

🔒 OSINT - Passive Recon and Discovery of Assets

■ **Reconnaissance** recon, google, passive, osint, emails



pry0cc 🛡️ Leader & Offsec Engineer & Forum Daddy

1 ✎ May '18

OSINT - Passive Recon and Passive Discovery Of Assets

Sup 0x00'ers, to kick this badass series off, I am going to begin with the most important aspect of pentesting. Passive Recon and OSINT. Now, do not let the word 'passive' fool you. This is no light recon, you can uncover vast amounts of information through passive recon, without ever doing anything intrusive.

Define Passive

My definition of the word passive is probably different to what others would define as passive. In my books, it's anything that can be disguised as regular traffic, nothing intrusive, or easily detectable. Basically, if you can't be distinguished as an attacker from a visitor, and you aren't doing anything intrusive/potentially damaging, it's passive. I know I will get a few who will argue to the death this is not 100% passive, and you'd be right, but this is still my initial pre-pentest workflow.

Where do I start?

Good question. This will hugely depend on what sort of pentest you're doing. In penetration testing, there is a spectrum of different types of pentest:

- Black Box Pentest
- White Box Pentest
- Anything in between

A Black Box Pentest is when you're simulating an attacker, you're given a single starting host, and usually a list of in-scope IP addresses, and that is all. You must attempt to discover services, network

design, and things of that nature.

A White Box Pentest is similar, except you are given everything that an internal employee (and more) would have, this includes application source code, network design configurations, diagrams, stuff like that.

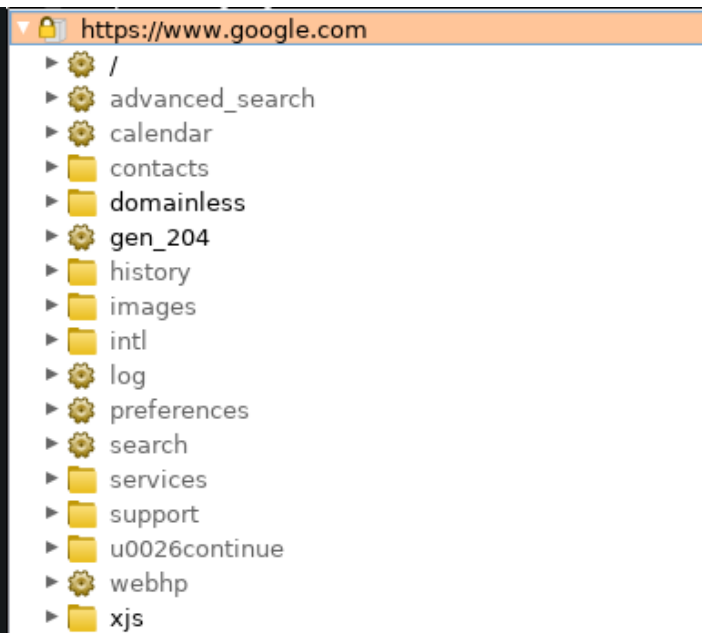
Both types of Pentest have their place, typically a Black Box Pentest is more telling than a White Box Pentest, as it shows what an external attacker could discover with little starting information. In this series, we're going to be covering the former, Black Box Pentesting, as it is the most common sort of Pentest, and will be what most of you are looking for.

Now, if you read the previous paragraphs, you'll quickly notice that I mentioned that you may be given a list of in-scope IP addresses. To simply scan all of these IP's and be done with it is very unrealistic, and so you should first begin with your asset discovery stage. Now there are a tonne of ways we can do this, and I am going to use Google as an example of this, and we're going to be using the company website as our starting point.

<https://www.google.com/> 1

Burp Suite Passive, as observant as Sherlock Holmes

What can we do with this? The first thing I like to do, is actually visit the website with my browser configured to use Burpsuite as a proxy, and with a root SSL cert installed. I am not going to explain how to do this, if anybody wants to write an article on how to do this feel free and I'll link it here!



If you look on the side bar of your BurpSuite, under **Target > Site map > https://www.google.com/**

We can already see a quick overview of immediate file structure of the site, loaded scripts, and also, if we take a look at the wider Site map, a long list of other requested hosts, we may decide to use this information later, as always in pentesting, it is better to collect too much information, and decide not to use it, than to remove it, and end up recollecting it unnecessarily. Remember, this is 'passive' recon, we need to stay very quiet in this stage, and not set off any alarm bells. *

If you have Burp Suite Professional, you can also right click this asset and select "Passively Scan this Host", and it will hunt your existing, requested page code for known vulnerabilities, email disclosure and the like. I do not have Burp Suite Professional, and so I will not be using it 😊

*Note, if this particular website does not receive many page views, you may decide to use a VPN using the same region as the target demographic for this website. If the SOC/SIEM solutions are really sensitive, even a page view from a foreign location could create flags. If your goal is to not be detected by any monitoring solution

(as usually is the goal with a good Pentest), then this might be a thing you would go for.

Okay, so that hasn't helped us significantly, occasionally, things will pop out like `/cgi-bin/`, `/admin/` or `/includes/`, if something does catch your eye, immediately write this down in your reporting software (perhaps in a tool like [Lair](#) ⁵⁴, if somebody makes a writeup on how to spin this up, let me know, and I'll link it here!

Passive port scanning with Shodan, wait, passive, portscanning? What?

So you've been to the website, you know at least port 80 or port 443 is open, but what else is running? You can just open up shodan, or you can use the insanely cool nmap scripts.

Obtain a shodan API key, (if somebody makes a tutorial on this, I'll link it!), and place it inline with this nmap command.

```
nmap --script=shodan-api --script-args 'shodan-api.apikey=XXXXXX' google.com
```

In this image, I have censored my API key, although this is a very simple example, this will do multiple things:

- `-sn` - Disable Port Scan
- `-Pn` - Skip host discovery, don't ping the host
- `-n` - Skip DNS Resolution

Nmap will then realise it has nothing left to do, and will run the shodan-api script. The shodan API script will go out to shodan.io ¹⁰ and retrieve all information it knows about the host, including sometimes host versions and port numbers. This has been way more informational in other circumstances, your mileage may vary.

The point of these tutorials is to provide a very realistic view of what you will see in a normal pentest, and usually, you will not get a single Severity 5 vulnerability with remote code execution, you'll find 5 Severity 3's, and you can then string these together to get a shell, to get access to a panel, or even make a very convincing phishing page. We will cover that in more detail in the exploitation section of this series.

DNS Bruteforcing, but, really fast

Before we go into DNS bruteforcing, we'll look into the low hanging fruit of DNS, and that is zone transfers. Zone transfers was initially a tool used for server administrators to allow them to easily replicate a DNS database, such as transferring to new domain names. If the target company has ever migrated their website, and have little security awareness, this will usually work.

We can easily do domain transfers using a tool called [dnsenum](#) ³⁰, it is written in perl ([@nugget](#)), and is a kind of old-reliable tool used in my pentest arsenal. As you can see here, dnsenum uncovered a few interesting things about the host, namely:

- Host Address
 - This is the IP that you'll get when you do a simple nslookup on the domain
- Wild card host
 - This is the IP that will be returned when you call a random subdomain, such as `kttfvatukbld`, unless you get really lucky (or unlucky, your call), and that is a real subdomain, this is usually an IP from your DNS provider, or from your ISP. If they're really secure, the wildcard domain will be the same as the Host IP, which makes domain

enumeration a bitch.

- Nameservers
 - These are the nameservers that you have used to do the lookup, usually, in a small-medium sized company, this DNS is hosted elsewhere, and can often be courtesy of the domain name registrar. This can be insanely useful information.
- MX Servers
 - Now this information is so easily overlooked, although it is very surprising what this can yield. This will reveal the MX servers of the domain, a lot of companies in the corporate space now a days will use externally hosted email, such as Google or Microsoft/Outlook, quite often these things link to the entire workflow of the company, which can lead you to discover things like Microsoft Lync Servers, Login panels for user email etc.
- Zone Transfers
 - This is quite a rare one now a days, you won't see this work on a lot of hosts that are in the public domain, facebook, twitter etc. Although on pentests, this is surprisingly common. If this succeeds, it will return a list of all registered sub-domains, which is huge. You're better off trying it and not finding anything, than never knowing.

Ok, the fast bit.

Ever heard of Aiodns? Well now you have. Aiodns is a DNS resolver that does synchronous calls over an asynchronous medium. In short, that means that you can efficiently make more than one call without closing the connection after each request. This means it can be, really fast.

Introducing [Aiodnsbrute](#) ⁵⁵ ! Black had the smarts to turn this into a bruteforcing tool, and man it works well. I mean read this:

"Benchmarks on small VPS hosts put around 100k DNS resolutions at 1.5-2mins. An amazon M3 box was used to make 1 mil requests in just over 3 minutes. Your mileage may vary. It's probably best to avoid using Google's resolvers if you're purely interested in speed."

1 million requests, uh, what? Feed this tool a Discovery Dictionary nabbed from [SecLists](#) ²⁸ (Discovery>DNS>subdomains-top1mil-110000.txt is great), and you're on the road to discovering every damn subdomain this domain name has.

What is cooler, is that PCI compliance is a standard, and part of it's requirements is that every host have a signed, valid certificate. Can't have a valid certificate on an IP, so what do firms do? They create

subdomains for everything they need SSL access to. Yep, you guessed it, that includes VPN portals, email logins, development sites (please).

Lets try it on [google.com](https://www.google.com)

```
$ aiodnsbrute google.com
```

Uh, well that's weird, why are there so many of those 92 IP addresses?

Those are actually the DNS resolver. Remember when we did a DNS Lookup of a wildcard? Yeah, it's the same IP. We can easily remedy this with a grep command.


```
aiodnsbrute google.com | grep -v "the resolver IP"
```

Okay, so granted the formatting is a little broken, but this is real world pentesting people, this isn't supposed to be glamorous!

Look at those subdomains, they actually go somewhere. Now you can save these IP addresses in a long text file for further examination, or maybe you want to scan them with your newly learned passive shodan skills? You can do reverse nslookups on the IP's too and see if they resolve somewhere else.

Scraping Emails

You might have used this tool before, it's pretty cool.

It's called theHarvester. What it does, is scrape google results, titles, descriptions, metadata, and looks for things resembling email addresses.

```
./theHarvester.py -d companydomain.com -b google
```

For demonstration purposes, I used protonmail.com ⁴, but put in any company domain name, and you'll usually nab a few emails. If this doesn't work, you can try Hunter.

16,000 results, not bad. What this also does that is so cool is that it tells you the common pattern of emails. What you can do with this (I'll go into this more in detail in part 2), is use this to generate email addresses from names, which can later be used for password spraying (trying a single password for every single email address).

We can do a lot with this information, we can load it into an email program and send out phishing emails, we can password spray with them (I will show you the way), or, we can check them for leaks. This where weleakinfo comes in...

Remember those data breaches, for Adobe, LinkedIn, Myspace? Stuff like that? Well the dumps from these are still out there, and people have published them, you can still find them in old magnet links around the way, however there are thousands of dumps to recover, and they're hard to find.

Weleakinfo 41 takes all this information and compiles it into one big, fast, searchable database. And if you decide to stuff all these emails in there, you might even be lucky enough to get a few old passwords, which you can now variate and try on existing company accounts, how cool is this! We haven't even intrusively touched their servers, and we already have their account passwords? Not so fast.

16,000 emails! If an email lookup took you 10 seconds to do, and take down the results, (which is very fast) it would take you about 48 hours of non-stop copy pasting to go through them all.

If you actually want to be able to sleep, and don't have fingers as muscly as John Cena, then I recommend you automate it. Weleakinfo are actually nice enough to provide a public API, <https://weleakinfo.com/api/public> 15, which allows 3 requests per second, if you want to go faster, try out a tool like ProxyDock

(If somebody wants to write an article on this, and make a tool for this, let me know and I'll link it here)

Geo2IP - Extra

Okay, scenario time. You know about a host's location, but you don't have a clue about its exact details. No problem. Time to use [geo2ip](#) 22. Geo2ip is a tool I developed with @loTh1nkN0t, and it basically takes a rough coordinate location, and then gives you ranges associated with that coordinate. From there, you can feed these into an nslookup tool (maybe make your own with aiodns!), and discover all assets associated with them.

Reverse Whois

Another tool you can use for asset discovery is [reverse whois](#) 47.

Now you can use these tools to enter in a company name, email address, or registrant name (obtained from your previous recon), and then go through these same steps with the newly obtained emails,

domain names, and extra information. Repeat until you have no more passive information to obtain. Now you are ready for Active Recon.

In conclusion, there is a vast amount of information you can obtain just by knowing where to look. This is by no means an exhaustive list, but it contains a lot of the things that I use in my day to day OSINT that really give me a step up in the pentesting scene.



Let me know in the comments if you liked this article, and make sure to click that <3 button to let me know you want more like this. And be sure to share this if you learnt anything to spread the [knowledge](#) 8 !

Thank you for reading, and as always, stay snappy 😊

30 ❤️ 🔗

🔗 Clientside Exploitation in 2018 - How Pentesting Has Changed 6

🔗 OSINT - Shodan API Key 1

created	last reply	12	6.1k	7	46	19	6	 VIP	 VIP
May '18	Jun '18	replies	views	users	likes	links			



zSec The Thoughtful SecAdmin

May '18

Great article @pry0cc ! 😊

Fun fact: In one of my recent pentests I was able to do a DNS zone transfer. I have to admit, I was quiet shocked to see it work in a real scenario. Like you already mentioned, it is pretty rare nowadays.

3 ❤️ 🔗



pry0cc  Leader & Offsec Engineer & Forum Daddy

May '18

A few of the pentests I have done, we got DNS zone transfers. It's worth trying.

I'm glad you enjoyed the article. As a fellow pentester, I would love to see your take on important parts of a pentest. Maybe we can expect a pentesting series from [@zSec](#)'s perspective?



zSec The Thoughtful SecAdmin

May '18

I'd love to do that. Sadly I'm currently really busy as well and I have to travel quite a lot. That's why I'm currently rarely on here as well.
But If I find the time I hope to write something regarding pentesting 😊




zifor zief four

May '18

Great article bro...

how about recon people?



pry0cc  Leader & Offsec Engineer & Forum Daddy

May '18

Recon on individuals? I shall get to that in part 2 😊 Stay tuned.

2 ❤️ 🔗



0xE02B7 irsi

May '18

That is a really cool article.

I have to admit, I have never done a pentest - ok, maybe because I am a Linux System Engineer and not a security consultant / *hacker.

Sometimes our customers hire some to do black box tests.

Most of the times those people are totally bad.

❤️ 🔗



Nitrax 1337 Of The m0n7h

May '18

Dope article mate. Love it. Didn't know about shodan nmap script 😊 Could be useful to import result within metasploit which provides a nice way to handle detected services.

By the same token, have you ever heard about virustotal passive DNS replication? It's redoubtable, you will thanks me later 😊

<https://www.virustotal.com/fr/domain/google.ch/information/> 21

P.S: I will soon release a recon-ng like tool developed with golang, providing well more fonctionnalites than existing tools on the market. Stay tuned.

Best,
Nitrax

1 Reply ▼

5 ❤️ 🔗



Techno_Forg Zain

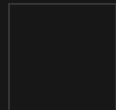
May '18

I think there's a tool written in Python that can do this job... let me see if I can find it (it's saved to my drive)...

(Five hours later)

Alright, I'm back... found the github link after saving the world from aliens.

GitHub 52



graniet/operative-framework

operative-framework - This is a framework based on fingerprint action, this tool is used for get information on a website or a enterprise target with multiple modules (Viadeo search, LinkedIn search...

On a more serious note... this I think should be worth looking into. It does *ALMOST* everything. So yeah... ~Cheers!

—Techno Forg—

2 ❤️ 🔗



pry0cc 🛡️ Leader & Offsec Engineer & Forum Daddy



Nitrox

May '18

That is really cool! Nice share. I might add it later.

1 ❤️ 🔗

16 DAYS LATER



newnoobie

Jun '18

Thanks for sharing your knowledge. It will take me a long time to learn everything in this post, but I am grateful for the easy to understand purpose and background on each tool.

1



pry0cc Leader & Offsec Engineer & Forum Daddy

Jun '18

I love hearing this kind of thing 😊 Thanks for reading and being a part of this journey!

12 DAYS LATER



CLOSED JUN 16, '18

This topic was automatically closed after 30 days. New replies are no longer allowed.

Reply

Suggested Topics

Topic	Replies	Activity
HackTheBox Write-Up - Curling 	11	May 28

Topic	Replies	Activity
[KEYGEN] Balanced Tree ■ Challenges keygen	6	Jan 24
[VulnHub] SickOS 1.1 Writeup ■ CTF	3	5d
Changes to categories, Asking questions on 0x00sec ■ 0x00sec Announcem... support, questions	0	May 21
HackTheBox Writeup: Frolic ■ Hackthebox Writeups	8	Mar 25
Want to read more? Browse other topics in ■ Reconnaissance or view latest topics.		