

# Shodan - The Complete Guide, Featured on TryHackMe

January 1, 2021 · 10 min · Bee

▶ [Table of Contents](#)

Shodan.io is a search engine for the Internet of Things.

Ever wondered how you can find publicly accessible CCTV cameras? What about finding out how many Pi-Holes are publicly accessible?

Or whether your office coffee machine is on the internet?

Shodan.io is the answer!

Shodan scans the whole internet and indexes the services run on each IP address.

Note: if you are following along, you'll need a premium Shodan account.

## Finding services

Let's say we are performing a pentest on a company, and we want to find out what services one of their servers run.

We need to grab their IP address. We can do this using `ping`.

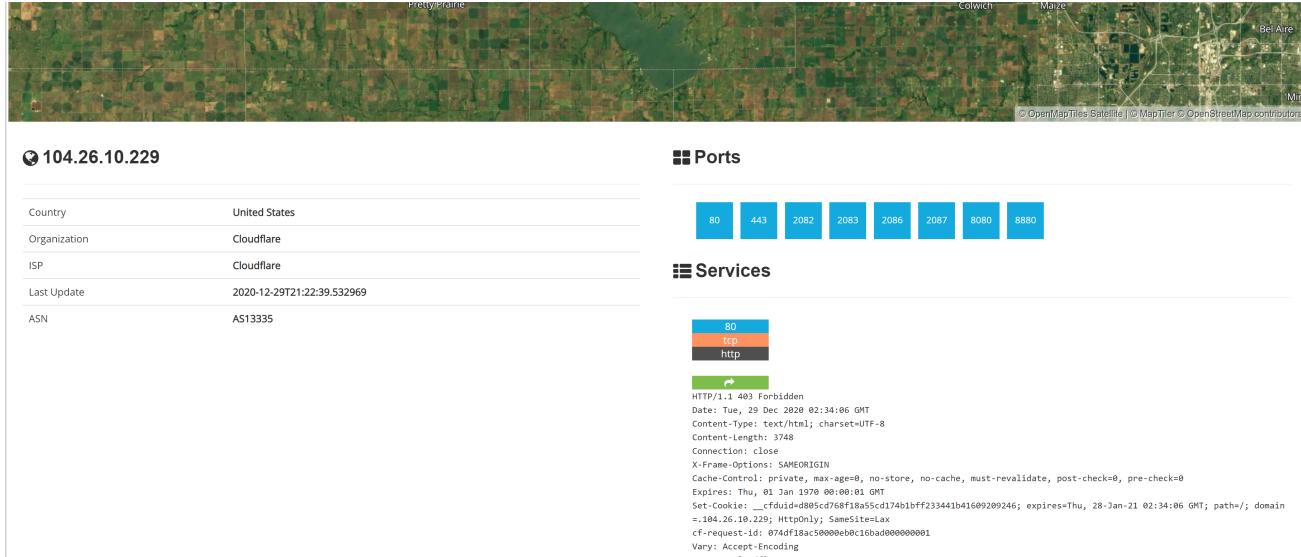
We can ping `tryhackme.com` and the ping response will tell us their IP address.

`Pinging tryhackme.com [142.93.194.248] with 32 bytes of data:`



Then once we do this, we put the IP address into Shodan to get:





We can see that TryHackMe runs on Cloudflare in the United States and they have many ports open.

Cloudflare acts as a proxy between TryHackMe and their real servers. If we were pentesting a large company, this isn't helpful. We need some way to get their IP addresses.

We can do this using Autonomous System Numbers.

## Autonomous System Numbers

An autonomous system number (ASN) is a global identifier of a range of IP addresses. If you are an enormous company like Google you will likely have your own ASN for all of the IP addresses you own.

We can put the IP address into an ASN lookup tool such as <https://www.ultratools.com/tools/asnInfo>,

Which tells us they have the ASN AS14061.

Tryhackme isn't a mega large corporation, so they don't own their own ASN. When we google AS14061 we can see it is a DigitalOcean ASN number.

On Shodan.io, we can search using the ASN filter. The filter is ASN:[number] where number is the number we got from earlier, which is AS14061.

Doing this, we can see a whole range 6.2 million websites, in fact) that are on this

one single ASN!

<https://www.shodan.io/search?query=asn%3AAS14061>

The screenshot shows the Shodan search interface with the query `asn:AS14061`. The results page displays the following information:

- TOTAL RESULTS:** 6,253,778
- TOP COUNTRIES:**

Country	Count
United States	2,605,473
Germany	892,469
Singapore	774,949
United Kingdom	675,308
Netherlands	649,002
- TOP SERVICES:**

Service	Count
SSH	1,242,891
HTTP	1,167,167
HTTPS	937,184
MySQL	108,177
SMTP	101,908
- TOP ORGANIZATIONS:**

Organization	Count
Digital Ocean	5,417,532
DigitalOcean	619,285
DigitalOcean, LLC	216,105
Tiekoetter.NET	44
SMV Host	6
- TOP OPERATING SYSTEMS:**

OS	Count
Windows Server 2012 R2 Datacenter	2,381
Windows Server 2012 R2 Standard Evaluate...	1,734
Windows Server 2012 R2 Standard	1,124
Windows 8.1	978
Windows Server 2012 R2 Datacenter Evalu...	666
- TOP PRODUCTS:**

Product	Count
Digital Ocean	162,243,249.12
DigitalOcean	138.197.54.191

Two detailed result cards are shown:

- 162.243.249.12** (Digital Ocean):
  - Added on 2020-12-30 00:23:15 GMT
  - United States, Clifton
  - Cloud

**Banner:**

```
220 (vsFTPD 3.0.3)
530 Login incorrect.
530 Please login with USER and PASS.
211-features:
  EPRT
  EPSV
  MDTM
  PASV
  REST STREAM
  SIZE
  TVFS
  211 End
```
- 138.197.54.191** (Digital Ocean):
  - Added on 2020-12-30 00:23:43 GMT
  - United States, Clifton
  - Cloud

**Banner:**

```
HTTP/1.1 400 Bad Request
Date: Wed, 30 Dec 2020 00:22:30 GMT
Server: Apache/2.4.46 (Ubuntu) OpenSSL/1.1.1i
X-Powered-By: PHP/7.4.13
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Security-Policy: default-src 'self'; script-s...
```

**SSL Certificate:**

```
SSL Certificate
Issued By: R3
Common Name: Let's Encrypt
Organization: Let's Encrypt
Issued To: battleblock.mooo.com
Common Name: battleblock.mooo.com
Supported SSL Versions: TLSv1.2, TLSv1.3
```

**Diffie-Hellman Parameters:**

```
Fingerprint: RFC3526/Oakley Group
14
```

```
HTTP/1.1 302 Found
Date: Wed, 30 Dec 2020 00:20:44 GMT
Server: Apache
Vary: Accept-Encoding
X-Powered-By: PHP/5.3.29
Location: http://rschooltoday.com
Content-Length: 0
Connection: close
Content-Type: text/html
```

Knowing the ASN is helpful, because we can search Shodan for things such as coffee makers or vulnerable computers within our ASN, which we know (if we are a large company) is on our network.

## Getting started

Time to dig in! If you get stuck, look at the previous task for some help! :)

## Banners

To get the most out of Shodan, it's important to understand the search query syntax.

Devices run services, and Shodan stores information about them. The information is stored in a *banner*. It's the most fundamental part of Shodan.

An example banner looks like:



{

```
"data": "Moxa Nport Device",
"Status": "Authentication disabled",
"Name": "NP5232I_4728",
"MAC": "00:90:e8:47:10:2d",
"ip_str": "46.252.132.235",
"port": 4800,
"org": "Starhub Mobile",
"location": {
    "country_code": "SG"
}

}
```

We're looking at the output of a single port, which includes information about the IP and authentication details.

You don't really see this outside of the API, so we won't delve into it.

## Filters

On the Shodan.io homepage, we can click on “explore” to view the most up voted search queries. The most popular one is webcams.

<https://www.shodan.io/explore>

Note: this is a grey area. It is legal to view a publicly accessible webcam, it is illegal to try to break into a password protected one. Use your brain and research the laws of your country!

One of the other most up voted searches is a search for MYSQL databases.

<https://www.shodan.io/search?query=product%3AMySQL>

If we look at the search, we can see it is another filter.

product:MySQL

Knowing this, we can actually combine 2 searches into 1.

On TryHackMe's ASN, let's try to find some MYSQL servers.



We use this search query

asn:AS14061 product:MySQL

And ta-da! We have MySQL servers on the TryHackMe ASN (which is really the DigitalOcean ASN).

<https://www.shodan.io/search?query=asn%3AAS14061+product%3AMySQL>

Shodan has many powerful filters. My favourite one is the vuln filter, which lets us search for IP addresses vulnerable to an exploit.

Let's say we want to find IP addresses vulnerable to Eternal Blue:

vuln:ms17-010

However, this is only available for academic or business users, to prevent script kiddies from abusing this!

Here are some nice filters we can use on Shodan:

City Country Geo (coordinates) Hostname net (based on IP / CIDR) os (find operating systems) port before/after (timeframes)

## API

Shodan.io has an API! It requires an account, so I won't talk about it here.

If you want to explore the Shodan API, I've written a blog post about finding Pi-Holes with it here:

<https://github.com/beesecurity/How-I-Hacked-Your-Pi-Hole/blob/master/README.md>

The API lets us programmatically search Shodan and receive a list of IP addresses in return. If we are a company, we can write a script to check over our IP addresses to see if any of them are vulnerable.

PS: You can automatically filter on Shodan by clicking the things in the left hand side bar!

# Shodan Monitor

Shodan Monitor is an application for monitoring your devices in your own network. In their words:

Keep track of the devices that you have exposed to the Internet. Setup notifications, launch scans and gain complete visibility into what you have connected.

Previously we had to do this using their API, but now we have this fancy application.

Access the dashboard via this link:

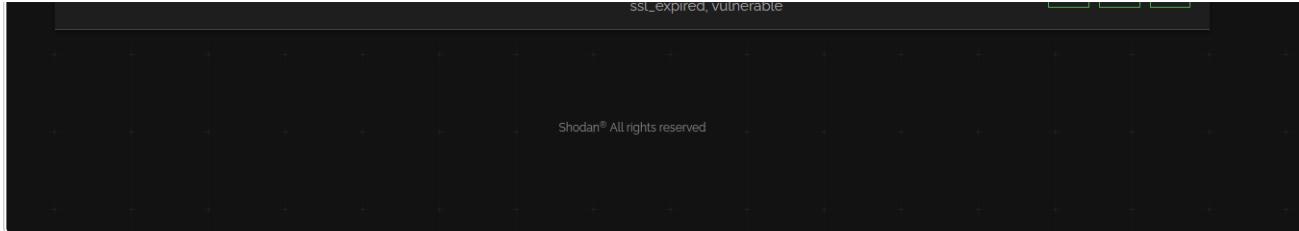
<https://monitor.shodan.io/dashboard>

You'll see it's asking for an IP range.

The screenshot shows the 'Monitor Network' setup page. On the left, there's a 'General Information' section with a 'Name' field containing 'Nmap'. Below it is a large input field showing '45.33.32.156' with a loading icon. To the right, a box displays '16 IPs remaining' and a message about monitoring up to 16 IPs in total, with a link to upgrade from the 'Billing section'.

Once we add a network, we can see it in our dashboard.

The screenshot shows the 'Manage Assets' dashboard. A green success message box at the top right says 'Success: New network range is being monitored'. Below it, the 'Manage Assets' section has 'ADD NETWORK' and 'ADD DOMAIN' buttons. At the bottom, there's a summary row with icons for home, Nmap, IP address (45.33.32.156), and count (1 IP). The IP address is also listed in the middle of the row. The bottom right corner features three small circular icons.



If we click on the settings cog, we can see that we have a range of “scans” Shodan performs against our network.

A screenshot of the Shodan Trigger Rules configuration page. It shows a list of notification types with checkboxes. Most checkboxes are checked, except for "uncommon". A "SAVE CHANGES" button is at the bottom. To the right, there is a sidebar with the heading "What is a trigger?" and a description explaining that triggers are rules that cause Shodan to send notifications if certain conditions are met.

Anytime Shodan detects a security vulnerability in one of these categories, it will email us.

If we go to the dashboard again we can see it lays some things out for us.

A screenshot of the Shodan Dashboard. It includes sections for "Top Open Ports", "Notable Ports", "Top Vulnerabilities", and "Potential Vulnerabilities". The "Top Open Ports" section lists ports 22, 80, and 123. The "Notable Ports" section says "Looking good! No unusual ports exposed to the Internet." The "Top Vulnerabilities" section says "No vulnerabilities identified". The "Potential Vulnerabilities" section lists several CVE entries: cve-2013-6438, cve-2014-0098, cve-2014-0117, cve-2014-0118, cve-2014-0226, and cve-2014-0231.



Most notably:

- Top Open Ports (most common)
- Top Vulnerabilities (stuff we need to deal with right away)
- Notable Ports (unusual ports that are open)
- Potential Vulnerabilities
- Notable IPs (things we should investigate in more depth).

The interesting part is that you can actually monitor other peoples networks using this. For bug bounties you can save a list of IPs and Shodan will email you if it finds any problems.

## Shodan Dorking

Shodan has some lovely webpages with Dorks that allow us to find things. Their search example webpages features some.

Some fun ones include:

`has_screenshot:true encrypted attention`

Which uses optical character recognition and remote desktop to find machines compromised by ransomware on the internet.

The screenshot shows the Shodan search results for the dork "has\_screenshot:true encrypted attention". The search bar contains this query. The results page indicates 8 total results. A world map shows the top countries affected, with China being the most prominent. On the right side of the results page, there is a summary of findings for one specific result, including an SSL certificate section and a vulnerabilities section. A large blue button at the bottom right says "Attention!!! your files are encrypted !!!".



screenshot.label:ics

Using Machine Learning, Shodan can identify industrial control systems which are connected to the internet.

vuln: CVE-2014-0160

Internet connected machines vulnerable to heartbleed. Note: CVE search is only allowed to academic or business subscribers.

Solar Winds Supply Chain Attack by using Favicons:

<http://favicon.hash:-1776962843>

You can find more Shodan Dorks on GitHub or in Shodan's Explore Page

# Shodan Extension

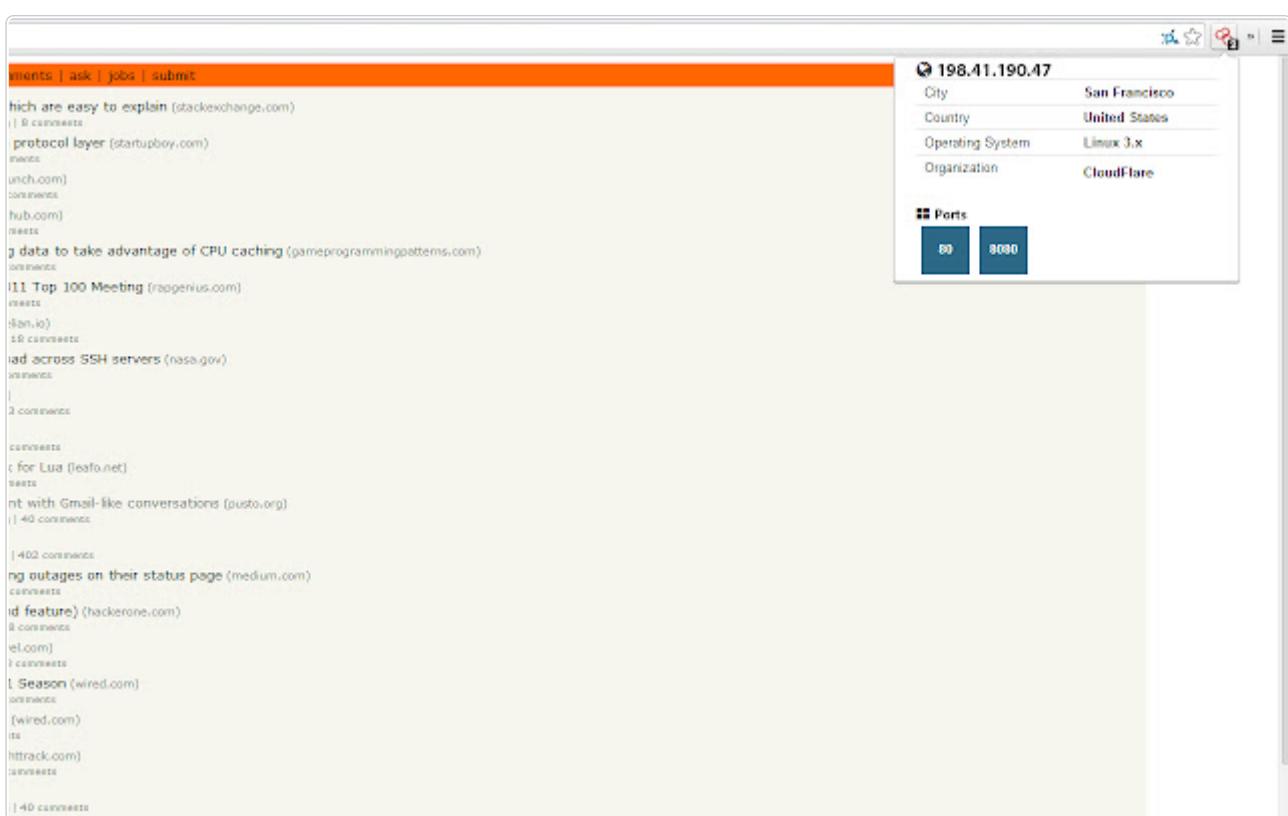


Shodan also has an extension.

<https://chrome.google.com/webstore/detail/shodan/jjalcfnidlmpjhdfepjhjhbnhkbgleap>

When installed, you can click on it and it'll tell you the IP address of the webserver running, what ports are open, where it's based and if it has any security issues.

I imagine this is a good extension for any people interested in bug bounties, being quickly able to tell if a system looks vulnerable or not based on the Shodan output.



## How I Hacked Your Pi-Hole

There are 5308 publically available Pi-Holes according to Shodan.io. This article will demonstrate how bad of an idea this is.

If you've spent any time over at r/pihole, you'll know they always talk about not having publically available Pi-Holes. But, why? What's the harm?

Out of the 5308 Pi-Holes publically available, I found 100 of them are vulnerable.

Vulnerable means:

1. You can access /admin/index.php?login. This is because not all Pi-Holes listed by Shodan work in the way we expect them to.

However, in my research I came across something interesting:

## Passwordless Pi-Holes

I came across many passwordless Pi-Holes. 578 passwordless Pi-Holes to be correct. These aren't considered "vulnerable", because they are by default open. It's incredibly easy to find these, if you click around on Shodan enough you'll find them. You can also search the Shodan API (like I did below) and try to access a page, /admin/queries.php and see if it doesn't prompt for a password. Some of these Pi-Holes are meant to be public. You can tell because their domain name includes "public-pihole".

It's hard to differentiate between Pi-Holes meant to be publically accessible and those that aren't, so I haven't explored these much. Just know that these exist and to not make a publically facing Pi-Hole without a password for your personal use.

## Finding these Pi-Holes

Shodan.io is a service that scans the web. It finds IoT or other devices like Pi-Hole. Using the Shodan API, we can programatically explore these Pi-Holes. Or, you can click [here](#) and explore them manually.

Finding these Pi-Holes took a minimal amount of code. I was surprised to find my Pi-Hole on this list. You need a Shodan membership. Also, don't attack Pi-Holes you don't own.

```
from shodan import Shodan
import requests
api = Shodan('API_KEY')
```



```
def url_ok(url):
    r = requests.head("http://" + url)
    return r.status_code == 200

def check_page(url):
    r = requests.get("http://" + url + "/admin/")
    return "Pi-hole" in r.text

def pruneIPs(vulnerableIPs):
    for i in vulnerableIPs:
        if not url_ok(i):
            if not check_page(i):
                vulnerableIPs.remove(i)
    return vulnerableIPs

result = api.search("pi-hole")

VulnerableIP = []
for service in result['matches']:
    VulnerableIP.append(service['ip_str'])
```

# Hacking the Pi-Hole

Pi-Hole doesn't block bruteforcing. You can enter 200,000 incorrect passwords and Pi-Hole wouldn't care. How easy is it to create a brute-force attack? With Hydra - very easy. Hydra is a brute-forcing tool that uses a dictionary to attack a target. The payload for brute-forcing a Pi-Hole is:

```
hydra -l '' -P /usr/share/wordlists/rockyou.txt 192.168.0.1 http-post-form ''
```

Our wordlist is the infamous rockyou.txt. We use the http-post-form module and enter some information. The form to post, the name of the variable for the password ("pw"), and what will be on the page when a failed login attempt happens (the forgotten password box).

Since Pi-Hole doesn't block brute-force attacks, it makes it trivial to brute-force most of the Pi-Holes assuming password length is low (under 9 chars optimally, although 9 chars can be done in 19 hours <https://howsecureismypassword.net/>). It's worth mentioning that Pi-Hole's default password is very secure, but the lack

of any timeouts sucks. Not to mention that most people change their password to something more human.

To recap, it is possible to get a list of all the vulnerable Pi-Holes and to brute-force their passwords with a dictionary attack.

I will not show code for this, because it is illegal. However, you can see the 2 separate parts in action above. Please test your own Pi-Hole. Do not test a Pi-Hole you do not own.

## What can an attacker see?

Once the attacker has gained access to your Pi-Hole, they can see every website you visit. Your hosts' files, your clients and their IP addresses. Your top domains and your top blocked domains.

Now, they have a pretty good idea of who you are. They know all your devices. They know your password (which, most people would reuse on at least one of their devices). And they know you, pretty well actually. Considering your entire internet history is there. **There is no point in using a Pi-Hole if all of your DNS information is as easily accessible as this.**

## DNS Amplification Attacks

An attacker could use your DNS server to perform a DDoS attack. This is very common and is called a DNS amplification attack. This is likely illegal for you to allow this, depending on the country. Another reason to not use a publically facing Pi-Hole.

## How to protect yourself

Search your IP address on Shodan.io. Read the output. Shodan will tell you what ports are open, and will let you know whether it thinks a Pi-Hole service is

running.

1. Use a VPN to connect to your Pi-Hole: [https://www.reddit.com/r/pihole/comments/bl4ka8/guide\\_pihole\\_on\\_the\\_go\\_with\\_wireguard/](https://www.reddit.com/r/pihole/comments/bl4ka8/guide_pihole_on_the_go_with_wireguard/)
2. Don't have a publically facing Pi-Hole
3. Choose a very strong password
4. Turn your Pi-Hole into a no-logs Pi-Hole

## Fun Facts

Statistically, most people with exploitable Pi-Holes used Deutsche Telekom AG. However, most publically accessible Pi-Holes are hosted on:

1. Digital Ocean
2. OVH SAS
3. Google Cloud
4. Deutsche Telekom AG

Infosec

