Shodan InternetDB

High-Throughput, Low-Latency IP Enrichment

OVERVIEW

The InternetDB provides a single-file, drop-in Shodan database that contains information about recently-seen banners for devices and lets you do fast IP lookups. Similar to a local GeoIP database but instead of providing location information we provide network information.

The following properties are currently provided by InternetDB:

- Open ports
- Vulnerabilities
- Hostnames
- CPEs
- Tags

Having the database locally available lets you do the following in real-time:

- Enrich netflow
- Block connections from IPs with known vulnerabilities or that have been compromised
- Block connections from 3rd-party VPNs, proxies
- Block outgoing connections to insecure IPs
- Real-time IP enrichment from Shodan partners

PERFORMANCE

Shodan API	**Shodan API (Bulk IP)**	**InternetDB (SQLite)**	**InternetDB (RocksDB)**
1 IP per second	100 IPs per second	60,000 IPs per second	40,000 IPs per second

The Shodan API allows 1 request per second. In practice, it can take longer than 1 second to grab the data for a host depending on latency, number of banners and other factors. For the purpose of this comparison, we assumed that you can get the maximum of 1 IP per second.

The InternetDB was loaded on a commodity SSD and we performed 10,000 IP lookups using individual queries. For the SQLite version of the database the lookups took the form of:

```
SELECT * FROM data WHERE ip=?
```

It could be further optimized by loading the entire database into memory, putting it on faster drive or batching the IP lookups.

For RocksDB, we used the Python bindings for the database with its default options aside from setting the `read_only` flag to `True`. Every IP lookup was done with a single `get()` request. This could be further optimized by increasing its cache, enabling bloom filters or loading it from a faster drive.

DATABASE SIZE

The database is currently provided in 2 different file formats: SQLite and RocksDB. They both contain the same data and the choice of file format depends on the environment. The SQLite version of the database is 14 GB at the moment while the RocksDB version is 4 GB. If storage space is the paramount consideration then we would recommend using the RocksDB format. Otherwise, the SQLite database is recommended as it provides more flexibility.

QUICKSTART

For most systems, there will not be much to install. You only need a way to download the InternetDB SQLite file and a cronjob to periodically update it. Here's how you could do it using the official Shodan CLI:

1. Install the Shodan command-line interface (CLI):

   ```
   pip install --user shodan
   ```

2. Initialize the CLI using your Shodan API key. You can get your API key from the Shodan account website (https://account.shodan.io):

   ```
   shodan init API_KEY
   ```

3. Download the file:

   ```
   shodan data download internetdb internetdb.sqlite.bz2
   ```

4. Rename and uncompress the file:

   ```
   mv internetdb-internetdb.sqlite.bz2 internetdb.sqlite.bz2
   bunzip2 internetdb.sqlite.bz2
   ```

In production, you would want to have a script that downloads the latest version once a week and rotates it. If you want help getting that configured in your environment simply reach out to us at enterprise@shodan.io

A simple version of such a script would look like the following:

```
#!/bin/sh

# We will store the data file in the /usr/local/shodan directory
mkdir -p /usr/local/shodan

# Download the InternetDB SQLite version and store it in the /tmp folder
shodan download -O /tmp/internetdb.sqlite.bz2 internetdb internetdb.sqlite.bz2

# Rotate out the files
bunzip2 /tmp/internetdb.sqlite.bz2
mv /tmp/internetdb.sqlite /usr/local/shodan/
```

Code Samples

Python

SQLite libraries are available for all programming languages. By default, Python includes a SQLite library that can be used with InternetDB. Below is sample code for accessing the SQLite version of InternetDB and grabbing information about an IP:

```python
import sqlite3

# Load the SQLite database
con = sqlite3.connect('internetdb.sqlite')
cur = con.cursor()

# Grab information about 1.1.1.1
cur.execute("SELECT ip,ports,hostnames,tags,vulns,cpes FROM data WHERE ip=?", ("1.1.1.1",)
)
info = cur.fetchone()
print(info)
```

Command-Line Interface

Many Linux distributions include the sqlite3 command which can be used to access the database:

```
$ sqlite3 internetdb.sqlite
SQLite version 3.24.0 2018-06-04 19:24:41
Enter ".help" for usage hints.
sqlite> SELECT * FROM data WHERE ip='1.1.1.1';
1.1.1.1|53,80,443||||one.one.one.one
sqlite>
```

Quick Links

Enterprise Portal:	https://enterprise.shodan.io	Developer Documentation:	https://developer.shodan.io
Help Center:	https://help.shodan.io	Search engine:	https://www.shodan.io
Videos:	https://asciinema.org/~Shodan	Sales:	sales@shodan.io

Contact us at enterprise@shodan.io with any questions or suggestions – we're here to help!