

# Task 2: Shodan Dorks for Reconnaissance

Maheshwar Anup

July 13, 2025

## Contents

<b>1</b>	<b>Task 2</b>	<b>2</b>
<b>2</b>	<b>Steps</b>	<b>2</b>
2.1	Used Kali GPT (Chat GPT) to generate Shodan dorks. . . . .	2
2.2	Used the dorks to search on Shodan. . . . .	5
2.3	Basic Brute Force . . . . .	6
2.4	BONUS . . . . .	6
<b>3</b>	<b>Conclusion</b>	<b>6</b>

## 1 Task 2

To use Shodan dorks to find login pages and other juicy stuff.

## 2 Steps

### 2.1 Used Kali GPT (Chat GPT) to generate Shodan dorks.

#### ◆ Basic Syntax Recap

```
1 keyword filter:value
```

Listing 1: Basic Shodan Dork Syntax

Common filters:

- port:
- country:
- org:
- os:
- product:
- hostname:
- title:
- http.html:
- ssl:
- before:
- after:

#### ◆ 1. Unprotected Devices and Databases Exposed MongoDB:

```
1 product:"MongoDB" port:27017 -authentication
```

Open Elasticsearch (no auth):

```
1 port:9200 json
```

Redis with no auth:

```
1 product:"Redis" port:6379
```

### CouchDB exposed with Fauxton:

```
1 "Welcome to Fauxton" port:5984
```

### Kibana dashboards (elastic):

```
1 title:"Kibana" port:5601
```

### MySQL with root access:

```
1 product:"MySQL" port:3306
```

## ◆ 2. Open Cameras / Surveillance

### General webcams (MJPEG streams):

```
1 port:554 has_screenshot:true
```

### Axis IP cameras:

```
1 "Server: Boa/0.94.13" "200 OK"
```

### AvTech DVRs (often default creds):

```
1 port:80 title:"VideoWebServer"
```

### RTSP live stream cameras:

```
1 port:554 state:"open"
```

## ◆ 3. Login Portals with Weak Configs

### Web login panels over HTTP:

```
1 http.title:"Login" http.html:"password" port:80
```

### Admin panels:

```
1 http.title:"Admin" OR http.title:"Dashboard"
```

### PHPMyAdmin interfaces:

```
1 http.favicon.hash:-293242991
```

### Jenkins dashboards (CI/CD):

```
1 http.title:"Dashboard [Jenkins]"
```

## ◆ 4. Cloud Storage & Git Exposure

### Exposed Git repositories:

```
1 "Index of /.git"
```

**AWS S3 buckets (misconfigured):**

```
1 http.title:"Amazon S3" http.status:200
```

**Open FTP servers:**

```
1 port:21 "220" anonymous
```

◆ **5. Industrial Control Systems (ICS)**  
**SCADA HMI interfaces:**

```
1 title:"HMI" port:8080
```

**Modbus TCP (danger zone):**

```
1 port:502
```

**Hikvision NVR/DVRs:**

```
1 port:8000 Hikvision
```

◆ **6. Miscellaneous Goldmines**  
**Printers leaking documents:**

```
1 port:80 product:"HP" title:"Printer"
```

**VOIP phone interfaces:**

```
1 port:80 title:"Cisco IP Phone"
```

**VPN gateways (often brute-forceable):**

```
1 title:"GlobalProtect Portal"
```

**Apache status pages (info leak):**

```
1 title:"Apache Status" "Server Status"
```

## Ethics Reminder

- ✓ Only use these dorks on assets you own or have explicit permission to test.
- ✗ Unauthorized access or exploitation is illegal and against ethical guidelines.

## 2.2 Used the dorks to search on Shodan.

Even though I couldn't find any juicy stuff, I did find some interesting results including login pages.

Here is an example:

```
1 "Server: Boa/0.94.13" "200 OK" has_screenshot:true
```

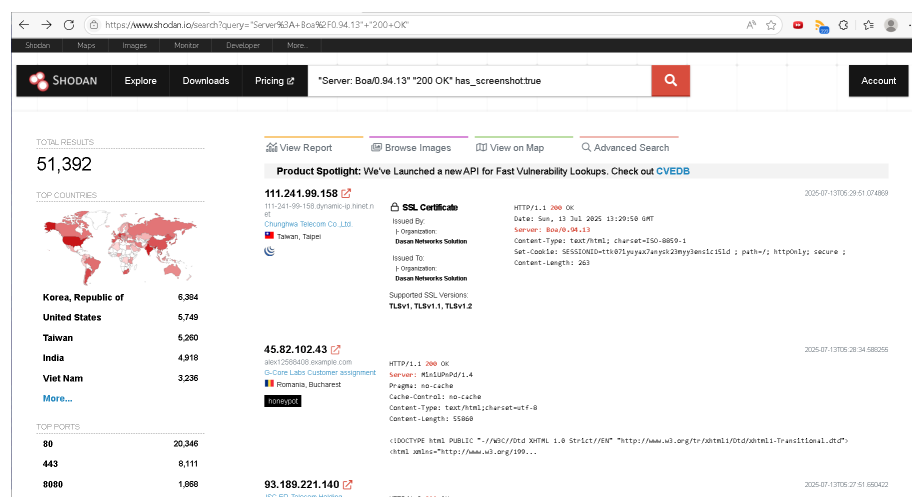


Figure 1: Shodan Search Result Example

Login page:

Incorrect username or password, please try again.

Login to your website

Username \*

admin

Password \*

\*\*\*\*\*

Login Credentials:

Username: admin, Password: password

Login

Figure 2: Login Page Example

## 2.3 Basic Brute Force

- Try using **hydra** or **medusa** to brute force the login page (AT YOUR OWN RISK!!).
- I tried default credentials like **admin:admin**, **root:root**, etc. but it didn't work.

## 2.4 BONUS

Finally just randomly searching for "**robots.txt**" on Shodan gives some interesting results.

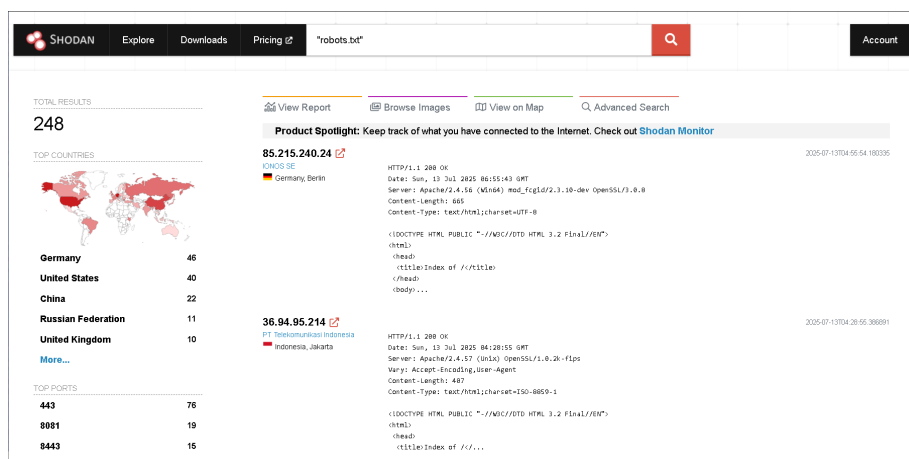


Figure 3: Shodan Search for robots.txt

Got log files:



Figure 4: Log Files Found

Lets see whats inside the log file:

## 3 Conclusion

This task was a great way to learn about Shodan dorks and how to use them to find interesting stuff on the internet. I learned how to use Shodan dorks to

```

-:001/SMS_BILL_IT/6MS_BILL_IT/Log/4-10-2024.txt

4/10/2024 6:49:05 PM:-----
4/10/2024 6:49:05 PM:Sending SMS: Process Started
4/10/2024 6:49:05 PM:-----
4/10/2024 6:49:05 PM:Scheduler Started
4/10/2024 6:49:05 PM:Server=192.168.200.12uidmeshopaid;pmdnunderinitial Catalogmeshopaid_IndianTerrain;
4/10/2024 6:49:05 PM:Connection Started
4/10/2024 6:49:05 PM:Get SMS Details
4/10/2024 6:49:06 PM:URLString from Web.config :https://japl.instaalerts.coma/httpsapi/QuerystringReceiver?
ver=1.0&msg=25V5iKtX0kVvVv&M=dmcrpt&dest=974771745&send=1NG77N&dt_entity_id=@entityid&dt_template_id=@templat&dt_text=@MESSAGE
MESSAGE
4/10/2024 6:49:06 PM:CustomerURL from Web.config :https://japl.instaalerts.coma/httpsapi/QuerystringReceiver?
ver=1.0&msg=25V5iKtX0kVvVv&M=dmcrpt&dest=974771745&send=1NG77N&dt_entity_id=@entityid&dt_template_id=@templat&dt_text=@MESSAGE
4/10/2024 6:49:06 PM:The underlying connection was closed: An unexpected error occurred on a send.
4/10/2024 6:49:07 PM:The underlying connection was closed: An unexpected error occurred on a send.
4/10/2024 6:49:07 PM:Scheduler Started - Main()
4/10/2024 6:49:15 PM:-----
4/10/2024 6:49:15 PM:Sending SMS: Process Started
4/10/2024 6:49:15 PM:-----
4/10/2024 6:49:15 PM:Scheduler Started
4/10/2024 6:49:15 PM:Server=192.168.200.12uidmeshopaid;pmdnunderinitial Catalogmeshopaid_IndianTerrain;
4/10/2024 6:49:15 PM:Connection Started
4/10/2024 6:49:15 PM:Get SMS Details
4/10/2024 6:49:15 PM:URLString from Web.config :https://japl.instaalerts.coma/httpsapi/QuerystringReceiver?
ver=1.0&msg=25V5iKtX0kVvVv&M=dmcrpt&dest=974771745&send=1NG77N&dt_entity_id=@entityid&dt_template_id=@templat&dt_text=@MESSAGE
MESSAGE
4/10/2024 6:49:15 PM:CustomerURL from Web.config :https://japl.instaalerts.coma/httpsapi/QuerystringReceiver?
ver=1.0&msg=25V5iKtX0kVvVv&M=dmcrpt&dest=974771745&send=1NG77N&dt_entity_id=@entityid&dt_template_id=@templat&dt_text=@MESSAGE
4/10/2024 6:49:15 PM:The underlying connection was closed: An unexpected error occurred on a send.
4/10/2024 6:49:15 PM:The underlying connection was closed: An unexpected error occurred on a send.
4/10/2024 6:49:15 PM:Scheduler Started - Main()
4/10/2024 6:50:15 PM:-----
4/10/2024 6:50:15 PM:Sending SMS: Process Started
4/10/2024 6:50:15 PM:-----
4/10/2024 6:50:15 PM:Scheduler Started
4/10/2024 6:50:15 PM:Server=192.168.200.12uidmeshopaid;pmdnunderinitial Catalogmeshopaid_IndianTerrain;
4/10/2024 6:50:15 PM:Connection Started
4/10/2024 6:50:15 PM:Get SMS Details
4/10/2024 6:50:15 PM:-----
4/10/2024 6:50:15 PM:Sending SMS: Process Completed
4/10/2024 6:51:15 PM:-----
4/10/2024 6:51:15 PM:Sending SMS: Process Started
4/10/2024 6:51:15 PM:-----
4/10/2024 6:51:15 PM:Scheduler Started
4/10/2024 6:51:15 PM:Server=192.168.200.12uidmeshopaid;pmdnunderinitial Catalogmeshopaid_IndianTerrain;
4/10/2024 6:51:15 PM:Connection Started
4/10/2024 6:51:15 PM:Get SMS Details
4/10/2024 6:51:15 PM:URLString from Web.config :https://japl.instaalerts.coma/httpsapi/QuerystringReceiver?

```

Figure 5: Content of Log File

find login pages, unprotected devices, and other juicy stuff. I also learned how to use hydra and medusa to brute force login pages. Finally, I learned how to search for robots.txt files on Shodan and found some interesting results.