Passive Reconnaissance

**Task 1 – Introduction**

In this room, after we define passive reconnaissance and active reconnaissance, we focus on essential tools related to passive reconnaissance. We will learn three command-line tools:

* whois to query WHOIS servers
* nslookup to query DNS servers
* dig to query DNS servers

We use whois to query WHOIS records, while we use nslookup and dig to query DNS database records. These are all publicly available records and hence do not alert the target.

We will also learn the usage of two online services:

* DNSDumpster
* Shodan.io

These two online services allow us to collect information about our target without directly connecting to it.

**Task 2 – Passive Versus Active Recon**

Before the dawn of computer systems and networks, in the Art of War, Sun Tzu taught, “If you know the enemy and know yourself, your victory will not stand in doubt.” If you are playing the role of an attacker, you need to gather information about your target systems. If you are playing the role of a defender, you need to know what your adversary will discover about your systems and networks.

Reconnaissance (recon) can be defined as a preliminary survey to gather information about a target. It is the first step in The Unified Kill Chain to gain an initial foothold on a system. We divide reconnaissance into:

1. Passive Reconnaissance
2. Active Reconnaissance

In passive reconnaissance, you rely on publicly available knowledge. It is the knowledge that you can access from publicly available resources without directly engaging with the target. Think of it like you are looking at target territory from afar without stepping foot on that territory.



Passive reconnaissance activities include many activities, for instance:

* Looking up DNS records of a domain from a public DNS server.
* Checking job ads related to the target website.
* Reading news articles about the target company.

Active reconnaissance, on the other hand cannot be achived so directly. It require direct engagement with the target. Think of it like you check the lock on the door and windows, among other potential entry points.



Example of active reconnaissance activities include:

* Connecting to one of the company servers such as HTTP, FTP, and SMTP.
* Calling the company in an attempt to get information (social engineering).
* Entering company premises pretending to be a repairman.

Considering the invasive nature of active reconnaissance, one can quickly get into legal trouble unless one obtains proper legal authorisation.

**Answer the question:**

Q1.You visit the Facebook page of the target company, hoping to get some of their employee names. What kind of reconnaissance activity is this? (A for active, P for passive)

Ans – Passive

Q2. You ping the IP address of the company webserver to check if ICMP traffic is blocked. What kind of reconnaissance activity is this? (A for active, P for passive)

Ans - Active

Q3.You happen to meet the IT administrator of the target company at a party. You try to use social engineering to get more information about their systems and network infrastructure. What kind of reconnaissance activity is this? (A for active, P for passive)

Ans - Passive

**Task 3 - Whois**

WHOIS is a request and response protocol that follows the RFC 3912 specification. A WHOIS server listens on TCP port 43 for incoming requests. The domain registrar is responsible for maintaining the WHOIS records for the domain names it is leasing. The WHOIS server replies with various information related to the domain requested. Of particular interest, we can learn:

* Registrar: Via which registrar was the domain name registered?
* Contact info of registrant: Name, organization, address, phone, among other things. (unless made hidden via a privacy service)
* Creation, update, and expiration dates: When was the domain name first registered? When was it last updated? And when does it need to be renewed?
* Name Server: Which server to ask to resolve the domain name?

To get this information, we need to use a whois client or an online service. Many online services provide whois information; however, it is generally faster and more convenient to use your local whois client. Using the AttackBox (or your local Linux machine, such as Parrot or Kali), you can easily access your whois client on the terminal. The syntax is whois DOMAIN\_NAME, where DOMAIN\_NAME is the domain about which you are trying to get more information. Consider the following example executing whois tryhackme.com.

user@TryHackMe$ whois tryhackme.com

[Querying whois.verisign-grs.com]

[Redirected to whois.namecheap.com]

[Querying whois.namecheap.com]

[whois.namecheap.com]

Domain name: tryhackme.com

Registry Domain ID: 2282723194\_DOMAIN\_COM-VRSN

Registrar WHOIS Server: whois.namecheap.com

Registrar URL: http://www.namecheap.com

Updated Date: 2021-05-01T19:43:23.31Z

Creation Date: 2018-07-05T19:46:15.00Z

Registrar Registration Expiration Date: 2027-07-05T19:46:15.00Z

Registrar: NAMECHEAP INC

Registrar IANA ID: 1068

Registrar Abuse Contact Email: abuse@namecheap.com

Registrar Abuse Contact Phone: +1.6613102107

Reseller: NAMECHEAP INC

Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited

Registry Registrant ID:

Registrant Name: Withheld for Privacy Purposes

Registrant Organization: Privacy service provided by Withheld for Privacy ehf

The information collected can be inspected to find new attack surfaces, such as social engineering or technical attacks. For instance, depending on the scope of the penetration test, you might consider an attack against the email server of the admin user or the DNS servers, assuming they are owned by your client and fall within the scope of the penetration test.

It is important to note that due to automated tools abusing WHOIS queries to harvest email addresses, many WHOIS services take measures against this. They might redact email addresses, for instance. Moreover, many registrants subscribe to privacy services to avoid their email addresses being harvested by spammers and keep their information private.

**Answer the question:**

1.When was TryHackMe.com registered?

Ans- 2018-07-05

2.What is the registrar of TryHackMe.com?

Ans- namecheap.com

3.Which company is TryHackMe.com using for name servers?

Ans – cloudflare.com

**Task 4 – nslookup and dig**

In the previous task, we used the WHOIS protocol to get various information about the domain name we were looking up. In particular, we were able to get the DNS servers from the registrar.

Find the IP address of a domain name using nslookup, which stands for Name Server Look Up. You need to issue the command nslookup DOMAIN\_NAME, for example, nslookup tryhackme.com. Or, more generally, you can use nslookup OPTIONS DOMAIN\_NAME SERVER. These three main parameters are:

* **OPTIONS** contains the query type as shown in the table below. For instance, you can use A for IPv4 addresses and AAAA for IPv6 addresses.
* **DOMAIN\_NAME** is the domain name you are looking up.
* **SERVER** is the DNS server that you want to query. You can choose any local or public DNS server to query. Cloudflare offers 1.1.1.1 and 1.0.0.1, Google offers 8.8.8.8 and **8.8.4.4**, and Quad9 offers 9.9.9.9 and 149.112.112.112. There are many more public DNS servers that you can choose from if you want alternatives to your ISP’s DNS servers

Query type Result

A IPv4 Addresses

AAAA IPv6 Addresses

CNAME Canonical Name

MX Mail Servers

SOA Start of Authority

TXT TXT Records

For instance, nslookup -type=A tryhackme.com 1.1.1.1 (or nslookup -type=a tryhackme.com 1.1.1.1 as it is case-insensitive) can be used to return all the IPv4 addresses used by tryhackme.com.

user@TryHackMe$ nslookup -type=A tryhackme.com 1.1.1.1

Server: 1.1.1.1

Address: 1.1.1.1#53

Non-authoritative answer:

Name: tryhackme.com

Address: 172.67.69.208

Name: tryhackme.com

Address: 104.26.11.229

Name: tryhackme.com

Address: 104.26.10.229

The A and AAAA records are used to return IPv4 and IPv6 addresses, respectively. This lookup is helpful to know from a penetration testing perspective. In the example above, we started with one domain name, and we obtained three IPv4 addresses. Each of these IP addresses can be further checked for insecurities, assuming they lie within the scope of the penetration test.

Let’s say you want to learn about the email servers and configurations for a particular domain. You can issue nslookup -type=MX tryhackme.com. Here is an example:

user@TryHackMe$ nslookup -type=MX tryhackme.com

Server: 127.0.0.53

Address: 127.0.0.53#53

Non-authoritative answer:

tryhackme.com mail exchanger = 5 alt1.aspmx.l.google.com.

tryhackme.com mail exchanger = 1 aspmx.l.google.com.

tryhackme.com mail exchanger = 10 alt4.aspmx.l.google.com.

tryhackme.com mail exchanger = 10 alt3.aspmx.l.google.com.

tryhackme.com mail exchanger = 5 alt2.aspmx.l.google.com.

We can see that tryhackme.com’s current email configuration uses Google. Since MX is looking up the Mail Exchange servers, we notice that when a mail server tries to deliver email @tryhackme.com, it will try to connect to the aspmx.l.google.com, which has order 1. If it is busy or unavailable, the mail server will attempt to connect to the next in order mail exchange servers, alt1.aspmx.l.google.com or alt2.aspmx.l.google.com.

Such pieces of information might prove valuable as you continue the passive reconnaissance of your target. You can repeat similar queries for other domain names and try different types, such as -type=txt. Who knows what kind of information you might discover along your way!

For more advanced DNS queries and additional functionality, you can use dig, the acronym for “Domain Information Groper,” if you are curious. Let’s use dig to look up the MX records and compare them to nslookup. We can use dig DOMAIN\_NAME, but to specify the record type, we would use dig DOMAIN\_NAME TYPE. Optionally, we can select the server we want to query using dig @SERVER DOMAIN\_NAME TYPE

A quick comparison between the output of nslookup and dig shows that dig returned more information, such as the TTL (Time To Live) by default. If you want to query a 1.1.1.1 DNS server, you can execute dig @1.1.1.1 tryhackme.com MX.

Q:

Check the TXT records of thmlabs.com. What is the flag there?

Flag - THM{a5b83929888ed36acb0272971e438d78}

Tip: dig thmlabs.com TXT # this is the command that you can get flag!

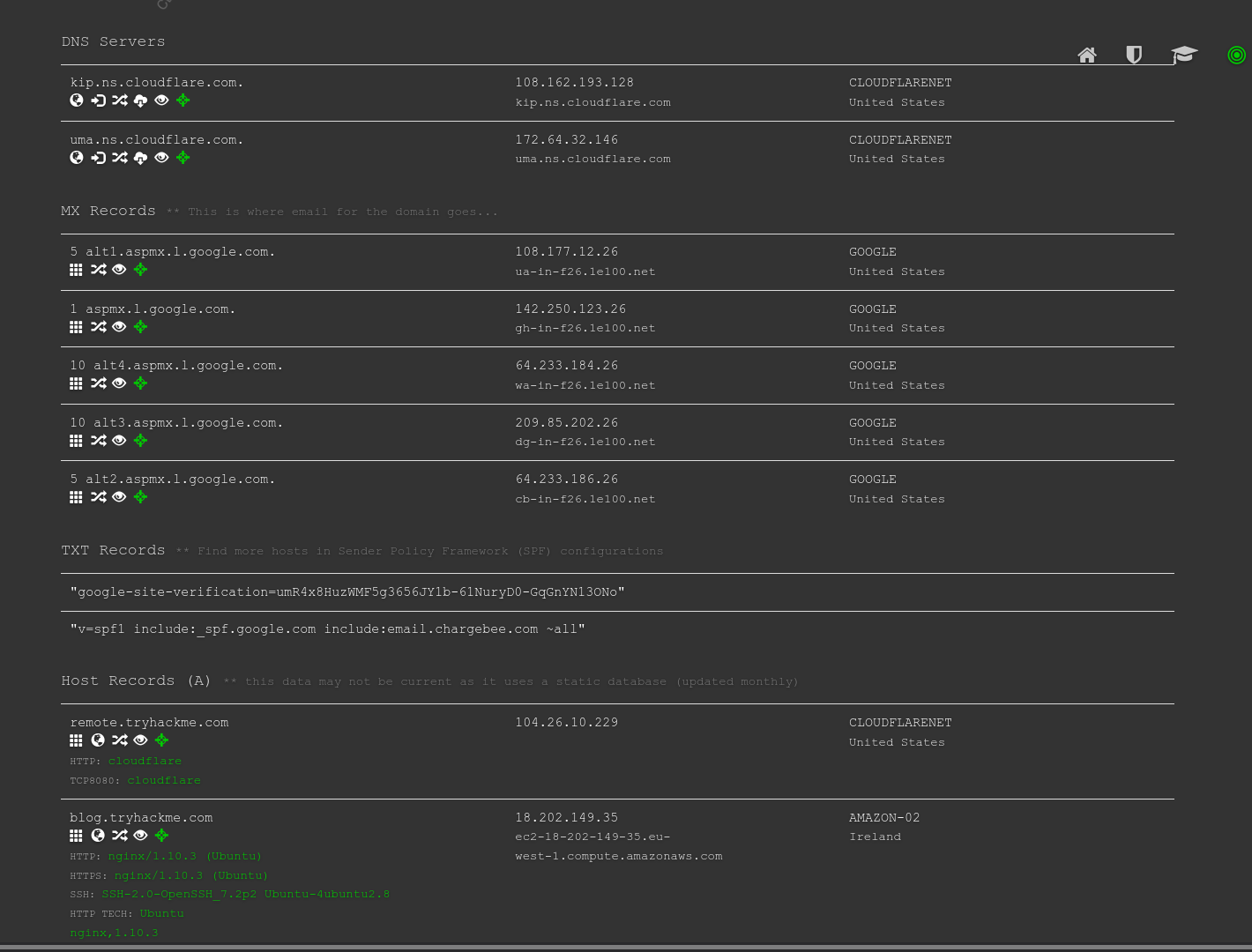
**Task 5 - DNSDumpster**

DNS lookup tools, such as nslookup and dig, cannot find subdomains on their own. The domain you are inspecting might include a different subdomain that can reveal much information about the target. For instance, if tryhackme.com has the subdomains wiki.tryhackme.com and webmail.tryhackme.com, you want to learn more about these two as they can hold a trove of information about your target. There is a possibility that one of these subdomains has been set up and is not updated regularly. Lack of proper regular updates usually leads to vulnerable services. But how can we know that such subdomains exist?

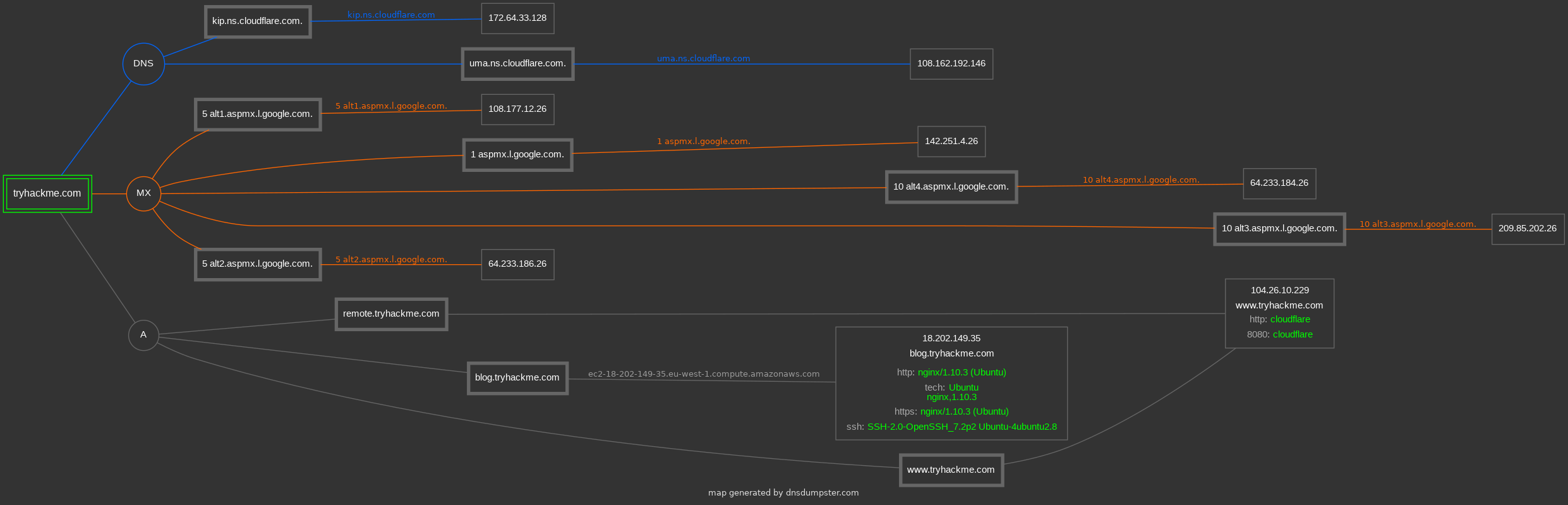
We can consider using multiple search engines to compile a list of publicly known subdomains. One search engine won’t be enough; moreover, we should expect to go through at least tens of results to find interesting data. After all, you are looking for subdomains that are not explicitly advertised, and hence it is not necessary to make it to the first page of search results. Another approach to discover such subdomains would be to rely on brute-forcing queries to find which subdomains have DNS records.

To avoid such a time-consuming search, one can use an online service that offers detailed answers to DNS queries, such as DNSDumpster. If we search DNSDumpster for tryhackme.com, we will discover the subdomain blog.tryhackme.com, which a typical DNS query cannot provide. In addition, DNSDumpster will return the collected DNS information in easy-to-read tables and a graph. DNSDumpster will also provide any collected information about listening servers.

We will search for tryhackme.com on DNSDumpster to give you a glimpse of the expected output. Among the results, we got a list of DNS servers for the domain we are looking up. DNSDumpster also resolved the domain names to IP addresses and even tried to geolocate them. We can also see the MX records; DNSDumpster resolved all five mail exchange servers to their respective IP addresses and provided more information about the owner and location. Finally, we can see TXT records. Practically a single query was enough to retrieve all this information.



DNSDumpster will also represent the collected information graphically. DNSDumpster displayed the data from the table earlier as a graph. You can see the DNS and MX branching to their respective servers and also showing the IP addresses.

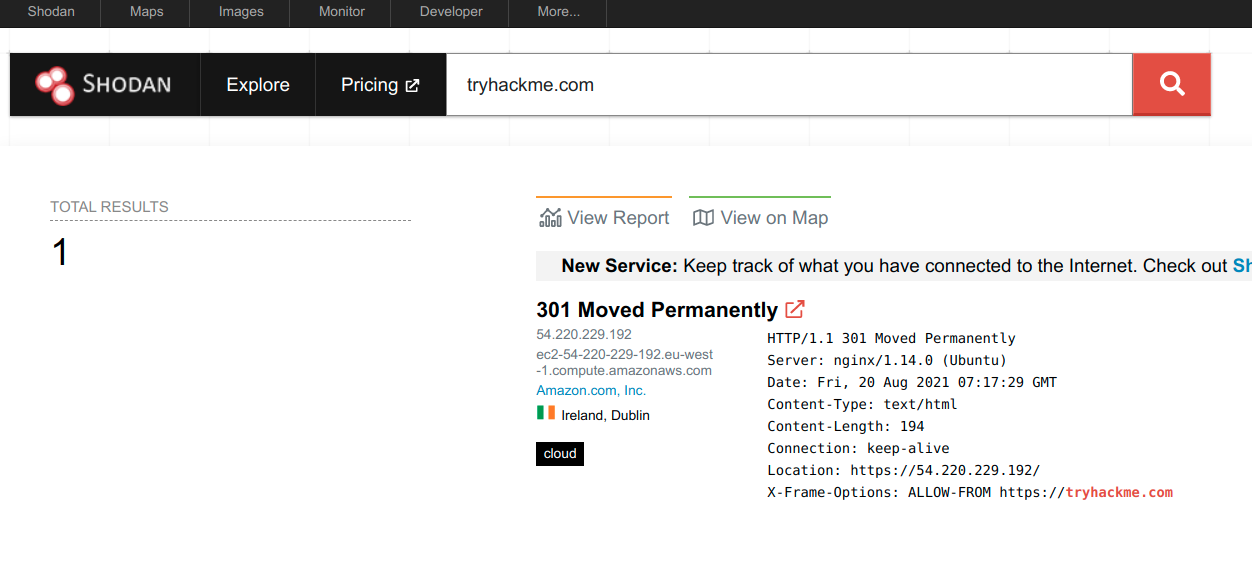


There is currently a beta feature that allows you to export the graph as well. You can manipulate the graph and move blocks around if needed.

**Task 6 – Shodan.io**

When you are tasked to run a penetration test against specific targets, as part of the passive reconnaissance phase, a service like Shodan.io can be helpful to learn various pieces of information about the client’s network, without actively connecting to it. Furthermore, on the defensive side, you can use different services from Shodan.io to learn about connected and exposed devices belonging to your organization.

Shodan.io tries to connect to every device reachable online to build a search engine of connected “things” in contrast with a search engine for web pages. Once it gets a response, it collects all the information related to the service and saves it in the database to make it searchable. Consider the saved record of one of tryhackme.com’s servers.



This record shows a web server; however, as mentioned already, Shodan.io collects information related to any device it can find connected online. Searching for tryhackme.com on Shodan.io will display at least the record shown in the screenshot above. Via this Shodan.io search result, we can learn several things related to our search, such as:

* IP address
* hosting company
* geographic location
* server type and version

**Answer the question:**

Q1. According to Shodan.io, what is the 2nd country in the world in terms of the number of publicly accessible Apache servers?

Ans- Germany # search on shodan all answer

Q2. Based on Shodan.io, what is the 3rd most common port used for Apache?

Ans – 8080

Q3. Based on Shodan.io, what is the 3rd most common port used for nginx?

Ans- 8888