزار خیلی استفاده میشه و خیلی هم معروفه و کارش از طریق Brute-Force هست و کارهای زیادی انجام میده ، Subdomain ها رو پیدا میکنه ، URI هارو چک میکنه ، Host هارو Brute-Force میکنه و های دیگه Fuzzing .
18. **Fuzzing Wordlist** : اینم کارش Fuzzing هست ، همونطور که از اسمش پیداست و یک Wordlist میگیره و توی Payload که بهش میدی کلمات داخلش رو جایگزین کلمه کلیدی FUZZ میکنه و تست میکنه و نتیجه رو نشون میده . ابزار هایی مثل gobuster , subbrute , ffuf , wfuzz FUZZING جهت استفاده میشن و حالا دست خودتونه کدو مشون رو انتخاب کنید و استفاده ببرید .
19. **Scanning Asset** : یک پلترمه بیشتر و کارش Scanning هست و برای ما Asset ها و اسیب پذیری ها رو کشف میکنه و یه چیز کامله واسه Notepad هم داره که میتوانید پادداشت کنید . حتی یک Bug-Bounty و Penetration Testing هایی رو توش بنویسید و باگهایی که پیدا کردید رو پادداشت کنید .
20. **Vulnerability Management** : این هم به مانند SnIper عمل میکنه و یک Wappalyzer : یک افزونه هست واسه مرورگر ها که میتوانید نصبش کنید و توی هر سایتی که میرید بهتون اطلاعاتی از اون سایت میده . مثلا از چه تکنولوژی هایی برای ساختش استفاده شده و CDN Provider چی هست ؟ نسخه های کتابخونه هاش چی هستند و ... البته نسخه پولی هم فک کنم داره .

- چیه ؟ حالا که بحث پیش اومد گفتم بیام یه خلاصه ای از این مفهوم بگم ، زمانی که شما میاید و ورودی های مختلف رو روی یک End-Point Fuzzing امتحان میکنید بهش میگن . ما دو نوع Fuzzing داریم :
1. **Web Fuzzing** : یعنی اینکه یک URL داریم و بیایم و مقادیری رو به عنوان پارامتر روش تست کنیم و Status Code هر Response که به ما میرسه رو چک کنیم .
 2. **Binary Fuzzing** : این Fuzzing یعنی اینکه بیایم و به یک نرم افزار تحت دسکتاب ، ورودی های مختلفی رو بدیم تا ببینیم کی Buffer-Over Flow Crash میکنه و مثلا Crash .

Fuzzing List ؟ واسه اینکه Fuzzing انجام بدیم ، باید یک لیست از کلماتی داشته باشیم که اونا رو امتحان کنیم و به نظر من لیست زیر یکی از بهتریناست و واسه هر چیزی هم لیست داره و فقط Fuzzing نیست و همچنین میتوانید لیست خودتون را بسازید . <https://github.com/danielmiessler/SecLists>

شـت ، خیلی زیاد بودن و دهنـم صاف شـد تـا نوشـتمـشـون ولـی خـب تـمـوم شـد صـلاح مـیدـونـم کـه یـه مـبحث جـدـید رو باـز کـنم چـون نـیـاز هـست کـه بـدونـیـم . بـایـد بـدونـیـم DNS چـطـورـی کـار مـیـکـنه و چـرا اـزـش استـفادـه مـیـشـه ؟ DNS Record هـا چـیـستـد ؟ من اـین جـزوـه رو قـبـلـا نـوـشـتم واسـه هـمـین فـقـط مـحتـواـش رو Copy/Paste مـیـکـنم .

What is DNS and Why we use it?

DNS یا Domain Name System را دفترچه تلفن اینترنت میگن . انسان از طریق Domain Name هایی مثل nytimes.com, google.com, yahoo.com, ... به اطلاعات انلاین دسترسی پیدا میکنه ولی وب بروزرهای IP کار میکند . یعنی ما انسان ها Domain Name ها رو درک میکنیم و وب بروزرهایی که ازشون استفاده میکنیم تا توی اینترنت بچرخیم، IP Address رو به جای Domain Name میفهمند . این یک مشکله دیگه واسه همین امدن و DNS رو بوجود اوردن که پروتکل لایه Application هست و روی پورت 53 کار میکنه و این پروتکل وقتی شما بهش یک Domain Name رو میدید بهتون منتظر با اون Domain Name رو بر میگردونه .

میدونیم که هر دیوایسی که متصل به اینترنت هست دارای یک IP Address اختصاصی می باشد و هر دیوایسی که خواست با یک دیوایس دیگه ارتباط برقرار کنه باید IP Address اون دیوایس رو داشته باشه . مثلا Device1 IP آدرس 192.168.1.1 داره و Device2 آدرس IP برابر 192.168.1.120 و برای اینکه این دو بتوان با هم ارتباط برقرار کنند باید آدرس IP های همیگه رو بدونن . واسه کامپیوتر ها اینکه یک آدرس IP رو حفظ کنن کار راحتیه و خب مشکلی ندارن اما ما انسانها توی حفظ کردن مطالب مشکل داریم مخصوصاً اگه مجموعه ای از اعداد باشه . به خاطر این مشکل، وارد صحنه شد و یک Database داره که مجموعه ای از IP Address ها و Domain Name های منتظر رو توی اون ذخیره کرند . مثلا Device1 (IP Address: 192.168.1.1) که گفتیم داره و device2.local Domain Name هم Device2 (IP Address: 192.168.1.120) داره و device1.local داره و (device1.local داره) چنین Record هایی توی پایگاه دادهش هست .

How does DNS work?

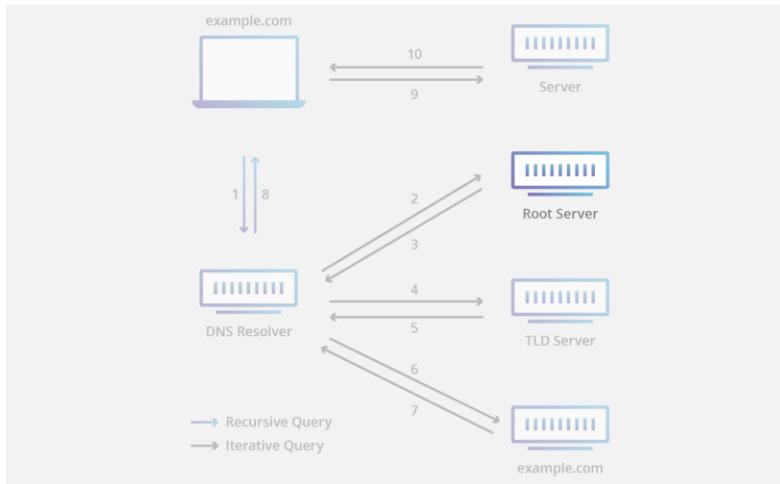
کاری که DNS انجام میده رو بهش میگن Resolution، یعنی تبدیل یک hostname مثل device1.local به یک IP Address مثل 192.168.1.1 . گفتیم که IP Address ها به هر دیوایسی که به اینترنت متصل است داده میشه و برای پیدا کردن هر دیوایس مانیز به IP Address اون دیوایس داریم . وقتی که کاربر میاد و میخواهد یک Web Page رو لود کنه، باید یک ترجمه یا تبدیلی انجام بشه و اون Domain Name که کاربر وارد کرده مثل device1.local، تبدیل بشه به این IP Address متناظر با این Domain Name تا بشه به اون Web Page دسترسی پیدا کرد . میدونید که Web Page ما روی یک وب سرور قرار داره . زمانی که ما چنین درخواستی یعنی درخواست بارگذاری یک Web Page از طریق Domain Name اون رو داریم، 4 تا DNS Server برای بارگذاری اون صفحه وب دخیل میشن :

1. DNS Recursor Server : DNS Recursor (Recursive Resolver) کتاب خونه کار میکنه و ازش درخواست میشه که یک کتابدار فرض کنیم که توی یک DNS Query محسوب میشه یعنی وقتی یک DNS Query ارسال میشه در اولین مرحله به یک Recursive Resolver داده میشه، حالا Recursive Resolver یا میاد و از طریق داده های Cache شده خودش پاسخ کلاینت رو میده و یا میاد و یک TLD ارسال میکنه و سپس پس از دریافت پاسخ از Root Nameserver، درخواستی دیگه به Nameserver و پس از دریافت پاسخ از TLD Nameserver، درخواستی به Authoritative Nameserver و پس از دریافت پاسخ از ایشون که حاوی IP Address موردنظر هست، Recursive Resolver میاد و پاسخ رو به کلاینت ارسال میکنه . Recursive Resolver ها اطلاعاتی که از Authoritative NameServer میکنن تا اگر بعد که یک کلاینت اومد و باز درخواست Resolution همون Domain Name رو داد سریعاً از طریق Cache ها بتوونه پاسخ بده . بسیاری از کاربران از Recursive Resolver هایی استفاده میکنن که ISP هاشون بهشون میگن و خب گرینه های دیگه ای هم هست که بشه ازشون استفاده کرد مثل: 1.1.1.1 که واسه Cloudflare هست، 8.8.8.8 و 4.4.4.4 که گوگل هستند و ... اینو هم یاد نره که بگم Recursive Resolver ها رو توی لینوکس ما توی /etc/resolv.conf تنظیم میکنیم، میبینید که در تصویر زیر ما دوتا nameserver تنظیم کردیم که اگه اولی جواب نداد بره و از دومی بپرسه :

```
kali㉿kali:~$ cat /etc/resolv.conf
# Generated by NetworkManager
search localdomain
nameserver 4.2.2.4
nameserver 8.8.8.8
```

2. Root nameserver : 13 تا Root nameserver توی دنیا وجود داره که Recursive Resolver ها اونها رو میشناسند و این 13 تا، اولین وقه برای یک درخواست از طرف Recursive Resolver هست . یعنی بعد از اینکه کاربر DNS Query خودش رو به Recursive Resolver داد، Recursive Resolver میاد و درخواستی رو به Root nameserver ها میده . یک Root nameserver درخواست رو درصورتی میپذیره که شامل یک Domain Name باشه و Root nameserver پاسخ رو با ارجاع

دادن Recursive Resolver به یک TLD nameserver میده و بر اساس پسوند Domain Name یعنی .ir, .com, .org, .net ... چند Root nameserver را پیشنهاد میکنه . هستند که یک سازمان ICANN غیر انتقائی هست.



لیست این 13 نوع Root nameserver را توی تصویر زیر میبینید :

HOSTNAME	IP ADDRESSES	OPERATOR
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	Verisign, Inc.
b.root-servers.net	170.247.170.2, 2801:1b8:10::b	University of Southern California, Information Sciences Institute
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	Verisign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

همچنین اگه بخوايم از طریق ابزار dig این مرحله رو مشاهده کنیم هم کافیه که از فلگ +trace استفاده کنیم و تمام مرحله که طی میشه تا به IP Address تارگت ختم شه رو خواهیم دید .

```

$ dig +trace nic.ir
; <>> Dig 9.19.17-1-Debian <>> +trace nic.ir
;; global options: +cmd
      2232 IN  NS  b.root-servers.net.
      2232 IN  NS  c.root-servers.net.
      2232 IN  NS  d.root-servers.net.
      2232 IN  NS  e.root-servers.net.
      2232 IN  NS  f.root-servers.net.
      2232 IN  NS  g.root-servers.net.
      2232 IN  NS  h.root-servers.net.
      2232 IN  NS  i.root-servers.net.
      2232 IN  NS  j.root-servers.net.
      2232 IN  NS  k.root-servers.net.
      2232 IN  NS  l.root-servers.net.
      2232 IN  NS  m.root-servers.net.
      2232 IN  NS  a.root-servers.net.
      2232 IN  RRSIG
Sh /HfygT9a6880Op231aG04hsf41878kr/A+o6GRDw04dgraw1Bq1Qauru eZ37q73tVkdE3jKmqlN+X6ULtKdvqQIPB8IPCIYjCVKodoVK25LaFjqD +QhdStM69VBQSYe2GMPYLXTe6Okay7LXFlNlrF1InvG4IFax6bMkHR nlgKtc+R7Ins350bw00Zek5f187f1kt7m81/y
;; Received 1097 bytes from 4.2.2.4#53(4.2.2.4) in 151 ms

```

مرحله اول رو میبینید که تمام 13 نوع Root nameserver هایی که توسط شناخته شده هستند رو نشون داده .

3. عبارت "TLD" با Top-Level Domain یعنی ".com" در یک Domain Name عبارت میگن . کلا ما دو نوع TLD داریم (البته سه نوع هست که پیکیش استفاده نمیشه) عبارت اند از :

- .com, .org, .net, .edu, .edu : به دامنه هایی که مخصوص کشور خاصی نیستند مثل • ... و .gov

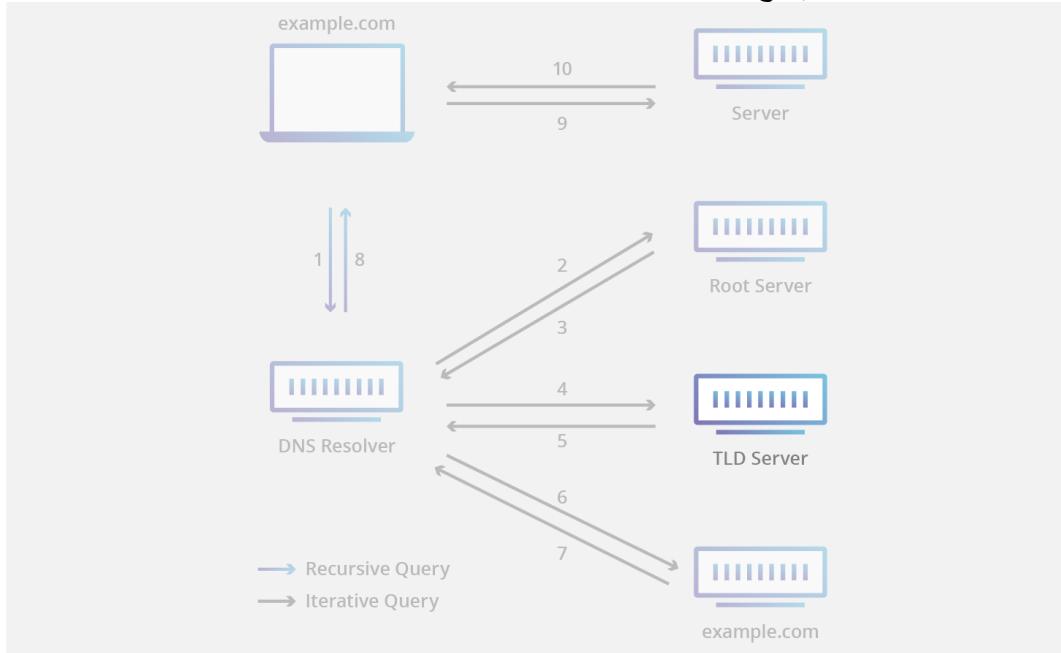
.ir, .us, .ru, .uk: به دامنه هایی که مخصوص یک کشور خاص هستند مثل .uk، Country Code Top-Level Domains •

و ...

Recursive TLD nameserver ها اطلاعات تمام دامنه هایی که یک TLD خاص را دارن نگهداری میکنند. درخواست Root nameserver وقیتی به Root nameserver میرسه Resolver بر اساس اینکه TLD دامنه چی هست Nameserver هایی رو در جواب ارسال میکنند که این TLD nameserver ها تمام اطلاعات دامنه هایی با اون TLD رو دارند و میشه ازشون پرسید. توی تصویر زیر که ادامه دستور dig با فلگ +trace میبینید که، چون ما اومدیم و یک دامنه ir. رو درخواست کردیم، به ما TLD nameserver های شامل دامنه های ir. رو برگرداند.

```
ip.          172800 IN  NS      a.nic.ir.
ip.          172800 IN  NS      b.nic.ir.
ip.          172800 IN  NS      c.nic.ir.
ip.          172800 IN  NS      d.nic.ir.
ir.          86400  IN  DS      47380 13 2 7AB287956A55ABD60DC8697C2049738FD5EEAB6F2070152B015D5992 9521DADA
ir.          86400  IN  DS      47380 13 4 F3E9ABF03AE031CB012F6B45C880B1B5CA68CD9046080C14AFBD 491AE3A9F60DFBFC4320111144C375D648A3CDF6
ir.          86400  IN  DS      47380 13 8 86400 20231121050000 20231129040000 46780 .bZ48c1Tp991x4gfeIDnTayUzthX0fCe6nv/w1lDpmw71bVlx464FYD m0X2zHn6tQmPQvsza3vnGln3N50mKsAcMfZ+oL9UCnigMNGlwJATAx
ir.          86400  IN  RRSIG  DS  8 1 86400 20231121050000 20231129040000 46780 .bZ48c1Tp991x4gfeIDnTayUzthX0fCe6nv/w1lDpmw71bVlx464FYD m0X2zHn6tQmPQvsza3vnGln3N50mKsAcMfZ+oL9UCnigMNGlwJATAx
; Received 674 bytes from 170.247.170.2#53(b.root-servers.net) in 147 ms
; Received 674 bytes from 170.247.170.2#53(b.root-servers.net) in 147 ms
```

بعد از اینکه Recursive Resolver در پاسخ دریافت شدند، Recursive Resolver ها توسط Authoritative nameserver میاد و درخواستی رو به اونها ارسال میکنند که در پاسخ Authoritative nameserver هایی رو دریافت خواهد کرد.



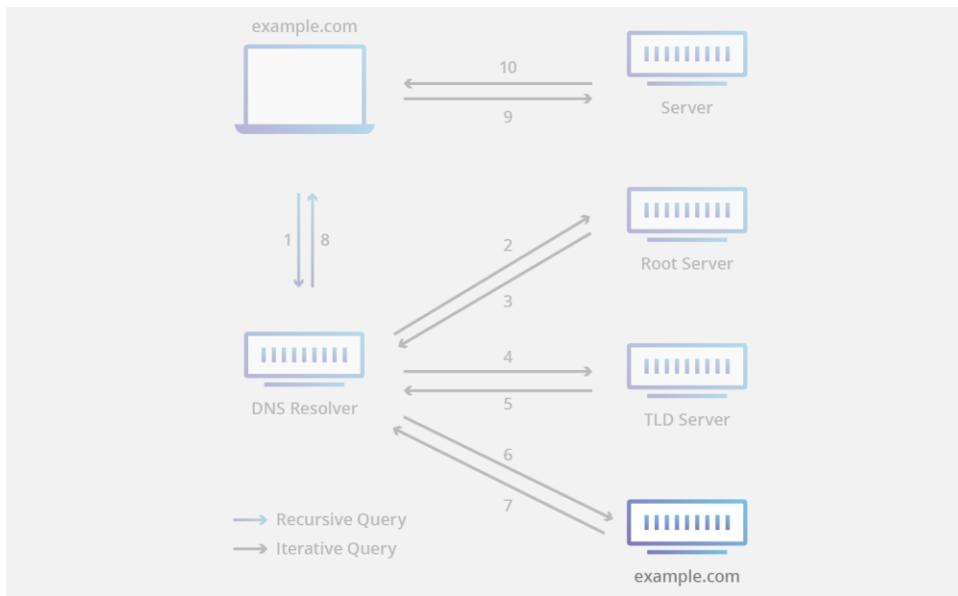
4. وقتی که یک Recursive Resolver یک پاسخ رو از Authoritative nameserver دریافت میکنه، این پاسخ Recursive Resolver رو ارجاع میده به Authoritative nameserver ها. Authoritative nameserver مرحله از مراحل پیدا کردن IP Address توسط Recursive Resolver است. Recursive Resolver شامل اطلاعات Authoritative nameserver میشود. مشخصی از Domain Name (مثل google.com) است و میتوانه IP Address (google.com) رو که متعلق به سرور تارگت است به بده یا اگر دامنه CNAME Record داره، اون رو در پاسخ ارسال کنه. Recursive Resolver بر میگردونه میتوانه متفاوت باشه که در ادامه دربارشون صحبت میکنیم. اون Record که IP Address رو داخل خودش داره A Record است. در تصویر زیر که ادامه اجرای دستور dig با فلگ +trace میبینید هست، پاسخ irantalent.ir Authoritative nameserver رو میبینید :

```
irantalent.ir.    180   IN  A      185.143.234.5
irantalent.ir.    180   IN  A      185.143.233.5
;; Received 112 bytes from 185.143.232.253#53(h.ns.arvancdn.ir) in 55 ms
```

میبینید که در اخرین خط نوشته که این پاسخ از h.ns.arvancdn.ir گرفته شده است که در واقع TLD nameserver این دامنه می باشد و مشخص میکنه که دامنه irantalent.ir یعنی ما میتوانیم از این طریق، پشت ابر بودن یک دامنه رو تشخیص بدیم. یا مثلا تصویر زیر از یک دامنه دیگه نشون میده که این دامنه پشت Cloudflare CDN هست :

```
shodan.io.        300   IN  A      188.114.98.0
shodan.io.        300   IN  A      188.114.99.0
shodan.io.        300   IN  RRSIG  A 13 2 300 20231130162051 20231128142051 34505 shodan.io. y3XpEuhwXz25ce5QntKMPjPcRY33AeOABuoSdP96k4CBmISsnmb4XK K9KeJhNExBU8+fI1xM1V6Bu1vouk6A==
;; Received 175 bytes from 172.64.33.111#53(ed.ns.cloudflare.com) in 151 ms
```

و اگه دامنه مورد نظر ما پشت CDN نباشه میتوnim IP Address واقعی اون رو بدست بیاریم.



این تمام روندی بود که قبل از اینکه Request ما به طرف وب سرور تارگت ارسال بشه طی میشه تا بتونیم به IP Address دست یابیم که به اون وب سرور ختم میشه، حال میتونه یک Reverse Proxy باشه که درخواست ما رو به سمت وب سرور اصلی میفرسته و یا هم میتونه متعلق به خود وب سرور باشه که مستقیماً باهاش تعامل برقرار میکنیم و گاهی اوقات هم متعلق به یک هاست اشتراکی است که با تنظیم کردن هدر Host میتونیم به وبسایت اصلی بريم . اما گفتیم که Record هایی که بر میگیرده میتونه انواع مختلفی داشته باشه و به صورت پیش فرض ما شاهد A Record هستیم که تو ش قرار داره :

```
kali㉿kali:~$ dig scsi4me.com
; <>> DiG 9.19.17-1-Debian <>> scsi4me.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<< opcode: QUERY, status: NOERROR, id: 20752
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;scsi4me.com.      IN      A
;; ANSWER SECTION:
scsi4me.com.    300   IN      A      208.113.190.175
;; Query time: 335 msec
;; SERVER: 4.2.2.4#53(4.2.2.4) (UDP)
;; WHEN: Wed Nov 29 10:09:33 EST 2023
;; MSG SIZE  rcvd: 56
```

. خب بريم سر وقت DNS Record ها

What are DNS Records?

های A چیستند؟ حرف A مخفف IP Address است و پایه ای ترین DNS Record ممکن هست. A Record به دامنه A Record موردنظر اشاره میکند. مثلاً اگه دامنه A Record را درخواست کنیم به جواب زیر میرسیم:

```
kali㉿kali:~$ dig A cloudflare.com
; <>> DiG 9.19.17-1-Debian <>> A cloudflare.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 46577
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;cloudflare.com. IN A
;; ANSWER SECTION:
cloudflare.com. 42 IN A 104.16.132.229
cloudflare.com. 42 IN A 104.16.133.229
;; Query time: 143 msec
;; SERVER: 4.2.2.4#53(4.2.2.4) (UDP)
;; WHEN: Wed Nov 29 10:16:48 EST 2023
;; MSG SIZE rcvd: 75
```

میبینید که از طریق dig تونستیم فقط درخواست A Record کنیم و کافیه که بعد از کلمه dig نام Record را بزنیم که در مثال بالا A هست. در جواب دوتا IP Address یکسان را برای ما برگرداند که اشاره میکند به سایت cloudflare.com. دقت کنید که تنها به IPv4 اشاره میکند و عموم سایتها یک A Record دارند مگر اینکه به خاطر Load Balancing تعداد اونها بیشتر بشه.

حالا این A Record کجا استفاده میشه؟ مرورگر ها برای بدست اوردن ادرس IP دامنه ای که کاربر قصد دسترسی به اون رو دارد درخواست DNS خودشون را ارسال میکنند و A Record را مد نظر رو بدست میارن و دسترسی رو ایجاد میکنند.

های AAAA چیستند؟ AAAA Record اشاره میکند به IPv6 متعلق به یک دامنه و کاملاً شبیه به A Record است با این تفاوت AAAA به IPv6 اشاره میکند. توی تصویر زیر میبینید که از طریق dig تونستیم AAAA Record را برای دامنه shodan.io بدست بیاریم.

```
kali㉿kali:~$ dig AAAA shodan.io
; <>> DiG 9.19.17-1-Debian <>> AAAA shodan.io
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 30522
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;shodan.io. IN AAAA
;; ANSWER SECTION:
shodan.io. 300 IN AAAA 2606:4700::6812:cee
shodan.io. 300 IN AAAA 2606:4700::6812:dee
;; Query time: 208 msec
;; SERVER: 4.2.2.4#53(4.2.2.4) (UDP)
;; WHEN: Wed Nov 29 10:24:46 EST 2023
;; MSG SIZE rcvd: 94
```

برخلاف A Record که همه دامنه ها یک یا بیشتر از یک مورد رو داشتن، AAAA Record رو همه دامنه ها ندارند چرا که هنوز اونقدر مورد استفاده قرار نگرفته و خب میتونم بگم کم کم داریم میریم سمت IPv6.

MX Record چیست؟ این رکورد که بهش Mail Exchange Record میگن در واقع زمانی استفاده میشه که لازم میشه یک ایمیل رو به سمت Mail Server هدایت کند و نشون میده که اون ایمیل چطوری توسط SMTP باید مسیر یابی بشه و به کجا برسه. این رکورد همیشه به یک یا چند دامنه اشاره میکند. مثلاً اگه بخوایم یک ایمیل رو از Gmail به Yahoo بفرستیم، میاد و MX Record های Yahoo.com رو میگیره و میبینه که مثلاً نتیجه به شکل زیر هست:

```
kali㉿kali:~$ dig mx yahoo.com
; <>> DiG 9.19.17-1-Debian <>> mx yahoo.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 56194
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1
;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
@yahoo.com.           IN      MX
;
;; ANSWER SECTION:
@yahoo.com.        1391    IN      MX    1 mta7.am0.yahoodns.net.
@yahoo.com.        1391    IN      MX    1 mta6.am0.yahoodns.net.
@yahoo.com.        1391    IN      MX    1 mta5.am0.yahoodns.net.

;; Query time: 172 msec
;; SERVER: 4.2.2.4#53(4.2.2.4) (UDP)
;; WHEN: Wed Nov 29 13:57:23 EST 2023
;; MSG SIZE rcvd: 117
```

یعنی اگه بخواهد اون ایمیل رو به سمت ایمیل سرور یاهو بفرسته باید به سمت یکی از اینها ارسال کنه.

DNS چیست؟ اگه یادتون باشه که قطعا هست، گفتیم بعد از اینکه TLD nameserver ها دریافت شد و به یکی از اونها Query ارسال شد در جواب Authoritative Nameserver ها برگردانده میشے. گفتیم که Authoritative Nameserver ها چی هستند و کارشون اینه که در نهایت Resolve رو انجام بدند و IP Address را رو به ما برگردانند. اگه بخواهیم Authoritative Nameserver های یک دامنه رو ببینیم، کافیه که بگیم فقط NS Record ها رو به ما بده. اگه به صورت عادی بیایم dig +trace با فلگ +trace روی yahoo.com انجام بدیم تا مرحله برگرداندن Authoritative Nameserver ها به ما لیست زیر رو میده:

```
com.          172800  IN      NS      c.gtld-servers.net.
com.          172800  IN      NS      d.gtld-servers.net.
com.          86400   IN      DS      30909 8 2 E2D3C916F6DEEAC73294E8268FB5885044A833FC5459588F4A9184CF C41A57
com.          86400   IN      RRSIG   DS 8 1 86400 20231212050000 20231129040000 46780 . EF4swrsy5wUZtYJp7pBzyI
2 k50xFYtkSFaW6u8J8tmSVmLW6Rm2EcRShsZC/xZSy5qbJG7dYNC3RoP dvY4C8Q3682Wc04EC7m+728dYiq4KIrYymfWbhmV3zKxIIju7PPBBGj j CAT1
uRJMh9uiytVqUoJLIK 0UPDWA==
;; Received 1169 bytes from 198.41.0.4#53(a.root-servers.net) in 148 ms

yahoo.com.    172800  IN      NS      ns1.yahoo.com.
yahoo.com.    172800  IN      NS      ns5.yahoo.com.
yahoo.com.    172800  IN      NS      ns2.yahoo.com.
yahoo.com.    172800  IN      NS      ns3.yahoo.com.
yahoo.com.    172800  IN      NS      ns4.yahoo.com.

CK0POJMG874LJREF7EFN8430QVIT8BSM.com. 86400 IN NSEC3 1 1 0 - CK0Q2D6NI4I7EQH8NA30NS61048UL865 NS SOA RRSTIG DNSKEY NSEC3P,
CK0POJMG874LJREF7EFN8430QVIT8BSM.com. 86400 IN RRSIG NSEC3 8 2 86400 20231204052540 20231127041540 63246 com. uXxG0/4gLw
krhzdNvaiKTq fNJWYCiTL103DMza87oftAUQ1F/wVDxp93rpwQJdeTdICHFJCjpe+05 NdfG4PyUZRqcsTLRD3uXU8xR2fmsZuTCUdCxqdxOKQuEIA==
GPIOVE5CC3CA0D1H14G1GI4J0835GEKB NS DS RRSIG
;; Received 1169 bytes from 198.41.0.4#53(a.root-servers.net) in 148 ms
```

حالا اگه دستور رو به شکل زیر وارد کنیم هم همین لیست رو میگیریم و دیگه چیزی اضافی به ما داده نمیشه:

```
kali㉿kali:~$ dig ns yahoo.com
; <>> DiG 9.19.17-1-Debian <>> ns yahoo.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 1229
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 10
;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
@yahoo.com.           IN      NS
;
;; ANSWER SECTION:
@yahoo.com.        31771   IN      NS      ns3.yahoo.com.
@yahoo.com.        31771   IN      NS      ns4.yahoo.com.
@yahoo.com.        31771   IN      NS      ns1.yahoo.com.
@yahoo.com.        31771   IN      NS      ns2.yahoo.com.
@yahoo.com.        31771   IN      NS      ns5.yahoo.com.

;; ADDITIONAL SECTION:
ns1.yahoo.com.     71531   IN      A      68.180.131.16
ns2.yahoo.com.     71374   IN      A      68.142.255.16
ns3.yahoo.com.     874     IN      A      27.123.42.42
ns4.yahoo.com.     71614   IN      A      98.138.11.157
ns5.yahoo.com.     72173   IN      A      202.165.97.53
ns1.yahoo.com.     39579   IN      AAAA   2001:4998:1b0::7961:686f:6f21
ns2.yahoo.com.     34487   IN      AAAA   2001:4998:1c0::7961:686f:6f21
ns3.yahoo.com.     672     IN      AAAA   2406:8600:f03f:1f8:1:1003
ns5.yahoo.com.     27975   IN      AAAA   2406:2000:1d0::7961:686f:6f21

;; Query time: 139 msec
;; SERVER: 4.2.2.4#53(4.2.2.4) (UDP)
;; WHEN: Wed Nov 29 14:10:26 EST 2023
;; MSG SIZE rcvd: 320
```

اون قسمت پایین یعنی ADDITIONAL SECTION: A Record های هر کدام از این NS ها که IP Address هاشونه و در برخی اوقات نشون نمیده . راستی اگه فلگ +short رو بزنید فقط و فقط نتیجه ها رو نشون میده . مثلاً توى دستور بالا فقط و فقط NS Record ها رو نشون میده بدون هیچ چیز اضافی :

```
kali@kali:~$ dig ns yahoo.com +short
ns4.yahoo.com.
ns3.yahoo.com.
ns2.yahoo.com.
ns1.yahoo.com.
ns5.yahoo.com.
```

Canonical Name CNAME Record هست اشاره میکنه از یک دامنه مستعار به یک دامنه مخفف Canonical Name چیست ؟ یعنی ابتدایی . همه CNAME ها باید به یک دامنه اشاره کنند و نمیتوانند IP Address یا چیز دیگری باشند . هر CNAME خودش میتوانه به یک CNAME دیگه اشاره کنه تا جایی که در نهایت به دامنه اصلی اشاره کنه که درواقع دامنه ای است که A Record داره . مثلاً فرض کنید دامنه blog.example.com داره با مقدار example.com . این به این معناست که وقتی یک DNS Server میاد و DNS Record های blog.example.com را رو نشون میده در واقع داره یک DNS Lookup واسه example.com انجام میده و در نهایت IP Address متعلق به example.com هست . در این مثال ما میتوانیم بگیم که نام example.com برای دامنه Canonical Name هست . به عنوان مثالی دیگه میتوانیم بگیم که وقتی یک CNAME Record www.example.com با مقدار example.com داره این به این معناست که هر وقت ما میگیم example.com را دریافت کنه میره و در نهایت A Record ما میخواه www.example.com DNS Query و www.example.com را دریافت میکنه چون که دامنه Canonical Name دامنه www.example.com دارمه example.com هست . بیشتر اوقات وقتی سایت ها Subdomain هایی مثل shop.example.com یا blog.example.com دارند، این Subdomain ها CNAME Record هایی دارند که اشاره میکنند به Root Domain یعنی example.com . از این طریق اگر IP Address هاست تغییر کنه و بقیه Subdomain ها نیاز نیست تغییری داشته باشند چرا که همه اونها در نهایت به example.com اشاره میکند . دقت کنید که وقتی یک Subdomain مثل shop.example.com داره مثل A Record shop.example.com هر دوی اونها به یک IP Address میکنند . یعنی وقتی کاربر shop.example.com رو درخواست میکنه، A Record اون مقداری برابر دامنه example.com داره و زمانی که به کلاینت به وب سرور رسید، وب سرور به URL کاربر نگاه میکنه و میبینه که کاربر قصد رفتن به سایت دامنه shop داره و میاد و صفحه shop رو به او تحويل میده .

```
kali@kali:~$ dig cname login.live.com
; <>> DiG 9.19.17-1-Debian <>> cname login.live.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<< opcode: QUERY, status: NOERROR, id: 50017
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: MBZ: 0x0005, udp: 1232
;; COOKIE: 917b53ed82b9784010000006567a3726ca00c734e902370 (good)
;; QUESTION SECTION:
;login.live.com.      IN      CNAME
;
;; ANSWER SECTION:
login.live.com.      5      IN      CNAME   login.msa.msidentity.com.
```

توى دامنه های CDN ها میان و از CNAME استفاده میکن که مثلاً یه دامنه چرت و پرت اشاره میکنه در نهایت به دامنه اصلی . این کار مرسومه و شاید ببینید .

اگه بخوايد تمام DNS Record هایی که یک دامنه داره ببینید، میتوانید از dig استفاده کنید و به جای اسم یک DNS Record خاص بنویسید که موجب میشه همه DNS Record ها رو به شما نشون بده : any

```

kali㉿kali:~$ dig any google.com @8.8.8.8

; <>> DIG 9.19.17-1-Debian <>> any google.com @8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>>HEADER<<- opcode: QUERY, status: NOERROR, id: 1976
;; Flags: qr rd ra; QUERY: 1, ANSWER: 22, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.           IN      ANY

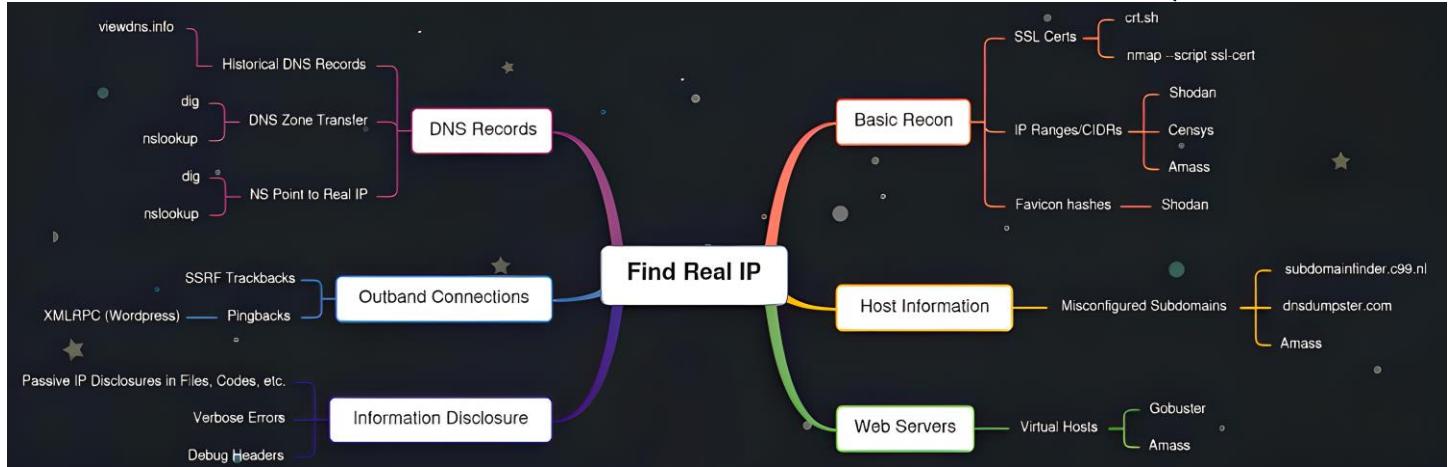
;; ANSWER SECTION:
google.com.          273    IN      A       172.217.17.46
google.com.          273    IN      AAAA    2a00:1450:4019:807::200e
google.com.          3573   IN      TXT     "globalsign-smime-dv=CDX+XFHJu2wmL6/Gb8+59BsH31KzUr6c112B PvqjX8="
google.com.          21573   IN      NS      ns4.google.com.
google.com.          3573   IN      TXT     "atlassian-domain-verification=5YjTmWmj192ewqkx2oXmBa0607d5zKonr6eakvHXG877zzkFQt08PQ9QsKnbf4I"
google.com.          21573   IN      NS      ns3.google.com.
google.com.          21573   IN      CAA    0 issue "pki.goog"
google.com.          21573   IN      NS      ns1.google.com.
google.com.          33     IN      SOA    ns1.google.com. dns-admin.google.com. 585914019 900 900 1800 60
google.com.          3573   IN      TXT     "webdomainverification.BYX6G-6e6922db-e3e-4a36-904e-a805c28087fa"
google.com.          3573   IN      TXT     "google-site-verification=w08N711jTNkez149swvM48f8_9xeREV4oB-0Hf5o"
google.com.          3573   IN      TXT     "docsign1=1b0a6754-49b1-4db5-8540-d2c1264b289"
google.com.          3573   IN      TXT     "NS=d46b89AB2B99670CE15412F62916164C0B20EB"
google.com.          21573   IN      NS      ns2.google.com.
google.com.          273    IN      MX      10 smtp.google.com.
google.com.          3573   IN      TXT     "google-site-verification=T9-D8e4R80X4v0M4U_bJ9cpO3M0nikft0jAgjmsQ"
google.com.          3573   IN      TXT     "facebook-domain-verification=22rm551c4Kab0bxsw536t1ds4h5"
google.com.          3573   IN      TXT     "onetrust-domain-verification=de01ed21f2fa4d8781cbc3ffbb89cf4cf"
google.com.          3573   IN      TXT     "docsign0=5958488-4752-4ef2-95eb-aa7ba8a3bd0e"
google.com.          3573   IN      TXT     "v=spf1 include:_spf.google.com ~all"
google.com.          21573   IN      HTTPS   1 . alpn="h2,h3"
google.com.          3573   IN      TXT     "apple-domain-verification=30afIBcvSuDv2PLX"

;; Query time: 92 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (TCP)
;; WHEN: Wed Nov 29 15:55:39 EST 2023
;; MSG SIZE  rcvd: 1120

```

شـت، DNS خـلـی گـسـترـه هـست و هـر چـی بـگـیـم باـزـم اـطـلاـعـاتـی اـزـش مـیـمـونـه و لـازـم هـم هـست کـه بـدوـنـمـانـم الـاـن بـگـم كـافـيـه و يـا اـدـامـه بـدم وـلـی خـبـخـسـته شـدـم 😊 وـالـبـتـه خـلـی چـیـزا روـيـادـگـرـقـم وـبـه نـظـرـمـبـاـيـدـتـکـرـار وـتـکـرـارـکـنـیـم تـاـيـدـمـوـنـنـرـهـ. حـالـا چـراـيـناـرـوـ گـفـتـم ؟ عـلـشـ اـيـنهـ کـهـ گـاهـیـ اوـقـاتـ کـهـ چـیـ عـرـضـ کـنـمـ، خـلـیـ اـزـ اوـقـاتـ یـهـ سـوـتـیـ هـایـ توـیـ DNS Nameserver هـایـ یـکـ تـارـگـتـ مـیـتـونـهـ پـیـشـ بـیـادـ کـهـ مـوـجـ بـرـخـیـ حـفـراتـ اـمـنـیـتـیـ بشـهـ مـثـلـ Zone Transfer Attack وـ نـمـیدـونـمـ دـیـگـهـ چـیـ چـونـ خـودـمـ هـمـ هـنـوزـ بـلـدـ نـیـسـتمـ وـبـهـ نـظـرـمـ بـهـتـرـهـ بـلـشـ باـشـیـمـ وـ هـمـچـنـینـ بـرـخـیـ اوـقـاتـ اـزـ هـمـینـ اـطـلاـعـاتـ مـیـشـهـ بـرـایـ کـشـفـ IP Address پـیـشـ اـبـرـ استـفـادـهـ کـرـدـ (ـمـنـ خـودـمـ اـیـنـکـارـ رـوـ کـرـدـ حـقـيقـتـاـ وـ جـوـابـ دـادـ) کـهـ درـ اـدـامـهـ توـضـيـحـ دـادـ .

پشت **IP Address** یکی از اولین کارهایی که باید روی یک تارگت که پشت یک **CDN** قایم شده انجام بدیم، اینه که **Find Real IP Address** اون رو پیدا کنیم . برای این کار 6 راه متنوع وجود دارد که در تصویر زیر میبینید و اگه از طریق این 6 راه نشد پیدا کنید دیگه تقریباً میشه گفت که نمیشه پیداش کرد .



۱. **DNS Records**: برخی وبسایت ها مثل نمونه زیر خودشون رو پشت **CDN** های مختلف مخفی میکنن، تا امکان دسترسی به **IP Address** واقعی سرورشون نباشه.

```

kali㉿kali:~$ dig ns booktopia.com.au
; <>> DiG 9.19.17-1-Debian <>> ns booktopia.com.au
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 16965
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: udp: 512
;; QUESTION SECTION:
;booktopia.com.au. IN NS
;;
;; ANSWER SECTION:
booktopia.com.au. 21600 IN NS ns-1465.awsdns-55.org.
booktopia.com.au. 21600 IN NS ns-1686.awsdns-18.co.uk.
booktopia.com.au. 21600 IN NS ns-418.awsdns-52.com.
booktopia.com.au. 21600 IN NS ns-898.awsdns-48.net.

;; Query time: 331 msec
;; SERVER: 8.8.8#53(8.8.8) (UDP)
;; WHEN: Wed Nov 29 17:40:45 EST 2023
;; MSG SIZE rcvd: 185
  
```

گاهی اوقات میتوانیم از طریق **DNS Record** ها به **IP Address** واقعی یک سایت دسترسی پیدا کنیم . به سه روش میتوانیم این کار رو انجام بدیم .

- میتوانیم از سایتهایی مثل **viewdns.info** استفاده کنیم و یک **Domain Name** رو بهشون بدهیم تا تاریخچه تغییرات **Historical DNS Records** داده شود . ممکن است، یکی از همین **IP Address** ها ادرس **IP Address** واقعی سرور باشد .

- **DNS Zone Transfer**: یکی از مشکلات امنیتی **DNS Server** هاست، که میشه از طریق ابزار هایی مثل **dig** و ... داده هایی رو ازش بدست اورد که شاید یکی از اونها **IP Address** واقعی تارگت ما باشد . در ادامه بررسی میکنیم .

- **NS Point to Real IP Address**: اگه **NS Record** های یک وبسایت که پشت ابر هست رو بگیریم خواهیم دید که اشاره میکنه مثل **Cloudflare** یا ... اما میدونیم که قبل از اینکه این وبسایت پشت ابر بره دو تا **NS Record** داشته که توی اونها **IP Address** واقعی اون وبسایت وجود داره و اگه ازشون درخواست کنیم بهمون بر میگردونن . مثلًا فرض کنید که یک دامنه داریم مثل **example.com** و **ns1.example.com** و **ns2.example.com** هستند و میتوانید از طریق **dig** به صورت مستقیم بدون ارسال **DNS Recursive Resolver** هایی مثل **4.2.2.4**, **8.8.8.8** و ... به همین **ns1.example.com** یا **ns2.example.com** درخواست **DNS** کنیم . به شکل زیر :

```

kali㉿kali:~$ dig example.com @ns1.example.com
  
```

در واقع کاری که ماتوی دستور بالا میکنیم اینه که میایم و Authoritative Nameserver های یک دامنه رو پیدا میکنیم، این Authoritative Nameserver ها اونهایی هستند که زمانی که دامنه ثبت شده اند، توسط اون سایت ثبت کنند به Domain داده شده اند و در اون موقع ازش استقاده میشده، ما میایم و مستقیماً بدون اینکه از Recursive Resolver ها شروع کنیم تا به Authoritative Nameserver ها برسیم از همینا میپرسیم و قاعدتاً در برخی اوقات پاسخ میدند.

اگه بتونیم از طریق این روش ها IP Address واقعی یک تارگت رو پیدا کنیم در برخی اوقات نیاز هست که بیایم و هدر Host رو در Request هامون برابر با دامنه قرار بدم و گرنه ممکن است که کار نکنه و خطابه چون ممکن هاست اشتراکی ما بین چندین دامنه باشه.

.2. **Outbound Connections**: یعنی اینکه راهی رو پیدا کنیم که از سمت Origin Server های تارگت یک درخواستی به سمت ما بیاد و در این صورت ما بتونیم IP Address واقعی رو پیدا کنیم. یکی از حفرات امنیتی که میتونه منجر به این اتفاق بشه SSRF هست که یه کمی توی جزوایات قبلی درموردش صحبت کردیم، SSRF یعنی اینکه بتونیم از طرف سرور های یک تارگت، یک درخواست بزنیم به یه جای خاص، یکی از توابع PHP که میتونه منجر به این حفره امنیتی بشه، file_get_contents، هست که علاوه بر نام یک فایل، میتونه یک URL رو هم بخونه.

یکی دیگه از چیزهایی که میتونه منجر به Outbound Connection بشه، اون هم توی سایتهای Wordpress، فایل XMLRPC هست. این فایل امکان این رو برای سایت های وردپرسی فراهم میکنه که به صورت Remote Authentication بتوان سایت خودشون رو مدیریت کنن، میشه از طریق Post قرار داد، پست ها رو اپدیت کرد و ... که نیاز به Authentication داره. ولی خب توابعی دیگه داره که ممکن است در برخی اوقات نیاز به Authentication نداشته باشه، مثل pingback.ping که میتونه pingback.ping رو انجام بد. این فایل از طریق متد POST یک XML رو دریافت میکنه و در جواب هم یک XML رو فرسته. توی تصویر زیر میبینید که من ازش خواستم لیست متد هایی که داره رو بهم نشون بده و بر ام برگردانده و pingback.ping هم داخلش هست.

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
<pre> 1 POST /xmlrpc.php HTTP/2 2 Host: www.uri.edu 3 Sec-Ch-Ua: "Not=A?Brand";v="99", "Chromium";v="118" 4 Sec-Ch-Ua-Mobile: ?0 5 Sec-Ch-Ua-Platform: "Windows" 6 Upgrade-Insecure-Requests: 1 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.88 Safari/537.36 8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8 application/signed-exchange;v=b3;q=0.7 9 Sec-Fetch-Site: none 10 Sec-Fetch-Mode: navigate 11 Sec-Fetch-User: ?1 12 Sec-Fetch-Dest: document 13 Accept-Encoding: gzip, deflate, br 14 Accept-Language: en-US,en;q=0.9 15 Content-Length: 128 16 17 <?xml version="1.0" encoding="iso-8859-1"?> <methodCall> <methodName> system.listMethods </methodName> <params> </params> </methodCall></pre>	<pre> 12 X-Cache: MISS, MISS, MISS, MISS, MISS 13 X-Cache-Hits: 0, 0, 0, 0, 0 14 X-Timer: S1701348348.694176,VSO,VE324 15 Vary: Accept-Encoding, orig-host 16 Content-Length: 4272 17 18 <?xml version="1.0" encoding="UTF-8"?> <methodResponse> <params> <param> <value> <array> <data> <value> <string> system.multicall </string> </value> <value> <string> system.listMethods </string> </value> <value> <string> system.getCapabilities </string> </value> <value> <string> demo.addTwoNumbers </string> </value> <value> <string> demo.sayHello </string> </value> <value> <string> pingback.extensions.getPingbacks </string> </value> <value> <string> pingback.ping </string> </value> </array> </value> </param> </params> </methodResponse></pre>

بیشتر از این توضیح نمیدم چون بعداً قرار درموردش بحث کنیم. تا همینجا درمورد xmlrpc.php کافیه. همین که فعلاً بدونید میشه از طریق درخواست ارسال کرد و IP Address رو بدست اورد کافیه.

.3. **Information Disclosure**: گاهی اوقات ممکن هست که از سمت سرور اطلاعاتی برای ما ارسال بشه که برخی دادهها رو لو بده مثل همین IP Address واقعی رو نشون بده. این اطلاعات میتونه توی فایلهای مختلف باشه یا توی کدهایی باشه که میتوانیم ببینیم. در بعضی موارد هم دیده شده که سایت خطاب داده و این خطاب در حالت Verbose بوده و اطلاعات سایت توی اون صفحه خطاب نشون داده شده و حتی چند بار هم دیدم که سایت توی Debug Header ها اطلاعاتی مثل IP Address واقعی رو لو داده.

.4. **Basic Reconnaissance**: قاعدتاً با Reconnaissance کردن یک تارگت، احتمال داره که یه جایی بتونیم Real IP Address رو بدست بیاریم. سه حالت واسه این روش وجود داره که عبارت اند از :

- اره، برسی SSL Certification میتونه موجب پیدا شدن Real IP Address بشه. قاعده اینجوریه که میایم و بررسی میکنیم چه دامنه هایی یا ساب دامنه هایی SSL Certification یکسان با تارگت ما دارن و برخی از اونا گاهی میتونن پشت ابر نباشن و Real IP Address رو فلاش کنن و چرا میتونن چنین کنن؟ چون وجود یک SSL Certification یکسان به معنی وجود وب سرور یکسان هست. برای اینکه چنین چیزی رو بررسی کنیم میتوانیم از هم استقاده کنیم. اینا قابلیت اینو دارند که بر اساس SSL Certification یا <https://censys.io> یا <https://crt.sh>، دامنه ها

و ساب دامنه هایی رو به شما پیشنهاد کنن و SSL هم یک Script داره به نام ssl-cert که برآتون اطلاعات Certification یک تارگت رو نشون میده .

IP Address/CIDRs: گاهی اوقات میتوانید که تارگت داره مثلا از افرانت سرویس میگیره ولی نمیتوانید که اون چیه . میتوانید برد و توی اینترنت دنبال CIDR های اون شرکت بگردید و باید توی Shodan, Censys و ... جستجوشون کنید و سعی کنید از طریق Host Header تارگت بهشون درخواست ارسال کنید و هر کدام که پاسخ داد رو میتوانید به عنوان Real IP Address در نظر بگیرید . هم یه ابزار نوشته شده توسط OWASP هست که میتوانه توی این مورد کمک کنه، درمورداش صحبت خواهیم کرد .

Favicon Hashes: برخی سرویس های مثلا OWA یا همون Outlook Web App که واسه ایمیل استفاده میشه یه لوگویی دارن که توی تصویر زیر مثالش رو میبینید . توی Shodan امکان این وجود داره که شما بتونید از طریق کد این لوگو جستجو کنید .

The screenshot shows a Shodan search interface with the query 'OWA http favicon hash:442749392'. It displays three results, each showing a thumbnail of the favicon and detailed information about the SSL certificate.

- Result 1:** IP 186.67.25.228, webmail.anticipa.cl, mail.anticipa.cl, amazonas.anticipa.cl, autodiscover.anticipa.cl, ENTEL CHILE S.A., Chile, Santiago. The SSL Certificate details include Issued By: Sectigo RSA Domain Validation Secure Server CA, Common Name: Sectigo RSA Domain Validation, Organization: Sectigo Limited, and Issued To: Sectigo Limited. Headers shown: HTTP/1.1 200 OK, Cache-Control: no-cache, no-store, Pragma: no-cache, Content-Type: text/html; charset=utf-8, Expires: -1, Set-Cookie: OutlookSession=ca4cb5f90e934040bb0a6, X-OWA-Version: 14.3.513.0, X-Powered-By: ASP.NET, Date: Thu, 30 Nov 2023 16:57:42 GMT.
- Result 2:** IP 177.177.33.229, symmox.co.uk, goodhealth.dl/29, United Kingdom, Basingstoke. The SSL Certificate details include Issued By: RapidSSL TLS RSA CA (2), Common Name: symmox.co.uk, Organization: Digicert Inc, and Issued To: symmox.co.uk. Headers shown: Access Granted: Want to get more out of your ex!, SSL.Certificate, Supported SSL Versions: TLSv1, TLSv1.1, TLSv1.2.
- Result 3:** IP 171.220.66, www.mail.unsulinesquebec.com, mail.unsulinesquebec.com, Videotron Ltee, Canada, Laval. The SSL Certificate details include Issued By: Go Daddy Secure Certif Authority - 02, Common Name: www.mail.unsulinesquebec.ca, Organization: GoDaddy.com, Inc., and Issued To: www.mail.unsulinesquebec.ca. Headers shown: SSL.Certificate, Supported SSL Versions: TLSv1, TLSv1.1, TLSv1.2.

بعد Shodan به شما تمام رکورد هایی که این لوگو رو دارند نشون میده و میتوانید دنبال تارگت خودتون بگردید و ادرس IP ها رو پیدا کنید . البته این مورد کم پیش میاد ولی خب شاید یه روزی یه جایی بکارتون بیاد .

Misconfiguration Subdomains . ۵ هاست . یک شرکت میتوانه چندین وеб اپلیکیشن مختلف داشته باشه و هر یه وеб اپلیکیشن میتوانه چندین Subdomain رو دارا باشه . از طریق پیدا کردن اون Subdomain ها و بررسی کردن اونها احتمال داره که بتونید به Real IP Address یا CIDR برسید . حالا چطوری چنین چیزی ممکن است ؟ احتمال دارد که تمام Subdomain های یک وеб اپلیکیشن پشت ابر نباشد و شما با پیدا کردن اون بتونید به Real IP Address برسید .

یکی از وبسایت هایی که میتوانید ازش استفاده کنید تا subdomainfinder.c99.nl هست و همچنین میتوانید از ابزار Amass هم استفاده کنید . این روش خیلی اوقات جواب میده، مخصوصا قسمت مربوط به DNS Record ها که احتمالش زیاده یه جایی یک پیکر بندی درست انجام نشده باشه . اینها روش هایی هستند که واسه پیدا کردن IP Address پشت ابر استفاده میشن و اگه از این روشها نشد IP رو پیدا کرد به نظرم دیگه نمیشه کاریش کرد XD .

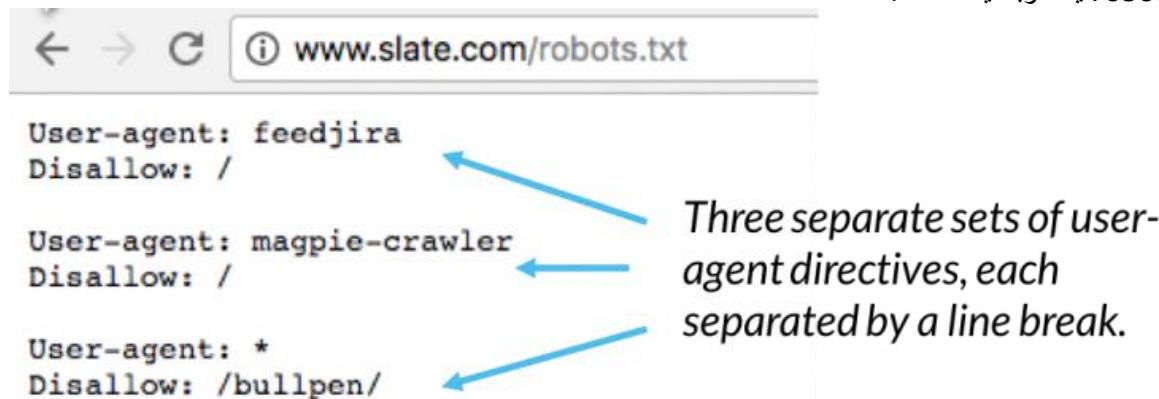
: گاهی پیش میاد که ما یک Real IP Address را پیدا نمیکنیم این IP Address متعلق به کدام یک از دامنه های تارگتمن هست و ما مجموعه ای از دامنه ها را داریم . مثل 5.247.100.97 رو پیدا کردیم و دامنه های example.com, mail.example.com, srv.example.com, meet.example.com, off.example.com, ... بتونیم بفهمیم کدام یک از این دامنه ها را رو میتوانیم روی IP Address پیدا شده پیدا کنیم باید بیاریم Host Header درخواستی که میره به سمت سرور رو هر دفعه یکی از این دامنه ها قرار بدیم .

Request

Pretty	Raw	Hex
1 GET / HTTP/2		
2 Host: example.com		
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.88 Safari/537.36		
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
5 Accept-Encoding: gzip, deflate, br		
6 Accept-Language: en-US,en;q=0.9		
7		
8		

و هر دفعه درخواست رو ارسال کنیم تا ببینیم کدام جواب میده . اگه تعداد دامنه هامون زیاد باشه اینکار به صورت دستی واقعا سخت میشه و اسه همین میتوانیم از ابزار هایی مثل GoBuster, Amass استفاده کنیم .

/چیست ؟ یک فایلی وجود داره توی همه وبسایت های به نام robots.txt که توی دایرکتوری اصلی وبسایت قرار داره . این فایل مشخص میکنه که چه مسیر هایی توسط Bot ها قابل دسترسی یا غیر قابل دسترسی باشند و همچنین میشه توش مشخص کرد که یک Bot با یک User-Agent خاص اجازه دسترسی یا عدم اجازه دسترسی به چه مسیر های داشته باشد . برای ارتباط با Crawler ها استفاده میشه . تصویر زیر robots.txt یک وبسایت است :



در دو خط اول گفته که feedjira هر رباتی که هیچ مسیری رو از / به بعد Crawl کنه و همچنین در دو خط بعدی هم درمورد رباتهایی با User-agent magpie-crawler همین رو گفته .

اما در دو خط آخر گفته که هر رباتی با User-agent magpie-crawler که مسیر هایی که با /bullepen/ شروع میشن Crawl کنه ولی بقیه مسیر ها رو میتوانه . این ساختار یک فایل robots.txt هست و البته دستور Allow هم داره که میتوانیم مشخص کنیم چه رباتهایی با چه User-agent هایی اجازه دارند چه مسیر هایی رو Crawl کنن . در برخی موارد یک گزینه هم وجود داره به نام Sitemap که به Crawler ها میگه، برای Crawl کردن سایت میتوانید این لینکها رو ببینید . تصویر زیر فایل robots.txt سایت دیجیکالا هست و میتوانید ته این فایل Sitemap رو ببینید .

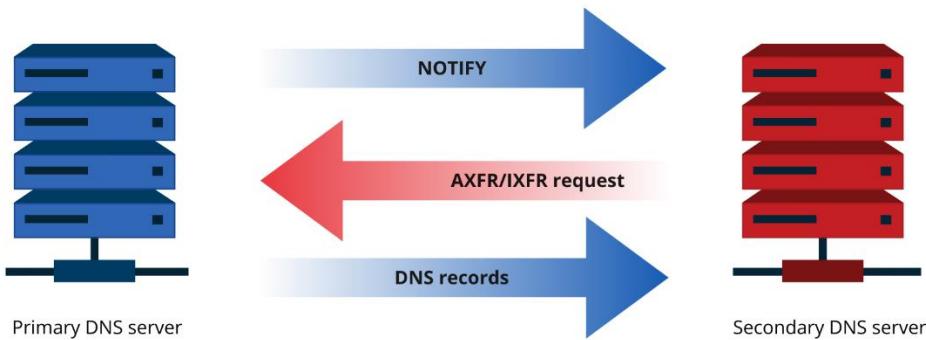
```
Allow: */mag/*.js
Allow: */mag/*.css
Allow: *?x-oss-process=image/*
```

Sitemap: https://www.digikala.com/sitemap.xml

خط این فایل چه کاربردی واسه ما که میخوایم باگ پیدا کنیم داره ؟ برخی اوقات برنامه نویسها مسیر های حساس سایت خودشون رو واسه اینکه توسط رباتها Crawl نشه میان و توی این فایل Disallow میکنن و ما میتوانیم اونها رو بdest بیاریم و ممکن هست که به اطلاعات جالبی دست پیدا کنیم .

یک نوع DNS Transaction هست یعنی به عملی که روی DNS Record های یک DNS Server انجام میشے و یکی از چندین مکانیزم موجود برای Administrator هاست که از طریق اون میتوونن DNS Database هاشون رو توی چندتا DNS Server دیگه Sync کنن . حالا چرا Administrator ها باید چنین کنن ؟ به خاطر اینکه اگه یکی از DNS Server ها از کار افتاد، های دیگه بتونن Resolution رو بدون مشکل انجام بده .

DNS Zone Transfer



برای انجام شدن Authentication هیچ DNS Zone Transfer نیاز نیست و خبر بد این هست که هر کسی که تظاهر کنه که یک کلاینت هست میتوونه یه کپی از یک Zone از DNS رو بگیره که میتوونه شامل Subdomain ها و ... باشه . برای اینکه چنین انفاقی نیفته دو راه حل وجود داره :

1. غیر فعال کردن قابلیت DNS Zone Transfer

2. تعریف کردن لیستی از IP Address های خاص که میتوونن DNS Zone Transfer رو انجام بدن و بقیه نتونن سایت یک پروژه ایجاد کرد که اسیب پذیری DNS Zone Transfer داره که میتوانید جزئیات این پروژه تو ادرس <https://digi.ninja> ببینید . یک تارگت داریم به ادرس <https://zonetransfer.me> که در تصویر زیر آمدیم و با dig اون رو بدست اوردیم :

```
kali㉿kali:~$ dig zonetransfer.me
; <>> DiG 9.19.17-1-Debian <>> zonetransfer.me
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 4150
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: MBZ: 0x0005, udp: 1232
;; COOKIE: c0a0181565d4df36010000006569c67e3faaf23d4ce6c9e (good)
;; QUESTION SECTION:
;zonetransfer.me.      IN      A
;;
;; ANSWER SECTION:
zonetransfer.me.      5       IN      A      5.196.105.14
;;
;; Query time: 843 msec
;; SERVER: 192.168.89.2#53(192.168.89.2) (UDP)
;; WHEN: Fri Dec 01 06:41:47 EST 2023
;; MSG SIZE  rcvd: 88
```

حمله DNS Zone Transfer به Name Server ها هست و خب ما باید این NS دستور dig و دستور NS این رکورد ها رو گرفتیم : بیاریم که در تصویر زیر میبینید با

```
kali㉿kali:~$ dig ns zonetransfer.me
; <>> DiG 9.19.17-1-Debian <>> ns zonetransfer.me
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 50728
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 1232
; COOKIE: c12589b9dc9485b7010000006569c6eb123bca10c1bdd9d7 (good)
;; QUESTION SECTION:
;zonetransfer.me.      IN      NS

;; ANSWER SECTION:
zonetransfer.me.      5       IN      NS      nsztm1.digi.ninja.
zonetransfer.me.      5       IN      NS      nsztm2.digi.ninja.

;; Query time: 799 msec
;; SERVER: 192.168.89.2#53(192.168.89.2) (UDP)
;; WHEN: Fri Dec 01 06:43:35 EST 2023
;; MSG SIZE rcvd: 124
```

میبینید که دو NS Record رو واسه ما برگرداند و حالا میایم از طریق dig و دستور AXFR به یکی از اینها میگیم که تمام Record های مربوط به ادرس (Zone) zonetransfer.me را بخواهیم.

```
kali㉿kali:~$ dig AXFR zonetransfer.me @nsztm1.digi.ninja
; <>> DiG 9.19.17-1-Debian <>> AXFR zonetransfer.me @nsztm1.digi.ninja
;; global options: +cmd
zonetransfer.me.    7200   IN      SOA     nsztm1.digi.ninja. robin.digi.ninja. 2019100801 172800 900 1209600 3600
zonetransfer.me.    300    IN      HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.    301    IN      TXT     "google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA"
zonetransfer.me.    7200   IN      MX      0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200   IN      MX      10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200   IN      MX      10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200   IN      MX      20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.    7200   IN      MX      20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.    7200   IN      MX      20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.    7200   IN      MX      20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me.    7200   IN      A       5.196.105.14
zonetransfer.me.    7200   IN      NS      nsztm1.digi.ninja.
zonetransfer.me.    7200   IN      NS      nsztm2.digi.ninja.
_acme-challenge.zonetransfer.me. 301 IN  TXT   "60a05hbUJ9xSsvYy7pApQwvCUSSGgxvrbdizjePEsZI"
_sip._tcp.zonetransfer.me. 14000 IN SRV   0 0 5060 www.zonetransfer.me.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me. 7200 IN PTR   www.zonetransfer.me.
asfdbauthdns.zonetransfer.me. 7900 IN AFSDB  1 asfdbbox.zonetransfer.me.
asfdbbbox.zonetransfer.me. 7200 IN A       127.0.0.1
asfdbvolume.zonetransfer.me. 7800 IN AFSDB  1 asfdbbox.zonetransfer.me.
canberra-office.zonetransfer.me. 7200 IN A       202.14.81.230
cmdexec.zonetransfer.me. 300 IN  TXT   "; ls"
contact.zonetransfer.me. 2592000 IN TXT   "Remember to call or email Pippa on +44 123 4567890 or pippa@zonetransfer.me when r"
dc-office.zonetransfer.me. 7200 IN A       143.228.181.132
deadbeef.zonetransfer.me. 7201 IN AAAA   dead:beaf::
dr.zonetransfer.me. 300 IN  LOC   53 20 56.558 N 1 38 33.526 W 0.00m 1m 10000m 10m
DZC.zonetransfer.me. 7200 IN  TXT   "AbCdEfG"
email.zonetransfer.me. 2222 IN  NAPTR  1 1 "P" "E2U+email" "" email.zonetransfer.me.zonetransfer.me.
email.zonetransfer.me. 7200 IN A       74.125.206.26
Hello.zonetransfer.me. 7200 IN  TXT   "Hi to Josh and all his class"
home.zonetransfer.me. 7200 IN A       127.0.0.1
Info.zonetransfer.me. 7200 IN  TXT   "ZoneTransfer.me service provided by Robin Wood - robin@digi.ninja. See http://dig
internal.zonetransfer.me. 300 IN NS      intns1.zonetransfer.me.
internal.zonetransfer.me. 300 IN NS      intns2.zonetransfer.me.
intns1.zonetransfer.me. 300 IN A       81.4.108.41
intns2.zonetransfer.me. 300 IN A       167.88.42.94
office.zonetransfer.me. 7200 IN A       4.23.39.254
ipv6actnow.org.zonetransfer.me. 7200 IN AAAA   2001:67c:2e8:11::c100:1332
www.zonetransfer.me. 7200 IN A       207.46.197.32
```

همه رو برگرداند و خب چون اسیب پذیر بود . از این کار میتوانیم واسه بدست اوردن Subdomain ها و Real IP Address ها استفاده کنیم . دستور AXFR ابزار dig واسه DNS Server را برگرداند . حالا اگه اسیب پذیر نبود چی ؟ اگر اسیب پذیر نبود پیامی به شکل زیر دریافت میکنیم :

```
kali㉿kali:~$ dig ns google.com +short
ns1.google.com.
ns3.google.com.
ns2.google.com.
ns4.google.com.

kali㉿kali:~$ dig AXFR google.com @ns1.google.com
; <>> DiG 9.19.17-1-Debian <>> AXFR google.com @ns1.google.com
;; global options: +cmd
google.com.          60      IN      SOA     ns1.google.com. dns-admin.google.com. 586618072 900 900 1800 60
; Transfer failed.
```

توى ويندوز واسه اينکه بتونيم اين حمله رو انجام بدید ميتوانيد از ابزار nslookup استفاده کنیم :

```
> set type=ns
> zonetransfer.me
Server: Unknown
Address: 192.168.1.1

Non-authoritative answer:
zonetransfer.me nameserver = nsztm1.digi.ninja
zonetransfer.me nameserver = nsztm2.digi.ninja

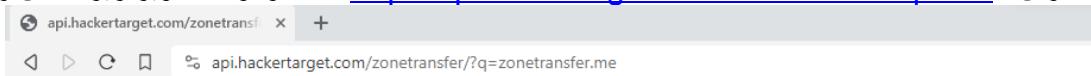
nsztm1.digi.ninja      internet address = 81.4.108.41
nsztm2.digi.ninja      internet address = 34.225.33.2
> server nsztm1.digi.ninja
Default Server: nsztm1.digi.ninja
Address: 81.4.108.41

> ls -d zonetransfer.me
[nsztm1.digi.ninja]
zonetransfer.me.          SOA    nsztm1.digi.ninja robin.digi.ninja. (2019100801 172800 900 1209600 3600)
zonetransfer.me.          HINFO   Casio fx-700G Windows XP
zonetransfer.me.          TXT    "google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA"

zonetransfer.me.          MX     0     ASPMX.L.GOOGLE.COM
zonetransfer.me.          MX     10    ALT1.ASPMX.L.GOOGLE.COM
zonetransfer.me.          MX     10    ALT2.ASPMX.L.GOOGLE.COM
zonetransfer.me.          MX     20    ASPMX2.GOOGLEMAIL.COM
zonetransfer.me.          MX     20    ASPMX3.GOOGLEMAIL.COM
zonetransfer.me.          MX     20    ASPMX4.GOOGLEMAIL.COM
zonetransfer.me.          MX     20    ASPMX5.GOOGLEMAIL.COM
zonetransfer.me.          A      5.196.105.14
zonetransfer.me.          NS     nsztm1.digi.ninja
zonetransfer.me.          NS     nsztm2.digi.ninja
_acme-challenge          TXT    "60a05hbUJ9xSsvYy7pApQvwCUSSGgxvrbdzjePEsZI"

_sip._tcp                 SRV    priority=0, weight=0, port=5060, www.zonetransfer.me
14.105.196.5.IN-ADDR.ARPA PTR    www.zonetransfer.me
asfdbauthdns              AFSDB  1     asfdbbox.zonetransfer.me
asfdbbox                  A      127.0.0.1
asfdbvolume               AFSDB  1     asfdbbox.zonetransfer.me
capbonra-office           A      202.111.81.230
```

همچنين يك ابزاری هم <https://hackertarget.com> به صورت انلاین قرار داده که میشه به صورت Passive این کار رو باهش انجام داد، این ابزار توى ادرس <<https://api.hackertarget.com/zonetransfer/?q=<url>>> را میبینید :



```
; <>> DiG 9.10.3-P4-Debian <>> axfr @nsztm1.digi.ninja zonetransfer.me
; (1 server found)
;; global options: +cmd
zonetransfer.me.          7200  IN      SOA    nsztm1.digi.ninja. robin.digi.ninja. 2019100801 172800 900 1209600 3600
zonetransfer.me.          300   IN      HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.          301   IN      TXT    "google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA"
zonetransfer.me.          7200  IN      MX     0     ASPMX.L.GOOGLE.COM
zonetransfer.me.          7200  IN      MX     10    ALT1.ASPMX.L.GOOGLE.COM
zonetransfer.me.          7200  IN      MX     10    ALT2.ASPMX.L.GOOGLE.COM
zonetransfer.me.          7200  IN      MX     20    ASPMX2.GOOGLEMAIL.COM
zonetransfer.me.          7200  IN      MX     20    ASPMX3.GOOGLEMAIL.COM
zonetransfer.me.          7200  IN      MX     20    ASPMX4.GOOGLEMAIL.COM
zonetransfer.me.          7200  IN      MX     20    ASPMX5.GOOGLEMAIL.COM
zonetransfer.me.          7200  IN      MX     20    ASPMX4.GOOGLEMAIL.COM
zonetransfer.me.          7200  IN      MX     20    ASPMX5.GOOGLEMAIL.COM
zonetransfer.me.          7200  IN      A      5.196.105.14
zonetransfer.me.          7200  IN      NS     nsztm1.digi.ninja.
zonetransfer.me.          7200  IN      NS     nsztm2.digi.ninja.
_acme-challenge.zonetransfer.me. 301 IN  TXT    "60a05hbUJ9xSsvYy7pApQvwCUSSGgxvrbdzjePEsZI"
_sip._tcp.zonetransfer.me. 14000 IN  SRV    0 0 5060 www.zonetransfer.me.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me. 7200 IN  PTR    www.zonetransfer.me.
asfdbauthdns.zonetransfer.me. 7900 IN  AFSDB  1     asfdbbox.zonetransfer.me.
asfdbbox.zonetransfer.me. 7200  IN      A      127.0.0.1
```

توی متابولویت هم یک مازول و اسه اینکار وجود داره به نام auxiliary/gather/enum_dns که میتونیم استفاده کنیم :

```
msf6 > use auxiliary/gather/enum_dns
msf6 auxiliary(gather/enum_dns) > set domain zonetransfer.me
domain => zonetransfer.me
msf6 auxiliary(gather/enum_dns) > exploit

[*] Querying DNS NS records for zonetransfer.me
[+] zonetransfer.me NS: nsztm1.digi.ninja
[+] zonetransfer.me NS: nsztm2.digi.ninja
[*] Attempting DNS AXFR for zonetransfer.me from 81.4.108.41
W, [2023-12-01T06:58:27.486278 #1631]  WARN -- : Failed to parse RR packet from offset: 634
W, [2023-12-01T06:58:27.486840 #1631]  WARN -- : Failed to parse RR packet from offset: 703
W, [2023-12-01T06:58:27.488235 #1631]  WARN -- : Failed to parse RR packet from offset: 995
W, [2023-12-01T06:58:27.488644 #1631]  WARN -- : Failed to parse RR packet from offset: 1050
W, [2023-12-01T06:58:27.490475 #1631]  WARN -- : Failed to parse RR packet from offset: 1581
W, [2023-12-01T06:58:27.490784 #1631]  WARN -- : Failed to parse RR packet from offset: 1646
[+] zonetransfer.me Zone Transfer:
;; Answer received from 81.4.108.41:53 (2046 bytes)
;;
;; HEADER SECTION
;; id = 22135
;; qr = 1      opCode: QUERY    aa = 1   tc = 0   rd = 0
;; ra = 0      ad = 0   cd = 0   rcode = NoError
;; qdCount = 1  anCount = 50    nsCount = 0     arCount = 0

;; QUESTION SECTION (1 record):
;; zonetransfer.me.          IN      AXFR

;; ANSWER SECTION (50 records):
zonetransfer.me.    7200  IN      SOA    nsztm1.digi.ninja. robin.digi.ninja. 2019100801 172800 900 1209600
zonetransfer.me.    300   IN      HINFO  Casio fx-700G
Windows XP®
zonetransfer.me.    301   IN      TXT
zonetransfer.me.    7200  IN      MX     0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200  IN      MX     10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200  IN      MX     10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200  IN      MX     20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.    7200  IN      MX     20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.    7200  IN      MX     20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.    7200  IN      MX     20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me.    7200  IN      A      5.196.105.14
zonetransfer.me.    7200  IN      NS    nsztm1_digi_ninja
```

تکنولوژی های قرن 21 😊 ؟ میتونم بگم که یکی از معضلات کسانی که میخوان توی حوزه Bug-Bounty و تست نفوذ وب کار کنن همین تکنولوژی های قرن 21 وب هستند . تکنولوژی هایی مثل ... Webpack, Angular JS, Vue JS, React JS در سمت کلاینت و همچنین فریمورک هایی مثل ... Laravel, Ruby On Rails, Django, Flask, ... در سمت سرور . با وجود امدن این تکنولوژی ها میشه گفت که برخی از حفرات امنیتی کلا برکنار شدن و خب قاعده ایک سری Attack Surface های جدید هم بوجود اومد .



در گذشته کدهای جاواسکریپت سمت کلاینت به صورت کاملا خام و بدون تغییرات اعمال میشدن و میشد از طریق Review کردنشون به حفرات امنیتی دست پیدا کرد اما در حال حاضر با بوجود امدن تکنولوژی هایی مثل ... Webpack, Vue JS, Angular JS, React JS, ... در

سمت کلاینت، خوندن کدهای جاواسکریپت به شدت دشوار شده است و تقریباً میشه گفت که Webpack کلی از Scanner هارو بازنشست کرد و حقوق بازنشستگی هم به خاطر کمبود بودجه نتوانست بده.

همچنین کدهای سمت سرور هم تغییرات زیادی داشت، در گذشته یک وب سایت با زبان PHP کدهای خامی داشت و فقط توسط Interpreter تفسیر میشد و نتیجه رو وب سرور به کلاینت ارسال میکرد ولی هم اکنون ما تعداد بسیار زیادی Framework مختلف سمت سرور داریم مثل ... در زبان PHP Laravel, Symfony, CakePHP, CodeIgniter, ... در زبان Python Django, Flask فریمورک هایی مثل Django, Flask در زبان Ruby on Rails در زبان رویی و همچنین ... در سمت سرور اوامندن و فریمورک ها رو پایتون و Spring, Express, ... در این روش هایی که در اینجا بررسی شد، از یک پلتفرم متناسب با این فریمورک استفاده میشود. در اینجا میخواهیم در مورد امنیتی این فریمورک ها توضیح دهیم.

خدا چیه اینا؟ اصن چرا؟ قبله اینطوری نبود، میکردن از تگهای HTML استفاده میکردن مث ادمی زاد اجرا میشد ولی بینید، متغیر ها هم مث ادم نامگذاری نشدن که اصن درکشون کنی. مثلا کد زیر نمونه یک Single Page Application هست، اومند پدر متغیر هارو در اون دن، کجه و کوله ناه گذاه، که اون اصن که حه؟

۱ | !function() {
۲ | //...
۳ | }();

```
▼ cloud lingohub.com
- "use strict";
- function e(e) {
```

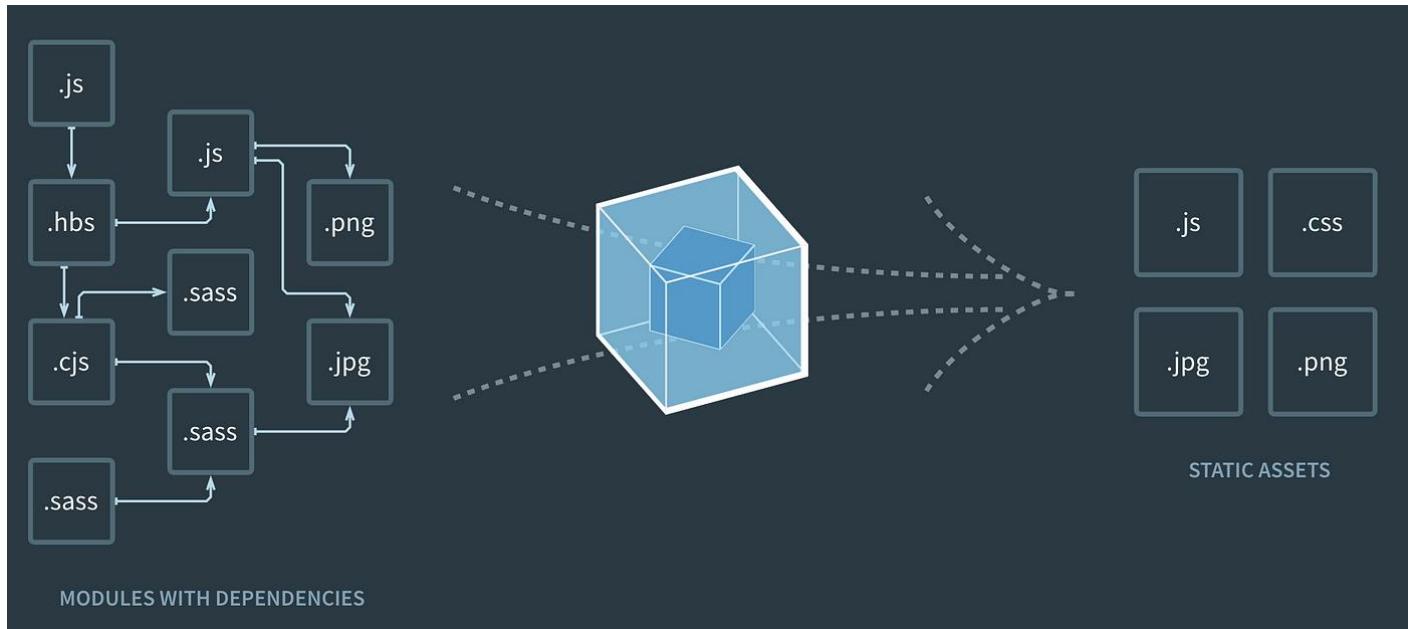
```
try {
  if ("undefined" === typeof console)
```

```
chunks
  pages
    242-2811e8962a744f89.js
    878-e7c4ca790db6eb78.js
    framework-3bd9ba481ce70ce9.js
    main-40861dfeb95e053e.js
    webpack-c17edaf63ccc8abe.js
css
media
rP-7XRnWiTi2JbJak4XSn
cdn-cgi/scripts
  5c5dd728/cloudflare-static
    email-decode.min.js
  7d0fa10a/cloudflare-static
images
?ref=mademith/vuejs.com

  if ( undefined == typeof console)
    return;
  "error"in console ? console.error(e) : console.log(e)
} catch (e) {}
}
function t(e) {
  return d.innerHTML = '<a href="' + e.replace(/\//g, """) + '">/a>',
  d.childNodes[0].getAttribute("href") || ""
}
function r(e, t) {
  var r = e.substr(t, 2);
  return parseInt(r, 16)
}
function n(n, c) {
  for (var o = "", a = r(n, c), i = c + 2; i < n.length; i += 2) {
    var l = r(n, i) ^ a;
    o += String.fromCharCode(l)
  }
  try {
    o = decodeURIComponent(escape(o))
  } catch (u) {
    e(u)
  }
}
```

خب دیگه و اسه امنیت بیشتره و خب درست هم پیش رفتن .

چه که اینقدر ازش می‌نال؟ Webpack یک بسته بندی کننده مازلول‌ها و اسه جاوا‌اسکریپت هست و در ابتدای فقط کدهای جاوا‌اسکریپت رو بسته بندی می‌کرد و الان Asset های سمت کاربر مثل ... HTML, CSS, Image, ... رو هم بسته بندی می‌کنه. حالا بسته بندی یعنی چی؟ یعنی اینکه فرض کنید مجموعه‌ای از کدها و Asset ها داریم که کاری رو انجام میدن و آگه نگاشون کنی راحت می‌فهمی چیکار می‌کنن چون کداشون قابل خوندن هست، Webpack می‌داد اونها رو تبدیل می‌کنه به یک بسته که همون کار رو انجام میده ولی فهمی اینکه چطوری اون کار انجام می‌شه سخت می‌شه یعنی می‌داد **Minify** و **Obfuscation** همچنین **Bundle** رو اعمال می‌کنه و در نهایت به شما فایلهایی میده که بپوشون می‌گن.

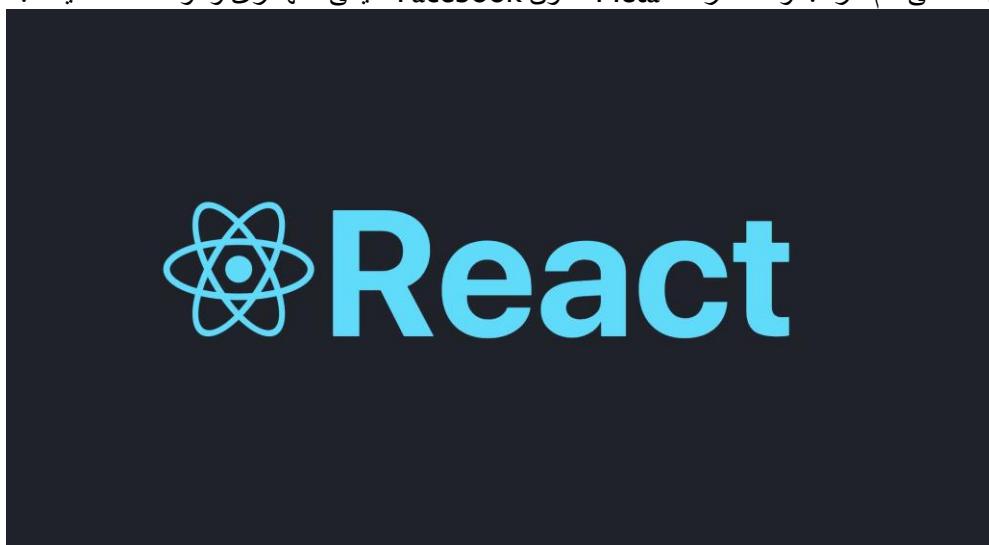


Vue JS چیست؟ یک کتابخانه جاوا اسکریپتی هست که بر اساس یک **Design Pattern** به نام **Model-View-Viewmodel** است که در این طراحی Front-End Web اپلیکیشن‌ها به صورت **Single Page Application** استفاده می‌شوند. این طراحی امروزه سرعت روبرو باشد و مابه عنوان هکر (ترجیحاً کلا سفید) می‌باشد. طرز رفتار تکنولوژی‌های این اپلیکیشن‌ها را بدون نیاز به Flow برای ما اهمیت داره، پس پیشنهادم اینه که لازم نیست توی این تکنولوژی‌ها عمیق بشیم ولی حداقل اگاهی نسبت به چگونگی رفتارشون داشته باشیم. همراه خودش اسیب پذیری‌هایی هم اورد مثل **Client Side Template Injection** که در اینده باهش آشنا خواهیم شد.



Angular JS چیست؟ ایشون یک فریمورک جاوا اسکریپتی جهت طراحی **Single Page Application** هاست به مانند Vue.js. البته این پروژه متوقف شده ولی ممکنه که گاهی جایی ببینیم.

چیست؟ ایشون هم به مانند React JS، Angular JS، Vue JS یک کتابخانه جاوا اسکریپتی هست، که واسه طراحی رابط کاربری وب اپلیکیشن ها، اندروید اپلیکیشن ها و حتی دسکتاپ اپلیکیشن ها استفاده میشه. این کتابخونه بر اساس Component ها رابط کاربری رو طراحی میکنه و Flow خاصی هم داره. توسط شرکت Facebook همون قدمی نگهداری و توسعه داده میشه.



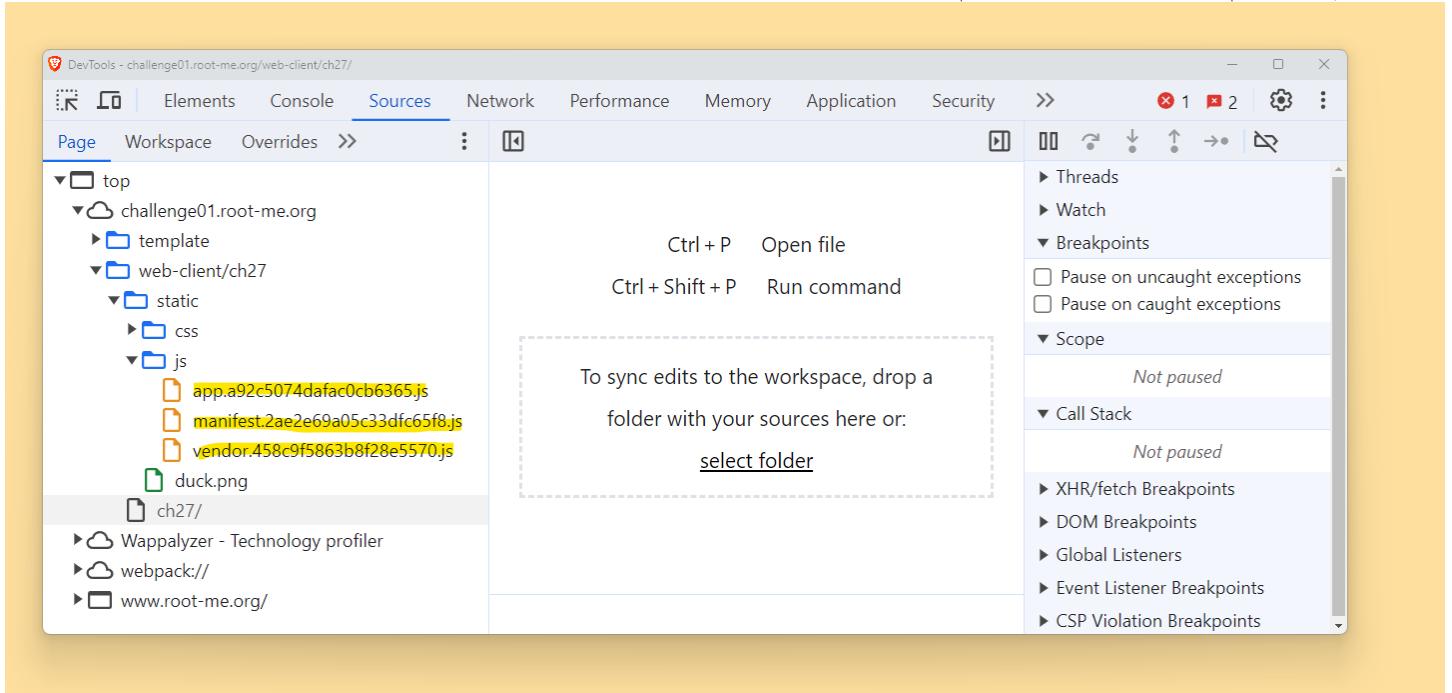
Source Map ها چیستند؟ امروزه با این تکنولوژی های جدیدی که وجود داره و بسیار زیاد هم استفاده میشن مثل گذشته نیست، که کدهای سمت کلاینت (Front-End) همان طوری باشه که توسط برنامه نویس نوشته شده اند. این کدها سمت برنامه نویس کاملاً باز و قابل خوندن هستند چرا که خودش نوشته ولی وقتی سمت کاربر که میان دمشون رو میگیرن بالا و توسط Webpack یا ... Vit و Bundle میشن و تبدیل میشن به یه مشت چرت و پرت یعنی Minify و Obfuscation میشن. اما یک فایلهایی وجود داره به نام Source Map که یه طور ای کلید خوندن فایل های Bundle شده هستند و اطلاعاتی توشون هست که برای Debugging استفاده میشه یعنی گاهی اوقات اطلاعات حساسی هستند. این فایلها باید به سمت کاربر ارسال بشه وقتی قرار هست که پروژه به صورت Deploy در بیاد چرا که ممکن هست داده های توشون باشه که حساس باشن. اما گاهی اوقات ممکنه که برنامه نویس تارگت ما، از این موضوع اگاهی نداشته باشه و این فایلها رو به سمت کاربر فرستاده باشه و از قضا اطلاعات حساسی هم توشون وجود داشته باشه، پس بهتره که ما به عنوان یک هکر این مورد رو روی تارگت های خودمون بررسی کنیم. توی وب سایت <https://root-me.org> یک چالشی وجود داره که همین موضوع رو نشون داده. ادرس این چالش <https://www.root-me.org/en/Challenges/Web-Client/Javascript-Webpack> هست و میتوانید اون رو حل کنید و خب من هم همین پایین درموردش توضیح میدم. بگم که چالش های root-me به صورت پیدا کردن عبارتی به نام flag هست که باید اون رو پیدا و تایید کنیم تا چالش حل بشه و این flag هر جایی میتوونه باشه، بریم دنبال Flag :

وقتی Start The Challenge میکنید صفحه زیر رو خواهید دید.

Quack Quack ! | Duck or Mandarin duck

This a normal duck !
It is just a duck... You can eat this one.

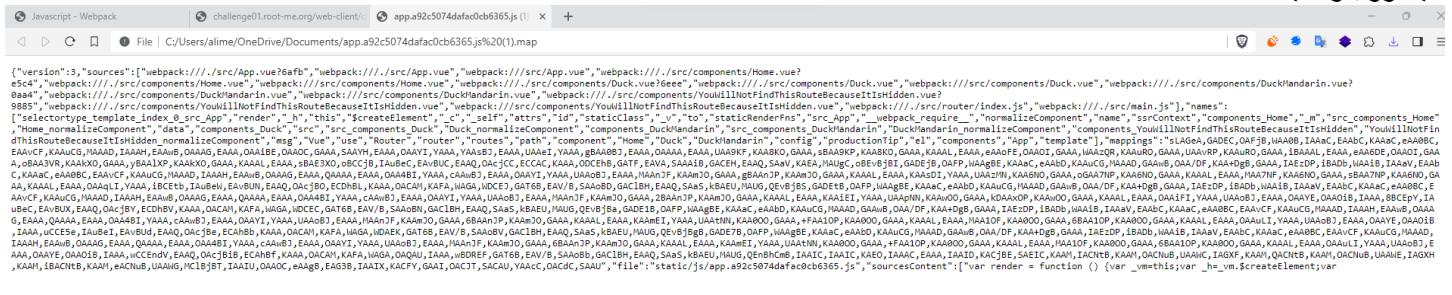
یک وبسایت خیلی ساده که تو ش درمورد دو نوع اردک صحبت شده و تفاوت این دو رو گفته . از اونجا که این چالش درمورد-**Javascript** هست پس باید قاعدتا به دنبال فایلهایی باشیم که به جاواسکریپت و **Webpack** مرتبط هستند . با فشردن F12 به صفحه **Front-End Source** میتوانیم فایلهایی که مرورگر میریم و توی تب **DevTools** جاواسکریپت بگردیم که به سه فایل زیر خواهیم رسید :



اگه این فایلهای را باز کنیم به جز چرت و پرتهایی به زبان جاواسکریپت چیزی نخواهیم یافت و واسه اطمینان واژه **flag** رو تو شون سرج کنید ولی چیزی پیدا نمیکنیم چون این چالش درمورد **Source Map** هاست نه خود فایلهای . حالا چطوری به **Source Map** دست پیدا کنیم ؟ کافیه که به ته نام هر فایل **.map** اضافه کنیم و اون رو باز کنیم به شکل زیر :

challenge01.root-me.org/web-client/ch27/static/js/app.a92c5074dafac0cb6365.js.map

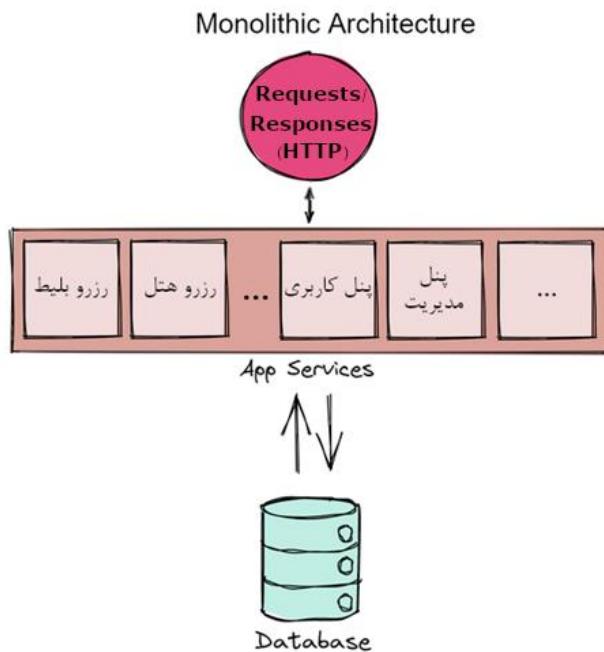
فایل رو باز کنید :



یه چنین چیزی به ظاهر چرت و پرت تر فایل اصلی خواهیم دید . حالا داخل این فایل دنبال واژه **flag** بگردید .

```
Here is you flag: xxxxxxxxxxxxxxxxxxxxxxxxx
Barat Jلوی عبارت Here is you flag: درواقع flag موردنظر ماست و البته عبارت جلوی Here is you flag: xxxxxxxxxxxxxxxxxxxxxxxxx
. اگه توی فایل js نگاه میکریدم چنین چیزی نبود ولی توی توی js.map شون داده شد . حالا توی این چالش چون Capture The Flag هست و باید Flag رو پیدا کنیم این بود ولی توی Real World استفاده میکن Source Map ها وجود داشته باشند و حاوی ... Credential, API Token
```

چیست؟ یک مدل **Monolithic Architecture** سنتی برای طراحی نرم افزارها و اپلیکیشن هاست. مفهوم کلمه **Monolithic** به معنی **Composed all in one piece** یعنی همه اجزا در یک کل قرار دارند می باشد. فرض کنید که یک شرکت مسافرتی وجود دارد. این شرکت کارهای مختلفی انجام میده، مثلاً به کلاینت ها اجازه میده که بلیط قطار، هوایپا و اتوبوس بگیرند، همچنین اجازه میده که هتل رزرو کنند و این شرکت میاد و یک اپلیکیشن به صورت **Monolithic Architecture** مینویسه و همه این قابلیت ها رو در قالب یک اپلیکیشن بر روی سرور های خودش **Deploy** میکنه. همه درخواست های کلاینت های وب اپلیکیشن به یک جا ارسال میشوند و فرقی نمیکنه که درخواست چه چیزی را دارند و پاسخ همه درخواست ها هم از یک جا به سمت کلاینت ها ارسال میشه. وب اپلیکیشن ماناما با یک دیتابیس در ارتباط است که این دیتابیس شامل جدولهایی برای هر کدام از سرویس ها می باشد. امروزه معماری **Microservices** در حال پر کردن جای این معماریست ولی باز هم این معماری به شکل گسترده ای استفاده میشه و خوب معايب و مزیت های خودش رو دارد.



این معماری دارای مزایایی می باشد که به عبارت زیر اند :

1. پیچیدگی کم

2. راحتی **Deploy** کردن

اما حال که مزايا رو گفتیم بهتر است چند مورد از معايب اين معماری رو هم بگيم :

1. گسترش دادن اپلیکیشن دشوار است. (سخت بودن **Scalability**)

2. تغییر در اپلیکیشن ناممکن يا در بسیاری اوقات بسیار سخت است. (مثلاً گسترش و تغییر جدول های دیتابیس)

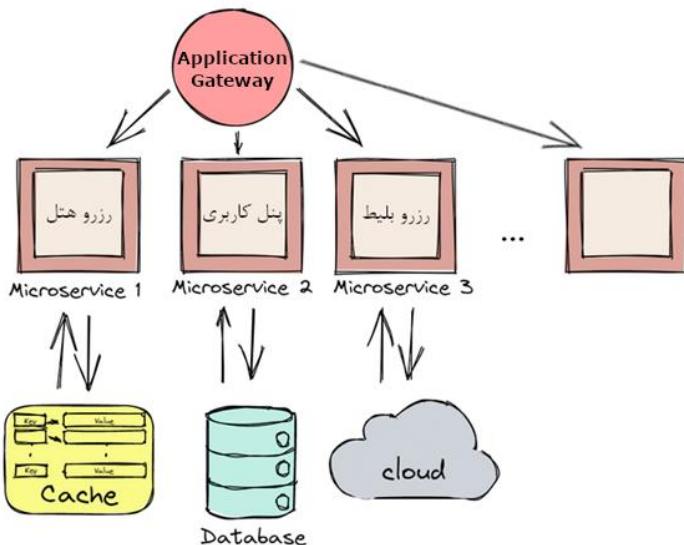
3. عملکرد اپلیکیشن اهسته است.

4. **Single Point of Failure** دارد.

5. ...

چیست؟ معماری مقابل **Monolithic Architecture** هست که توی این معماری یک سری از سرویس های مستقل از هم وجود دارد. هر سرویس دارای منطق خود، پایگاه داده خود و هدف خاص خود هست. بر خلاف **Monolithic Architecture** که همه درخواست های کلاینت ها فارغ از نوع انها به یک نقطه ارسال میشند و همه پاسخ های کلاینت ها هم از یک نقطه برای انها ارسال میشند در این معماری هر سرویس به صورت جداگانه از سرویس دیگه، درخواست های کلاینت ها رو دریافت میکنه و پاسخ های اوونها رو برآشون میفرسته.

Microservices Architecture



خب تصویر بالا رو یه توضیحی بدم، ببینید ما یک **API Gateway** هم باشه، در واقع یه جایی هست که همه درخواست ها بهش ارسال میشه و همه پاسخ ها هم ازش خارج میشه، وقتی یک درخواست به **Application Gateway** برسه، ایشون تشخیص میدن که این درخواست به کدام میکروسرویس تحويل داده بشه و اون درخواست رو به اون میکروسرویس میدن و همینطور درخواست های دیگه هم به همین شکل تحويل میکروسرویس های خودشون خواهند داد و در نهایت پاسخ ها رو از میکروسرویس ها میگیرند و به کلاینت ها ارسال میکنند.

مزایاهایی این معماری داره که به برخی از اونها در زیر اشاره کرده ایم:
1. **Loosely Coupled**: توی کامپیوتر زمانی که زمانیکه کامپوننت ها با هم ارتباطی ندارند و تغییرات توی انها کمترین تاثیر رو بر دیگری میداره میگن **Loosely Coupled** هستند. این معماری هم چنینه.

2. **Agile** و انعطاف پذیری

3. توسعه کاملا مستقل هر میکروسرویس از دیگری

4. **Deploy** کردن مستقل هر میکروسرویس از دیگری

5. خطاهای هر میکروسرویس نسبت به دیگری جداست و تاثیری بر دیگری ندارد.

6. ...

خب معایب این معماری هم چند مورد بگیم. لیست زیر چند مورد از آنهاست:
1. پیچیدگی

2. **Automation**

3. **Debugging**

4. ثبات

5. ...

امروزه کمپانی های بزرگ از معماری **Microservices** زیاد استفاده میکنند و ممکنه هر زیر دامنه و یا **Path** از اونها بر روی یک سرور جداگانه از سرویس های دیگه باشه. از این رو بهتر است که این مورد رو در حین **Penetration Testing** خودمون در نظر بگیریم. در آخر هم بگم که از **Netflix** اولین شرکتی بود که از **Monolithic Architecture** به **Microservices** مهاجرت کرد و به این خاطر یه کارت صد افرین هم گرفت.

اما حالا سوالی که ممکنه پیش بیاد اینه که این **Microservice** هایی که ازشون حرف میزنیم چیا هستند؟ یعنی چی از میکروسرویس ها استفاده میکن و چه کسایی اینا رو در اختیار شرکت ها قرار میدن؟ خب بریم سر وقت اینکه چندین تا از میکروسرویس هایی که زیاد استفاده میشن رو توضیح بدیم و اینکه چطوری میتوانیم این میکروسرویس ها رو پیدا کنیم و چه مشکلات امنیتی برخی از اونها ممکنه داشته باشند؟

یک سیستم **Memory-Caching** چیست؟ **Memcached** چند منظوره هست. این سرویس جهت افزایش سرعت وب سایت **Database-Driven** استفاده میشه و وب سایت های **Database-Driven** و بسایت های **Object** (دادهها) رو توی **RAM** ذخیره میکنه و تعداد دفعاتی که باید دادهها رو از روی یک منبع خارجی (دبایس) بخونیم رو کاهش میده. این سرویس رایگان و **Open-Source** می باشد و روی پورت 11211، به صورت پیش فرض اجرا خواهد شد. فقط بیاد داشته باشید که دادههایی که توسط **Memcached** ذخیره میشوند توی **RAM** هستند و اگه یه وقتی **Server** ریاستارت بشه اون دادهها هم حذف خواهند شد.



توی وبسایت <https://attackdefence.com> که Lab ها و چالش هایی رو در اختیار کاربرانش قرار میده یک مورد هم درمورد **Memcached** وجود داره که اینجا میخوایم با هم حلش کنیم. این چالش توی آدرس <https://attackdefense.com/listing?labtype=linux-security-recon&subtype=recon-cachingserver> قرار داره.

[All Section Labs](#)

Memcached Recon: Basics

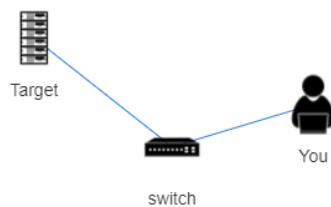
recon-cachingserver | Level: **Easy** | Total Lab Runs: 0 |


Run
Server: US-East

Lab Scoreboard

7153 # Played on AD	0 # Played by you	[0 / 1] # Flags Solved
------------------------	----------------------	-----------------------------

روی دکمه **Run** بزنیم تا اجرا بشه. چالش های این وبسایت هم به مانند **CTF** به صورت **root-me** هست و باید یک **Flag** رو پیدا و مقدار اون رو وارد قسمت جواب کنیم تا چالش حل بشه. توی این چالش ما یک مهاجم هستیم که به یک سوئیچ متصل شده ایم. این سوئیچ در یکی از پورتهاش یک سرور داره که تارگت اصلی ما هست.



وقتی که اجرا شد دکمه **Lab link** رو خواهیم دید، اگه بزنیم روش وارد صفحه ای به شکل زیر خواهیم شد:

```

root@attackdefense:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.0.8 netmask 255.255.0.0 broadcast 10.1.255.255
        ether 02:42:0a:01:00:08 txqueuelen 0 (Ethernet)
        RX packets 755 bytes 56274 (54.9 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 465 bytes 354044 (345.7 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.247.121.2 netmask 255.255.255.0 broadcast 192.247.121.255
        ether 02:42:c0:f7:79:02 txqueuelen 0 (Ethernet)
        RX packets 70477 bytes 3813166 (3.6 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 71993 bytes 4151119 (3.9 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
        RX packets 2038 bytes 87484 (85.4 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 2038 bytes 87484 (85.4 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

حالا که فهمیدیم IP Address ما چی هست میتوانیم از طریق nmap بیایم و کلا شبکه را اسکن کنیم مثل ARP Ping بزنیم و دیوایس هایی که وجود داره توی شبکه Lan رو پیدا کنیم .

```

root@attackdefense:~# nmap -PR 192.247.121.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2023-12-06 07:08 UTC
Nmap scan report for 192-247-121-1.cdma-pool.blue.net (192.247.121.1)
Host is up (0.0000070s latency).
Not shown: 997 closed ports
PORT      STATE     SERVICE
22/tcp    open      ssh
80/tcp    filtered http
443/tcp   filtered https
MAC Address: 02:42:C1:35:8C:14 (Unknown)

Nmap scan report for target-1 (192.247.121.3)
Host is up (0.0000090s latency).
All 1000 scanned ports on target-1 (192.247.121.3) are closed
MAC Address: 02:42:C0:F7:79:03 (Unknown)

Nmap scan report for attackdefense.com (192.247.121.2)
Host is up (0.0000070s latency).
All 1000 scanned ports on attackdefense.com (192.247.121.2) are closed

Nmap done: 256 IP addresses (3 hosts up) scanned in 3.40 seconds
root@attackdefense:~#

```

میبینید که سه تا تارگت برای ما پیدا کرد . ۱۹۲.۲۴۷.۱۲۱.۱ که فک کنم روتر هست و ۱۹۲.۲۴۷.۱۲۱.۲ هم که خودمون هستیم و میمونه ۱۹۲.۲۴۷.۱۲۱.۳ که تارگت اصلی ماست . بیایم پورت مربوط به Memcached رو روش چک کنیم یعنی پورت ۱۱۲۱۱ رو .

```
root@attackdefense:~# nmap -PR 192.247.121.0/24 -p 11211
Starting Nmap 7.70 ( https://nmap.org ) at 2023-12-06 07:11 UTC
Nmap scan report for 192-247-121-1.cdma-pool.blue.net (192.247.121.1)
Host is up (0.000033s latency).

PORT      STATE SERVICE
11211/tcp closed memcache
MAC Address: 02:42:C1:35:8C:14 (Unknown)

Nmap scan report for target-1 (192.247.121.3)
Host is up (0.000020s latency).

PORT      STATE SERVICE
11211/tcp open  memcache
MAC Address: 02:42:C0:F7:79:03 (Unknown)

Nmap scan report for attackdefense.com (192.247.121.2)
Host is up (0.000027s latency).

PORT      STATE SERVICE
11211/tcp closed memcache

Nmap done: 256 IP addresses (3 hosts up) scanned in 2.18 seconds
root@attackdefense:~#
```

میبینید که روی 192.247.121.3 این پورت بازه و داره Memcached را سرویس میده . توی nmap یک اسکریپت وجود داره به نام memcached-info که میاد و اطلاعات مربوط به سرویس Memcached، روی پورت 11211 را تا حد ممکن استخراج میکنه و به ما نشون میده .

```
root@attackdefense:~# nmap -p 11211 --script memcached-info 192.247.121.3
Starting Nmap 7.70 ( https://nmap.org ) at 2023-12-06 07:13 UTC
Nmap scan report for target-1 (192.247.121.3)
Host is up (0.000083s latency).

PORT      STATE SERVICE
11211/tcp open  memcache
|_ memcached-info:
|   Process ID: 8
|   Uptime: 984 seconds
|   Server time: 2023-12-06T07:13:47
|   Architecture: 64 bit
|   Used CPU (user): 0.109263
|   Used CPU (system): 0.050989
|   Current connections: 2
|   Total connections: 7
|   Maximum connections: 2147
|   TCP Port: 11211
|   UDP Port: 0
|_ Authentication: no
MAC Address: 02:42:C0:F7:79:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
root@attackdefense:~#
```

میبینید که بله اطلاعات رو داریم . اون قسمت اخر که نوشته Authentication: no همون جایی هست که بگایی میاره ، بدون هیچگونه احراز هویت، جالبه .

میتوانیم از طریق دستور memcstat که مربوط به Memcached هست (کلا دستورات مربوط به memc با شروع میشن) هم اطلاعاتی رو ببینیم :

```
root@attackdefense:~# memcstat --servers=192.247.121.3
Server: 192.247.121.3 (11211)
  pid: 8
  uptime: 1069
  time: 1701846912
  version: 1.5.12
  libevent: 2.0.21-stable
  pointer_size: 64
  rusage_user: 0.118703
  rusage_system: 0.055394
  max_connections: 2147
  curr_connections: 2
  total_connections: 8
  rejected_connections: 0
  connection_structures: 3
  reserved_fds: 20
  cmd_get: 0
  cmd_set: 10
  cmd_flush: 0
```

حالا وقتی سعی کنیم اگه میشه اطلاعاتی که توی Memory نوسط Memcached ذخیره شدن رو بدست بیاریم . برای اینکار یکی از ابزارها، memcdump هست که برای میشن key های مربوط به دادهها رو Dump میکنه . میتوانیم به شکل زیر استفاده کنیم :

```
root@attackdefense:~# memcdump --servers=192.247.121.3
flag
password
country
zip
state
city
address
nick_name
last_name
first_name
```

میبینید که توی کلید های موجود، کلیدی داریم به نام flag که مقدار داخلش همون جواب مورد نظر ماست. باید سعی کنیم مقدار داخلش را بخونیم. برای اینکار از طریق telnet به سرور 192.247.121.3 پورت 11211 متصل میشیم و دستور [KEY] رو وارد می کنیم و به جای [KEY] کلمه flag رو مینویسیم.

```
root@attackdefense:~# telnet 192.247.121.3 11211
Trying 192.247.121.3...
Connected to 192.247.121.3.
Escape character is '^].
get flag
VALUE flag 0 32
25c8dc1c75c9965dff9af3c8ced2775
END
[]
```

دیدید که مقدار flag رو بیرون کشیدیم. البته به جز این روش میتوانیم بیایم از مازول memcached_extractor توی متاسپلوبیت هم استفاده کنیم که نام کاملش رو زیر میبینید.

auxiliary/gather/memcached_extractor

خب ساده بود و در برخی اوقات به همین سادگیه و بدون Authentication میتوانیم چنین کنیم.

خب بریم سروقت مثل واقعی، شاید بعضیا بیان و بگن نه چنین هم نیست که بدون Authentication چنین شه. برای اینکه بگیم دقیقا چنینه میریم توی سایت shodan.io یا censys.io و دنبال سرور هایی میگردیم که Memcached رو دارند.

The screenshot shows the Censys search interface with the query "services.port=11211". The results page lists two hosts:

- 8.208.87.138** (ALIBABA-CN-NET Alibaba US Technology Co., Ltd. (45102), England, United Kingdom)
 - Ports: 143/IMAP, 11211/MEMCACHED, 6379/REDIS, 1801/MSMQ, 502/MODBUS
 - Services: >_22/SSH, >_21/FTP, >_25/SMTP, >_110/POP3, >_631/IPP
 - Protocols: 5432/POSTGRES, 3306/MYSQL, 1433/MSSQL, 5900/VNC, 427/UNKNOWN
 - Ports: 445/SMB, 102/S7, 9200/ELASTICSEARCH
 - Protocols: 53413/NETIS, 27017/MONGODB, 2404/IEC60870_5_104
- 81.25.126.253** (SWHO-AS swhosting.com (41541), Madrid, Spain)
 - Ports: 22/SSH, 993/IMAP, 2087/HTTP
 - Services: >_53/DNS, >_995/POP3, >_3306/MYSQL
 - Protocols: 80/HTTP, 2082/HTTP, 11211/MEMCACHED
 - Ports: 143/IMAP, 2083/HTTP
 - Protocols: 443/HTTP, 2086/HTTP

خب دومی رو من انتخاب کردم که ادرس IP اون 81.25.126.253 هست و از طریق اسکریپت memcached-info در nmap بررسی کردم و جواب زیر رو گرفتم:

```
root@kali:/home/kali# nmap --script memcached-info 81.25.126.253 -p 11211
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-06 02:41 EST
Nmap scan report for cl2023120518001.dnssw.net (81.25.126.253)
Host is up (0.024s latency).

PORT      STATE SERVICE
11211/tcp open  memcache
| memcached-info:
|   Process ID: 65424
|   Uptime: 33447 seconds
|   Server time: 2023-12-06T07:41:22
|   Architecture: 64 bit
|   Used CPU (user): 0.923289
|   Used CPU (system): 0.409865
|   Current connections: 10
|   Total connections: 19
|   Maximum connections: 1024
|   TCP Port: 11211
|   UDP Port: 11211
|_ Authentication: no

Nmap done: 1 IP address (1 host up) scanned in 1.65 seconds
```

در بسیاری از موارد فایروال میاد و اجازه نمیده که memcached-info اطلاعات رو جمع اوری کنه و خب به شکل زیر نتیجه میده :

```
root@kali:/home/kali# nmap --script memcached-info 8.208.87.138 -p 11211
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-06 02:44 EST
Nmap scan report for 8.208.87.138
Host is up (0.00085s latency).

PORT      STATE SERVICE
11211/tcp filtered memcache

Nmap done: 1 IP address (1 host up) scanned in 1.64 seconds

root@kali:/home/kali# nmap --script memcached-info 117.187.38.86 -p 11211
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-06 02:44 EST
Nmap scan report for 117.187.38.86
Host is up (0.00090s latency).

PORT      STATE SERVICE
11211/tcp filtered memcache

Nmap done: 1 IP address (1 host up) scanned in 4.00 seconds

root@kali:/home/kali# nmap --script memcached-info 149.129.129.147 -p 11211
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-06 02:45 EST
Nmap scan report for 149.129.129.147
Host is up (0.00095s latency).

PORT      STATE SERVICE
11211/tcp filtered memcache

Nmap done: 1 IP address (1 host up) scanned in 0.82 seconds
```

اما این به این معنی نیست که ما نتوانیم از طریق telnet یا هر ابزار دیگه بهش متصل بشیم و دادهای داخلش رو استخراج کنیم. توی تصویر زیر میبینید که شدیم :

```
root@kali:/home/kali# telnet 149.129.129.147 11211
Trying 149.129.129.147...
Connected to 149.129.129.147.
Escape character is '^]'.
```

خب دستور stats را میزنم تا وضعیت سرویس رو بگیرم :

```
root@kali:/home/kali# telnet 8.208.87.138 11211
Trying 8.208.87.138...
Connected to 8.208.87.138.
Escape character is '^]'.
stats
STAT pid 1296
STAT uptime 4119406
STAT time 1502202008
STAT version 1.4.22
STAT libevent 2.0.21-stable
STAT pointer_size 64
STAT rusage_user 90.704000
STAT rusage_system 142.976000
STAT curr_connections 5
STAT total_connections 458196
STAT connection_structures 25
STAT reserved_fds 20
STAT cmd_get 458100
STAT cmd_set 458109
STAT cmd_flush 0
STAT cmd_touch 0
STAT get_hits 430915
STAT get_misses 27185
STAT delete_misses 0
STAT delete_hits 0
STAT incr_misses 0
STAT incr_hits 0
STAT decr_misses 0
STAT decr_hits 0
```

دسترسی گرفتیم . حالا میتوانید روی اهداف دیگه این رو تست کنید .

گاهی اوقات ممکنه یک سازمان اومنده باشه و برنامه باگ بونتی گذاشته باشه و خب شما میتوانید از طریق چک کردن پورتهای باز روی سرور های اون سازمان یا کمپانی بررسی کنید ببینید مثلًا ایا از Memcached استفاده میکنن یا نه و اگه استفاده میکنن ایا Authentication داره یا نداره ؟ بعضی اوقات میتوانه منجر به Information Disclosure و بانتی خوبی بابتش پرداخت بشه .

سرور RabbitMQ چیست ؟ این نرم افزار بر روی پورت 15672 و 5672 با یک رابط کاربری تحت وب کار میکنه و بیشتر توی مبحث DevOps کاربرد داره . این نرم افزار یک Message-Queueing هست و همچنین بهش Message-Broker, Queue Manager میگن . به طور ساده تر بخواه RabbitMQ جایی هست که Queue ها یا صفحه ها تو شعری میشن و اونجا تعریف میشه که کدام اپلیکیشن ها جهت انتقال پیام یا پیامها متصل بشن . حالا مفهوم کلمه Message یا پیام چی هست که RabbitMQ برای انتقال اونها صفتندی میکنه ؟ پیام میتوانه هر نوع داده ای باشه مثلًا میتوانه اطلاعاتی درمورد یک Task یا پروسه باشه که باید توسط یک اپلیکیشن خاص اجرا بشه یا حتی میتوانه یک متن ساده باشه . این RabbitMQ پیام ها یا همون Message ها رو توی یک صفت زمانیکه یک اپلیکیشن ببیاد و اون رو برداره نگهداری میکنه . یه طور ابی کارش اینطوری است حالا اگه درک زیادی هم نکردید عیوبی نداره چون من خودم هم درست نمیفهمم ولی خب چیزی هست که تونستم از مقالات، اینجا بتوییم درموردش .



Web Application Penetration Testing Note

خب حالا بیایم و سرور هایی که این **RabbitMQ** روشون قرار داره رو پیدا کنیم و یه نگاهی به رابط کاربریش بندازیم . کافیه که توی شدنی **Shodan.io** یا **Censys.io** به دنبالش بگردیم . خب گفتیم پورت 5672 یا 15672 میتونه یک سرویس **RabbitMQ** توی خودش داشته باشه .

The screenshot shows the Censys search interface with the query "services.port=15672". The results page displays several hosts found, with two examples highlighted:

- 183.237.40.180**: Linux, CHINAMOBILE-CN China Mobile Communications Group Co., Ltd. (9808), Guangdong, China. Services listed: 1701/L2TP, 3306/MYSQL, 5000/HTTP, 6660/SSH, 6661/HTTP, 7547/HTTP, 15672/HTTP.
- 183.224.101.176**: Linux, CHINAMOBILE-CN China Mobile Communications Group Co., Ltd. (9808), Shanghai, China. Services listed: 22/SSH, 80/HTTP, 443/HTTP, 1000/HTTP, 1001/HTTP, 1200/HTTP, 1234/HTTP, 1300/HTTP, 1430/HTTP, 1503/HTTP, 1511/HTTP, 2377/UNKNOWN, 3202/HTTP, 3204/HTTP, 3206/HTTP, 3207/HTTP, 3208/HTTP, 3209/HTTP, 3210/HTTP, 3211/HTTP, 3212/HTTP, 4101/HTTP, 4102/HTTP, 4105/HTTP, 4106/HTTP.

Results

Host Filters

Labels:

105.19K jquery
81.33K remote-access
51.53K database
26.05K bootstrap
13.93K login-page
 More

Autonomous System:

23.94K ALIBABA-CN-NET
Hangzhou Alibaba Advertising Co.,Ltd.
11.04K TENCENT-NET-AP
Shenzhen Tencent Computer Systems Company Limited
11.04K AMAZON-02
5,150 DIGITALOCEAN-ASN
4,077 SPACENET-AS Internet Service Provider
 More

Location:

48.47K China
13.95K United States
7,965 Russia
7,748 Germany
3,103 Singapore

Hosts

Results: 109,890 Time: 0.11s

183.237.40.180

Linux, CHINAMOBILE-CN China Mobile Communications Group Co., Ltd. (9808), Guangdong, China. Services listed: 1701/L2TP, 3306/MYSQL, 5000/HTTP, 6660/SSH, 6661/HTTP, 7547/HTTP, 15672/HTTP.

183.224.101.176

Linux, CHINAMOBILE-CN China Mobile Communications Group Co., Ltd. (9808), Shanghai, China. Services listed: 22/SSH, 80/HTTP, 443/HTTP, 1000/HTTP, 1001/HTTP, 1200/HTTP, 1234/HTTP, 1300/HTTP, 1430/HTTP, 1503/HTTP, 1511/HTTP, 2377/UNKNOWN, 3202/HTTP, 3204/HTTP, 3206/HTTP, 3207/HTTP, 3208/HTTP, 3209/HTTP, 3210/HTTP, 3211/HTTP, 3212/HTTP, 4101/HTTP, 4102/HTTP, 4105/HTTP, 4106/HTTP.

37.187.73.58 (ns3362558.ip-37-187-73.eu)

Linux, OVH (16276), Hauts-de-France, France. Services listed: 22/SSH, 25/SMTP, 40/HTTP, 110/POP3, 111/PORTMAP, 120/IMAP, 443/HTTP, 465/UNKNOWN, 993/IMAP, 1000/SSH, 3000/HTTP, 3128/HTTP, 4190/PIGEONHOLE, 5672/AMQP, 6379/REDIS, 8006/HTTP, 15672/HTTP.

خب، صفحه لاگین **RabbitMQ** رو توی تصویر زیر میبینید .

The screenshot shows the RabbitMQ Management UI at the URL <http://183.237.40.180:15672>. The page displays a login form with fields for Username and Password, and a Login button.

به صورت پیش فرض Credential روی این سرویس **guest/guest** هست و گاهی اوقات پیش میاد که یادشون میره عوض کنن و میشه به راحتی لاگین کرد . مثلا توی همین مورد :

The screenshot shows the RabbitMQ Management UI Overview page at the URL <http://183.237.40.180:15672/#>. The page displays various metrics and statistics for the RabbitMQ cluster, including:

- Overview** tab selected.
- Totals** section: Queued messages (last minute) chart, Ready: 85,174, Unacked: 0, Total: 85,174.
- Message rates** section: Publish: 0.00/s, Publisher confirm: 0.00/s, Unroutable (return): 0.00/s, Disk read: 0.00/s, Disk write: 0.00/s.
- Global counts** section: Connections: 0, Channels: 0, Exchanges: 8, Queues: 1, Consumers: 0.

توی تصویر زیر میتوانید لیستی از Queue ها رو بینید که برای این سرویس تعریف شدند :

Queues

All queues (13)

Pagination

Page 1 of 1 - Filter: Regex ?

Overview				Messages			Message rates			+/-
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	
queue.fanout.new.media.resize.and.save	classic	D	idle	0	0	0				
queue.fanout.new.post.save.media.to.bucket	classic	D	idle	0	0	0				
queue.mail.listing.claim.accepted.alert.listing.email	classic	D	idle	2	0	2				
queue.mail.listing.claim.accepted.alert.owner	classic	D	idle	0	0	0				
queue.mail.listing.claim.refused.alert.listing.email	classic	D	idle	0	0	0				
queue.mail.listing.claim.refused.alert.owner	classic	D	idle	1	0	1				
queue.mail.listing.claim.request.alert.user	classic	D	idle	0	0	0				
queue.mail.listing.enabled.alert.listing.email	classic	D	idle	0	0	0				
queue.mail.listing.enabled.alert.owner	classic	D	idle	0	0	0				
queue.mail.listing.enabled.changed.alert.listing.email	classic	D	idle	0	0	0				
queue.mail.listing.enabled.changed.alert.owner	classic	D	idle	0	0	0				
queue.new.media.resize.and.save	classic	D	idle	13	0	13				
queue.new.post.save.media.to.bucket	classic	D	idle	26	0	26				

▶ Add a new queue

گاهی ممکنه توی لاگین کردن به RabbitMQ یا برخی سرویس های دیگه پیامی به شکل زیر دریافت کنید :



Login failed

Username: *

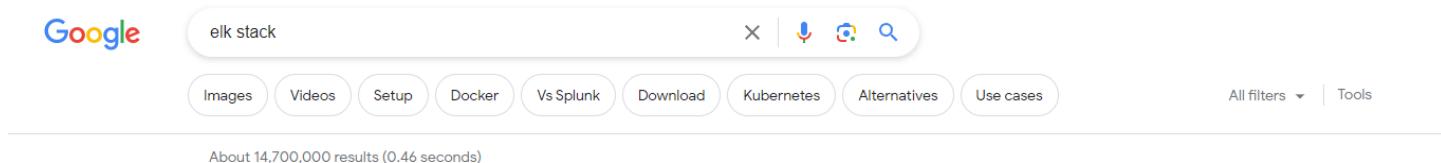
Password: *

User can only log in via localhost

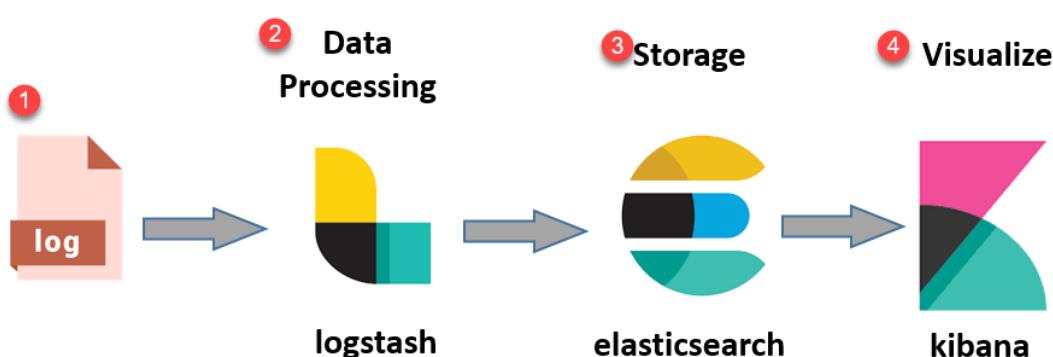
این یعنی اینکه شما نمیتوانید از بیرون از سرور به این سرویس متصل بشید و فقط باید localhost باشد و به این معنی هم هست که ورودی شما درسته ولی شما localhost Credential نیستید .

اینو هم بگم که هر کدام از این میکروسرویس ها خودش یک دوره کامل داره و نمیشه که عملکرد کامل اونها رو توی همین PDF توضیح داد و ما فقط در حد اشنایی باهشون صحبت میکنیم.

میکروسرویس ElsaticSearch چیست؟ یکی از محصولات کمپانی Open Source Elastic NV هست که به صورت رایگان و کلاینت ها قرار داده شده است. در دنیای امروز ما هر روزه مقدار بسیار زیادی داده انتقال پیدا میکند و چیزی حدود 2.5 کوینتیلیون بایت. این دادهها در این مقایسه رو Big Data میگن و بسیاری از انها برای تجزیه و تحلیل شدن نیازمند ابزار هایی هستند. یکی از پرکاربرد ترین ابزارهایی که برای تجزیه و تحلیل دادههای بزرگ ایجاد شده، ElasticSearch هست. ElasticSearch از کتابخونه ای به نام Apache Lucene استفاده میکنه که بسیار کتابخونه پیچیدگی اون رو تا حد بسیار زیادی کاهش داده. دادهها با یک API Restful جابجا میشوند و در نهایت به شکل یک JSON تحويل کلاینت داده خواهند شد. اغا اینکه اینجا توی این موقعیت بخوایم به تحقیق بیشتر درمورد ElasticSearch بپردازیم واقعاً جاش نیست چون با توجه به چیزایی که من شنیدم یه سرویس گردن کلفته و خودش دوره هایی داره و اسه خودش، اصن سر و سالاریه. بگم که این سرویس خودش با دونا سرویس دیگه که اونها هم رایگان و Open Source هستند و محصول شرکت Elastic NV میباشد، استفاده میشه به نامهای Logstash و Elasticsearch که دادهها رو لاغ میکنه و تحويل Kibana میده و Elasticsearch که دادههای Elasticsearch رو تحويل میگیره و به شکل GUI و Visualization شده به کلاینت نشون میده، میشه کوئری زد توش و ... جایی که Elasticsearch وجود داشته باشه معوملا هم وجود داره و همچنین Logstash. به ترکیب این سه تا ابزار ELK میگن:



اها یادم رفت بگم که این میکروسرویس روی پورت 9200 کار میکنه و Credential اون اگه وجود داشته باشه به صورت پیش فرض guest/guest هست. و خب تصویر زیر نحوه عملکرد این سه تا ابزار رو با هم به خوبی نشون داده و به نظرم برای درک بهتر کمک میکنه:



© guru99.com

حالا بریم و یه سرچ کنیم ببینیم ایا سرور هایی پیدا میکنیم که Elasticsearch رو روی پورت 9200 بدون Authentication داشته باشند یا خیر؟ من توی جستجوی ایندم از Shodan.io از طریق Favicon Elasticsearch مربوط به استفاده کردم:

Web Application Penetration Testing Note

TOTAL RESULTS: 35,745

TOP COUNTRIES:

Country	Count
China	14,651
United States	6,180
Germany	2,314
Hong Kong	1,472
India	1,196
More...	

TOP PORTS:

Port	Count
9200	29,453
443	3,920
80	598
8200	340
9201	214

Access Granted: Want to get more out of your existing Shodan account? Check out [everything you have access to](#).

124.221.214.183 (Tencent cloud computing (Beijing) Co., Ltd, China, Beijing) - database

182.92.216.246 (Aliyun Computing Co., LTD, China, Beijing) - database

124.221.74.248 (Tencent cloud computing (Beijing) Co., Ltd, China, Beijing) - database

58.123.210.27 (SK Broadband Co.Ltd, Korea, Republic of, Seoul) - database

نتایج رو میبینید که، کشور دوستمنون که خیلی دوشن داریم (اصن میمیرم برash) اقای چین در صدر استفاده از این ابزار هستند. خب بازنگیم چندتاش رو ببینیم چی بهمنون میده.

```
{
  "name" : "12926514ef96",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "gZE65d9ySbyNkhDjg_1-vA",
  "version" : {
    "number" : "7.3.0",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "de777fa",
    "build_date" : "2019-07-24T18:30:11.767338Z",
    "build_snapshot" : false,
    "lucene_version" : "8.1.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

این یکی از خروجی های Elasticsearch هست و میبینید یک JSON هست. اما برای نشون دادن این دادهها به صورت Visualization و دیدنشون، باید Kibana رو نصب کنیم. حالا میتوانیم بیایم روی سیستم خودمن نصب کنیم و یا هم شناس بیاریم و روی تارگت روی پورت 5601 نصب باش. در برخی موارد نصب هست و میاره و برخی هم موارد هم نصب نیست و میتوانید خودتون روی سیستم خودتون نصبش کنید. مثلًا توی مثال زیر نصیبه:

Help us improve the Elastic Stack by providing usage statistics for basic features. We will not share this data outside of Elastic. Read more

Yes No

Add Data to Kibana

Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.

خب حالا چرا باید این میکروسرویس رو پیدا کنیم؟ این میکروسرویس دید خوبی نسبت به دادههای ورودی به سرور های تارگت به ما میده و ما میتوانیم اونها رو شناسایی کنیم و خب توی باگ بونتی موفق تر عمل کنیم چون که تقریبا هر داده ای که بهش گفتن رو لاغ کرده. تا همینجا به نظر من درمورد این ابزار ها کافیه و بریم سروقت میکرسرویس بعدی.

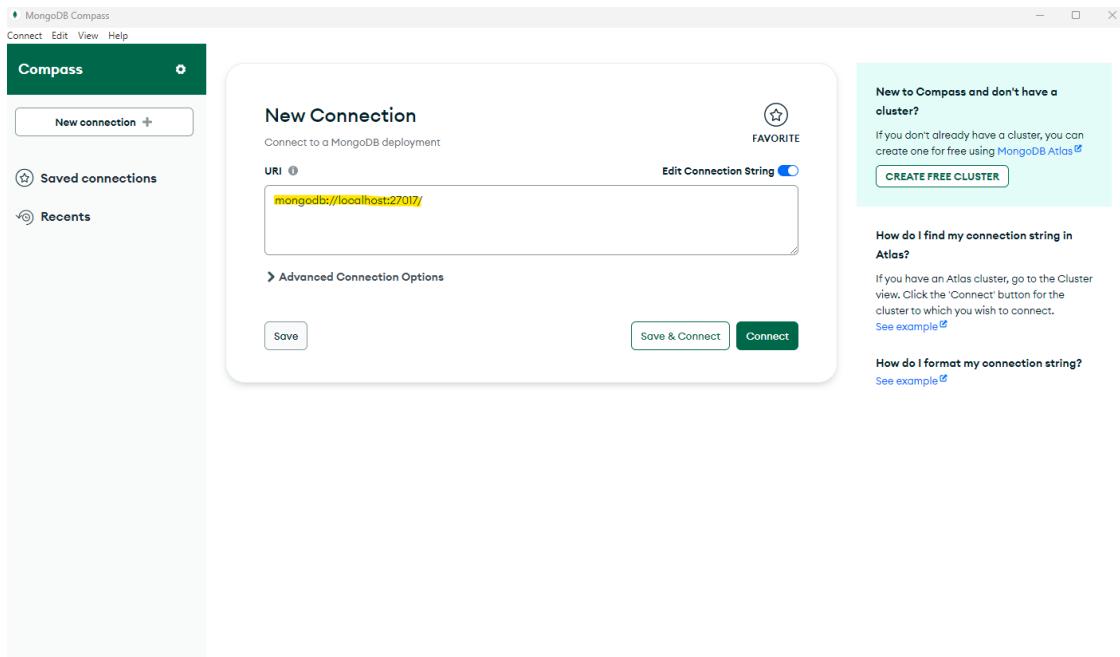
MongoDB چیه؟ یک نرم افزار Source-Available, Cross-Platform, Document-Oriented Database است. حالا اینا چی هستند باید بگم که Document-Oriented Database یعنی اینکه دادهها در Document هایی به صورت Binary ذخیره میشن و یک نوع دیتابیس NoSQL هستند. یعنی اینه روی هر Platform قابل اجراست و Source-Available هم یعنی اینکه شرایطی وجود داره که میشه Source-Code رو دید و یا گاهی اوقات تغییر داد. اینا مهم نیستن زیاد، همین که بدونیم یک MongoDB از نوع NoSQL هست و روی پورت 27017 کار میکنه کافیه. دادهها در Document ها به حالت JSON-Like Database ذخیره میشن. یه شکلی به صورت زیر:

```
{
  "_id": 1,
  "first_name": "Tom",
  "email": "tom@example.com",
  "cell": "765-555-5555",
  "likes": [
    "fashion",
    "spas",
    "shopping"
  ],
  "businesses": [
    {
      "name": "Entertainment 1080",
      "partner": "Jean",
      "status": "Bankrupt",
      "date_founded": {
        "$date": "2012-05-19T04:00:00Z"
      }
    },
    {
      "name": "Swag for Tweens",
      "date_founded": {
        "$date": "2012-11-01T04:00:00Z"
      }
    }
  ]
}
```

توی تصویر بالا دادهایی درمورد یک کاربر با first_name Tom ذخیره شده است. میبینید که به شکل JSON هستند و خب قاعدها هم به همون شکل قابل دسترسی می باشد. حالا وجود این میکروسرویس واسه چی واسه ماهایی که میخوایم Bug-Bounty کار کنیم مهمه؟ این میکروسرویس که روی پورت 27017 کار میکنه به صورت پیش فرض Authentication نداره و گاهی اوقات حتی کمپانی های بزرگ هم ممکن هست که این سرویس رو روی یکی از سرور هاشون داشته باشد، اطلاعات حساسی توشون باشه ولی Authentication نخواهد و همین میشه که دادهایی کمپانی و شرکت بزرگ Leak میشه. خب بریم چندتا سرج بزنیم توی Censys.io, Shodan.io ببینیم میتوانیم سروری رو پیدا کنیم که این سرویس رو بدون Authentication ارائه کنه یا خیر؟

Host	Ports	Services
34.216.131.181	22/SSH, 445/SMB, 110/POP3, 25/SMTP, 631/IPP, 2404/EC60870_5_104, 9900/VNC, 10001/ATG, 23/UNKNOWN	SSH, SMB, POP3, SMTP, IPP, EC60870_5_104, VNC, ATG, UNKNOWN
34.154.81.151	22/SSH, 445/SMB, 110/POP3, 25/SMTP, 631/IPP, 2404/EC60870_5_104, 1723/PTP, 21/FTP, 3306/MySQL, 1153/MONGODB, 27017/MONGODB, 5060/SIP, 1883/MQTT	SSH, SMB, POP3, SMTP, IPP, EC60870_5_104, PTP, FTP, MySQL, MONGODB, SIP, MQTT, UNKNOWN
34.213.49.214	22/SSH, 445/SMB, 110/POP3, 25/SMTP, 631/IPP, 2404/EC60870_5_104, 1723/PTP, 21/FTP, 3306/MySQL, 1153/MONGODB, 27017/MONGODB, 5060/SIP, 1883/MQTT	SSH, SMB, POP3, SMTP, IPP, EC60870_5_104, PTP, FTP, MySQL, MONGODB, SIP, MQTT, UNKNOWN

میبینید که Censys در حدود 300000 رکورد اورد برامون. جالبه. بگم که MongoDB یک رابط کاربری گرافیکی به نام Compass داره که میتوانیم از لینک زیر دانلود و نصب کنیم و بدون مشکل به سرور هامون متصل بشیم و دادهها رو ببینیم و Dump کنیم : <https://www.mongodb.com/try/download/compass>



باید توی اون قسمت که **Highlight** شده ادرس سرورمون رو بنویسیم . مثلا به شکل زیر :

URI i **Edit Connection String**

`mongodb://34.216.131.181:27017/`

یادمون باشه که پورت پیش فرض 27017 هست و اگه روی پورت دیگه ای بود باید اون رو وارد کنیم .

MongoDB Compass - 118.31.69.243:27017/READ_ME_TO_RECOVER_YOUR_DATA.README

Connect Edit View Collection Help

118.31.69.243:27... ... Documents READ_ME_TO_R... +

My Queries Databases Search

READ_ME_TO_RECOVER_YOUR_DATA.README n/a n/a DOCUMENTS INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#) + Explain Reset Find Options

[ADD DATA](#) EXPORT DATA

`_id: ObjectId('6570a0766c946746e324d1a6')`
`content: "All your data is backed up. You must pay 0.01 BTC to 14PYVptPexgRpHrm7..."`

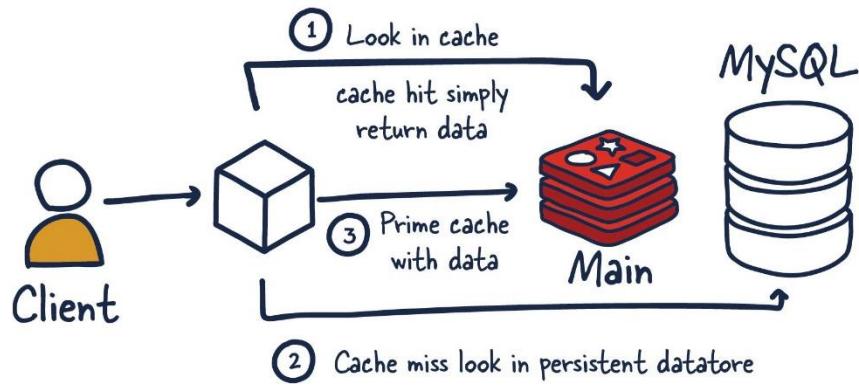
خب توی تصویر بالا میبینید که متصل شدیم ولی متنسفانه دادههای توی دیتابیس حذف شده و کسی که حذف کرده گفته اگه دادهها رو میخوای 0.01 BTC بزن به ولتم تا برات برگردونم 😊 جالبه .

Redis چیست؟ یک پایگاه داده NoSQL است که دادهها را داخل مموری (مثل Memcached) ذخیره میکنه . دادهها به شکل Key-Value ذخیره میشن یعنی اینکه به هر داده یک کلید تعلق میگیره که از طریق اون کلید میشه به مقدار داده متناسبی پیدا کرد . از این پایگاه داده به عنوان ذخیره کننده Cache ها، Message-Broker هم استفاده میشه . این پایگاه داده موقتی از انواع مختلف Data Structure مثل رشته، لیست، سرویس، sets، maps، bitmaps، HyperLogLogs و ... پشتیبانی میکنه و میتوانه ذخیره کنه . خب سرویس مربوط به Redis روی پورت 6379 کار میکنه و ما به عنوان یک باگ هانتر میتوانیم سرور های یک تارگت را بررسی کنیم و ببینیم که Redis داره یا نه ؟ خیلی اوقات پیش میاد که Authentication نداره و میتوانیم دسترسی بگیریم .



در تصویر زیر هم یک نمایی از استفاده از Redis به عنوان محل ذخیره Cache میباشیم که چطوری کاربر باهاش ارتباط برقرار میکنه :

How is redis traditionally used



بریم سروقت جستجو کردن توی Censys و Shodan برای پیدا کردن سرور هایی که روشن Redis وجود داره و ببینیم میتوانیم بدون Authentication متصل بشیم یا نه ؟

Host	Ports	Services
216.238.85.24 (vultrusercontent.com)	6379/Redis	Redis
213.233.182.8 (mail.yottab.io)	6379/Redis	Redis

خب میبینید که حتی یکی هم واسه ایران پیدا کرد که مثل اینکه یک میل سرور دانشگاه شریف هستش . وصل نمیشیم چون خطریه 😊

برای اتصال به Redis میتوانیم از redis-cli توی لینوکس استفاده کنیم و کافیه که به شکل زیر این کار رو انجام بدیم :

```
kali@kali:~$ redis-cli -h 3.36.104.99
3.36.104.99:6379> info
# Server
redis_version:7.2.0
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:f0c55e18c24c25f0
redis_mode:standalone
os:Linux 5.19.0-1025-aws x86_64
arch_bits:64
monotonic_clock:POSIX clock_gettime
multiplexing_api:epoll
atomicvar_api:c11-built-in
```

میبینید که هیچ Authentication از ما نخواست و ما متصل شدیم . حالا کافیه دستورات مربوط به Redis رو بزنیم و داده‌های داخلش رو استخراج کنیم . دستور info رو زدم و میبینید که اطلاعاتی از سرور بهمن داده . اگه بخوایم کلید ها رو ببینیم کافیه دستور زیر رو بزنیم .

```
3.36.104.99:6379> keys *
1) "backup4"
2) "backup3"
3) "backup2"
4) "backup1"
```

میبینید که چهارتا کلید داخل این Redis وجود داره و اگه بخوایم مقادیر اونها رو ببینیم کافیه دستور زیر رو بزنیم :

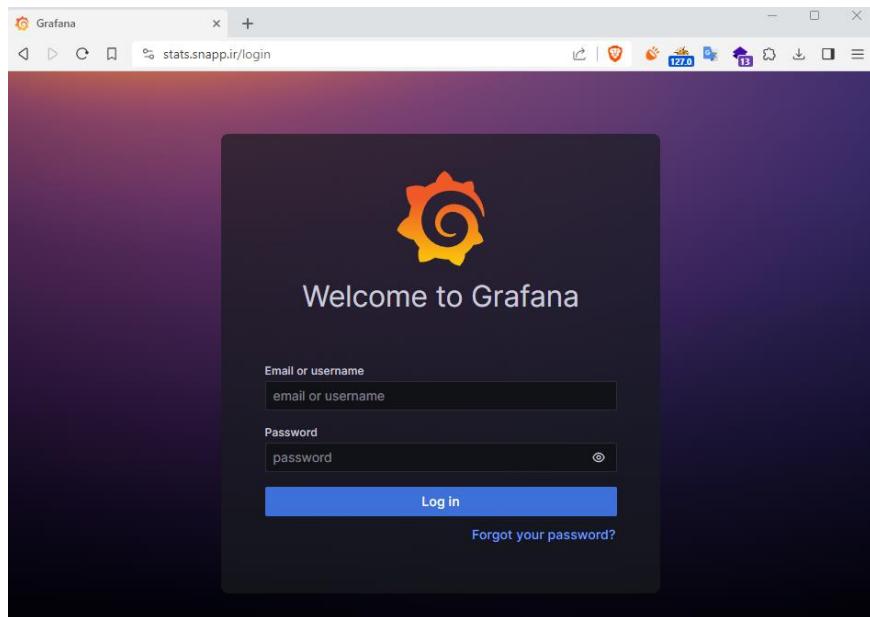
```
3.36.104.99:6379> dump backup1
"\x00@I\n\x0b\x00\xd36(\x0f\xab\xfeC\x15"
"*\x00@\x0b\x00\xd36(\x0f\xab\xfeC\x15" مقداری عجیب غریب داره و خب این یعنی اینکه ایشون یه بد افزار رو توی مموریشون اجرا کردن و الوده هستند . عموما میان و اینکار رو با Redis ها انجام میدن و معمولا هم بد افزارهای Miner هستند و خب ممکنه توی تستتون ببینیدشون . ولی خب اگه داده ای توی Redis ذخیره باشه اینجا میتوانید ببینید .
```

بهتره که بدونیم که از طریق Redis میان و RCE میزنان یعنی میان و Remote Command Execution میکنن . اگه خواستید میتوانید توی گوگل رو جستجو کنید و بخونید .

Grafana چیست ؟ یک سرویس بر پایه وب می باشد که به صورت Open-Source و Multi-Platform است و کارش تجزیه و تحلیل و visualization می باشد . مثل همون کاری که Kibana میکرد و گاهی اوقات هم با همون مجموعه اجرا میشه و کار میکنه . نسبت به Kibana رابط گرافیکی زیباتری داره و روی پورت 3000 کار میکنه و Credential پیش فرضش هم admin/admin هست .



خب مثل هم اگه بخوام بزنم میبینید که snapp.ir هم از این سرویس داره استفاده میکنه و توی ادرس stats.snapp.ir وجود داره .



خب رفته نوی shodan.io و عبارت Grafana رو جستجو کردیم و نتیجه رو در زیر میبینید :

COUNTRY	HOST COUNT
Germany	822
United States	797
China	491
Netherlands	243
Russian Federation	131
More...	

پنل کاربری این نرم افزار هم به شکل زیر است :



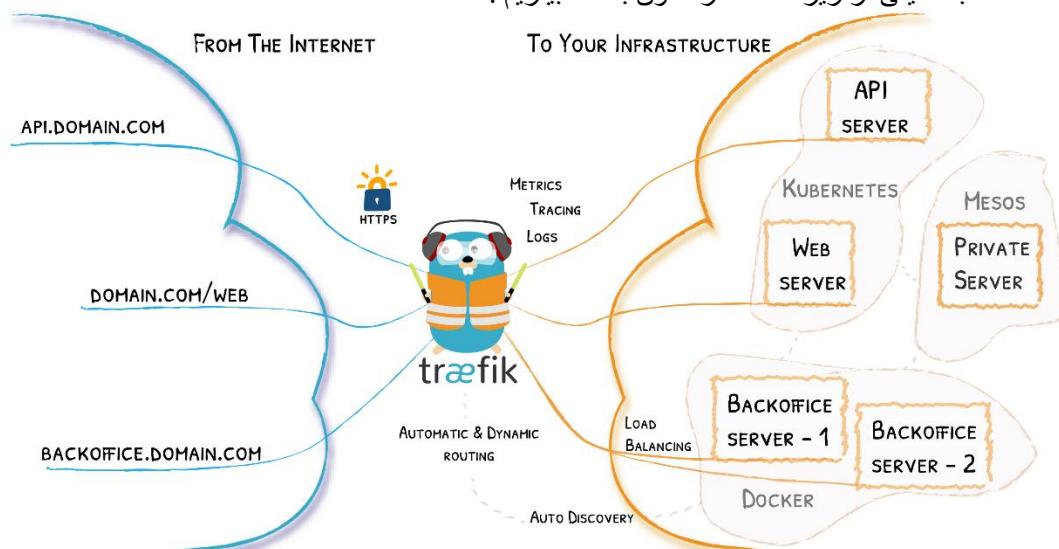
حالا چه کاربردی میتونه واسه ما داشته باشه ؟ گاهی اوقات پیدا میشه که یا admin/admin Credential نداره و یا هم به صورت پیش فرض هست و ما میتوانیم ازش اطلاعاتی بدست بیاریم، شاید بتونیم IP Address بدست بیاریم، شبکه سرور رو تحلیل کنیم و سخت افزار رو ببینیم ... اگه یه جایی روی یک تارگت خاص خواستیم تست کنیم که ایا Grafana داره یا نه میتوانیم Path زیر رو تست کنیم و اگه داشته باشه میاره [https://\[Domain\]/api/datasources/proxy](https://[Domain]/api/datasources/proxy)

میکروسرویس Graphite چیه؟ ایشون یک میکروسرویس جهت مانیتورینگ سخت افزار و زیر ساخت های ابری هست و گاهی اوقات ممکن هست که بینیدشون. این میکروسرویس روی پورت 8080 کار میکنه و خدمات ارائه میدهد.



اگه توی یکی از تارگتاون چنین سرویسی در حال اجرا بود شاید بتونید اطلاعاتی درمورد سخت افزار، گاهی اوقات IP Address هایی و ... رو بدست بیارید.

میکروسرویس Traefik چیست؟ خب این سرویس روی پورت 8080 کار میکنه و میاد به عنوان Edge Router مابین Internet و Infrastructure قرار میگیره و دادههایی که از سمت اینترنت با پروتکل HTTPS بهش میرسه رو ما بین سرویس های داخل زیرساخت ما مسیر دهی میکنه. یک سرویس بسیار جالب و خوب هست ولی کم ازش استفاده میشه و اگه ما به عنوان یک باگ هانتر بتونیم بهش دسترسی بگیریم میتوانیم تقریباً نقشه نسبتاً دقیقی از زیرساخت تارگتمون بدست بیاریم.



توی تصویر بالا یه نمایی از عملکرد این سرویس رو میبینید. از این سرویس به عنوان یک Reverse Proxy و Load Balancer هم یاد میشه.

برای پیدا کردن نمونه هایی از این میکروسرویس توی گوگل میتوانید از دورک زیر استفاده کنید:

inurl:8080/dashboard traefik

Entrypoint	Port
IMAP	:143
IMAPS	:993
SMTP	:25
SMTPS	:465
SUBMISSION	:587
TRAEFIK	:8080

میکروسرویس Jenkins چیست؟ یک سرور Open-Source Automation جهت کردن هست و کمک میکنه که قسمتهای مختلف توسعه یک نرم افزار مثل ... Automate Testing, Deploying ... را کنند. این میکروسرویس رو پورت 8080 کار میکنه و میتوانید از طریق گوگل دورک زیر توی گوگل جستجو کنید.

Intitl:"Dashboard [Jenkins]" Credentials

یا هم میشه از طریق کوئری زیر توی Shodan.io نمونه هایی ازش رو دید:

http.favicon.hash:81586312

این کوئری از طریق Favicon جستجو میکنه

The screenshot shows the Shodan search interface with the query 'http.favicon.hash:81586312'. It displays a world map with red dots indicating found hosts. Below the map, there's a table of 'TOP COUNTRIES' and 'TOP PORTS' with counts. On the right, detailed results for three hosts are shown:

- 112.111.16.67**: Fuzhou City, Fujian provincial network, China (PROVIDER: China, Future). Status: HTTP/1.1 403 Forbidden. Date: Thu, 07 Dec 2023 19:27:22 GMT. X-Content-Type-Options: nosniff. Set-Cookie: JSESSIONID=000f1f8d1f0000000000000000000000; Path=/; HttpOnly. Expires: Thu, 03 Jan 1970 00:00:00 GMT. Content-Type: text/html; charset=utf-8. X-Hudson: 1.395. X-Jenkins: 2.348...
- 34.233.100.164**: 34.233.100.164 (complete-fam.economi.com, Amazon Technologies Inc., United States, Ashburn). Status: HTTP/1.1 403 Forbidden. Date: Thu, 07 Dec 2023 19:24:43 GMT. Content-Type: text/html; charset=UTF-8. Transfer-Encoding: chunked. Connection: keep-alive. X-Content-Type-Options: nosniff. Set-Cookie: JSESSIONID=000f1f8d1f0000000000000000000000; Path=/; HttpOnly. Expires: Thu, 03 Jan 1970 00:00:00 GMT. Content-Type: text/html; charset=UTF-8. X-Hudson: 1.395. X-Jenkins: 2.428...
- 181.167.94.232**: 232.94.167.101 (ibertel.com.ar, Telecom Argentina S.A., Argentina, Buenos Aires). Status: HTTP/1.1 403 Forbidden. Date: Thu, 07 Dec 2023 19:24:12 GMT. X-Content-Type-Options: nosniff. Set-Cookie: JSESSIONID=000f1f8d1f0000000000000000000000; Path=/; HttpOnly. Expires: Thu, 03 Jan 1970 00:00:00 GMT. Content-Type: text/html; charset=UTF-8. X-Hudson: 1.395. X-Jenkins: 2.428...

در برخی اوقات این سرویس Authentication نیاز داره و در برخی اوقات هم نیاز نداره و میشه به پروژه هایی که داخلش تعریف شدن دسترسی پیدا کرد. تصویر زیر نمونه دشبورد سرویس Jenkins هست:

The screenshot shows the Jenkins dashboard with the URL 'http://jenkins:8080'. It features a sidebar with links like People, Build History, Project Relationship, Check File Fingerprint, Job Priorities, and Credentials. The main area shows a table of builds:

S	W	Name	Last Success	Last Failure	Last Duration
●	☀️	AKSW-Deb-Util	6 yr 8 mo - #5	5 yr 9 mo - #5	19 sec
●	☀️	AutoSPARQL	N/A	N/A	N/A
●	☀️	cstdaller-tools	6 yr 6 mo - #26	N/A	23 sec
●	🌧️	DL-Learner	1 yr 10 mo - #916	1 yr 9 mo - #920	27 min
●	☀️	DL-Learner CI-M	N/A	N/A	N/A
●	🌧️	DL-Learner Merge-M	5 mo 2 days - #1	6 mo 16 days - #1	2.2 sec
●	🌧️	DL-Learner regression statistics	6 yr 0 mo - #231	6 mo 16 days - #316	7.8 sec
●	☀️	DL-Learner examples	20 hr - #2700	6 mo 12 days - #2508	49 min

اگه کوئری رو به شکل زیر توی Shodan.io بزنید سرور هایی رو بهتون نشون میده که نیاز به Authentication ندارند:

http.favicon.hash:81586312 -403

به Shodan گفتیم که بیاد و از طریق Favicon جستجو کنه و اونهایی که 403 (Forbidden Status Code) توی نتیجه‌شون هست رو نشون نده. از این طریق فقط اونهایی رو نشون میده که 200 (OK Status Code) برگردوند.

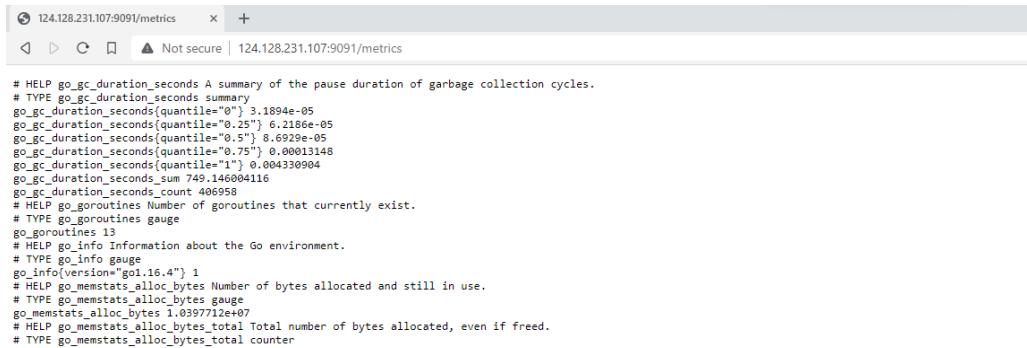


Jenkins

گاهی پیش میاد که توی دشبورد این سرویس سورس کد هایی وجود داره که مهم هستند و میشه از طریقشون به اطلاعات خیلی زیادی دسترسی پیدا کرد و یک حفره امنیتی Source Code Disclosure محسوب میشه و میشه گزارش کرد و بانتی خوبی گرفت.

میکروسرویس Prometheus چیست؟ یک میکروسرویس جهت مانیتورینگ هست. اطلاعاتی مثل اطلاعات سخت افزاری سرور، مقدار ترافیک شبکه و ... رو نشون میده. این میکروسرویس قابلیت اینو داره که برآتون نمودار رسم کنه، گراف بکشه و کلا دادهها رو به شکلی Visualization کنه. به صورت پیش فرض روی پورت 9090 و 9091 اجرا میشه و خب یک جایی داره که metrics آونجا هست و کلا دادهها رو به صورت غیر Visualization نشون میده و توی ادرس [https://\[SERVER_IP_OR_DOMAIN\]:9090/metrics](https://[SERVER_IP_OR_DOMAIN]:9090/metrics) هست.

این صفحه یه چیزی به شکل زیر هست:



```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 3.1894e-05
go_gc_duration_seconds{quantile="0.25"} 6.2186e-05
go_gc_duration_seconds{quantile="0.5"} 8.6929e-05
go_gc_duration_seconds{quantile="0.75"} 0.00013148
go_gc_duration_seconds{quantile="1"} 0.004330904
go_gc_duration_seconds_sum 749.146004116
go_gc_duration_seconds_count 406958
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 13
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.16.4"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.0397712e+07
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
```

ولی خب وقتی ما به این دادهها روی یک تارگت دسترسی پیدا کردیم و Authentication خواست قشنگ نیست به عنوان POC اینها رو بفرستیم برآشون. میتوانیم بیایم و یک Prometheus روی سیستم خودمون بیاریم بالا و این ادرس رو بهش بدیم تا برآمون دادههای داخل اینجا رو Visualization کنه و بعد به عنوان POC نماهایی که برآمون رسم میکنه رو بفرستیم. اگه بخوایم توی Shodan.io به دنبال سرور های Prometheus باشیم و اونها رو ببینیم کافیه که کوئری زیر رو جستجو کنیم:

prometheus +"200 OK"

بعد خواهیم دیدی که نتایجی رو برای ما نشون میده.
پیدا کردن این سرویس روی تارگت میتوانه یک مشکل امنیتی Information Disclosure محسوب بشه.



به عنوان اخرين جملات خدمتون عرض کنم که عموم صحبتايي که اينجا توی اين جزوه کردیم مربوط به حوزه باگ بونتی بود و توی حوزه تست نفوذ معمولاً چنین دادههای رو از شما نمیخوان و حتی در برخی اوقات بیانشون منجر به این میشه که بهتون بگن به شماربطری نداره که بخوايد دنبال چنین اطلاعاتی باشيد و همچنین توی Red Team هم به شکل باگ بونتی عمل میشه و هر چیزی رو که میشه از طریقش نفوذی رو تا هر عمقی انجام داد باید استفاده بشه و به ما ربطی نداره و ... نداریم. مثلاً توی تست نفوذ به شما IP Address های پشت ابر رو میدن و نیازی نیست که به دنبال اونا باشید یا IP Range ها و بسیاری از اطلاعات دیگه رو، حتی Credential ها رو در برخی اوقات و ...