Flipkart

GRiD 3.0

Problem Statement: Asset Discovery Tool

Team Name   : Helloworld.py

Institute Name: Manipal Institute of Technology

Team members details

Team Name	Helloworld.py		
Institute Name	Manipal Institute of Technology		
Team Members >	**1 (Leader)**	**2**	**3**
Name	Vishesh Tayal	Tishita Goel	Anushka Matta
Batch	4th year ECE (2018-2022)	4th year ECE (2018-2022)	4th year ECE (2018-2022)

Deliverables/Expectations for Level 2 (Idea Submission)

A detailed presentation (Max 10 slides) highlighting the key solution pointers as stated below:

An algorithm/approach with block diagrams and detailed explanation to accomplish the below:

Commands being used with Screenshots to be included in the report. A detailed explanation of what the command is meant to do with explaining all the options/switches of tool.

Running the scanning tool on a remote network would need formation of an SSH tunnel from the ground zero system. The scanning tool needs not to be installed on remote system, rather scan has to be initiated from ground zero system and the scan probes are to be tunneled through SSH tunnel and scan will be performed on the remote network. This will simulate as if the remote system (running SSH server) has initiated the scan. Scan results will be tunnelled back to ground zero system and will be written to a database as per choice below-

- ELK can be used to record the results, provided there is enough scope of correlation of records related to a particular asset. Correlation method also is to be documented.
- In case, a Database/ file is to be used for storing the results, a frontend GUI for interacting with records is also to be mentioned and functioning to be documented.

NOTE:Please provide definitions and working of the core components of the system. You can give references if any part of work is inspired by some previous work. - The solution should work for both a returning user as well as a new user. Considerations of different settings, edge cases will be given extra points.

Glossary

- AD : Active directory
- API : Application Programming Interface
- CRUD : create, read, update and delete
- ICMP : Internet Control Message Protocol
- IP : Internet Protocol
- NMAP : Network Mapper
- OS : Operating System
- SYN : Synchronize
- TCP : Transmission User Protocol
- UDP : User Datagram Protocol
- UI : User Interface

Use-cases

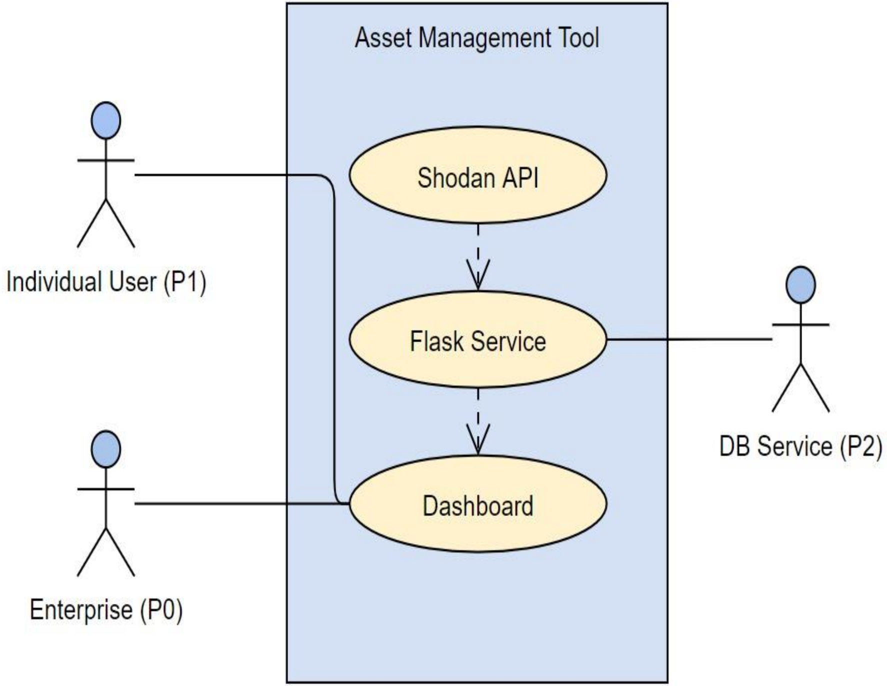
- The user can be either an organization or an individual.

- Application is scalable according to the needs of the user and the size of the network

- User gets to access a highly function dashboard with an easily accessible grid.

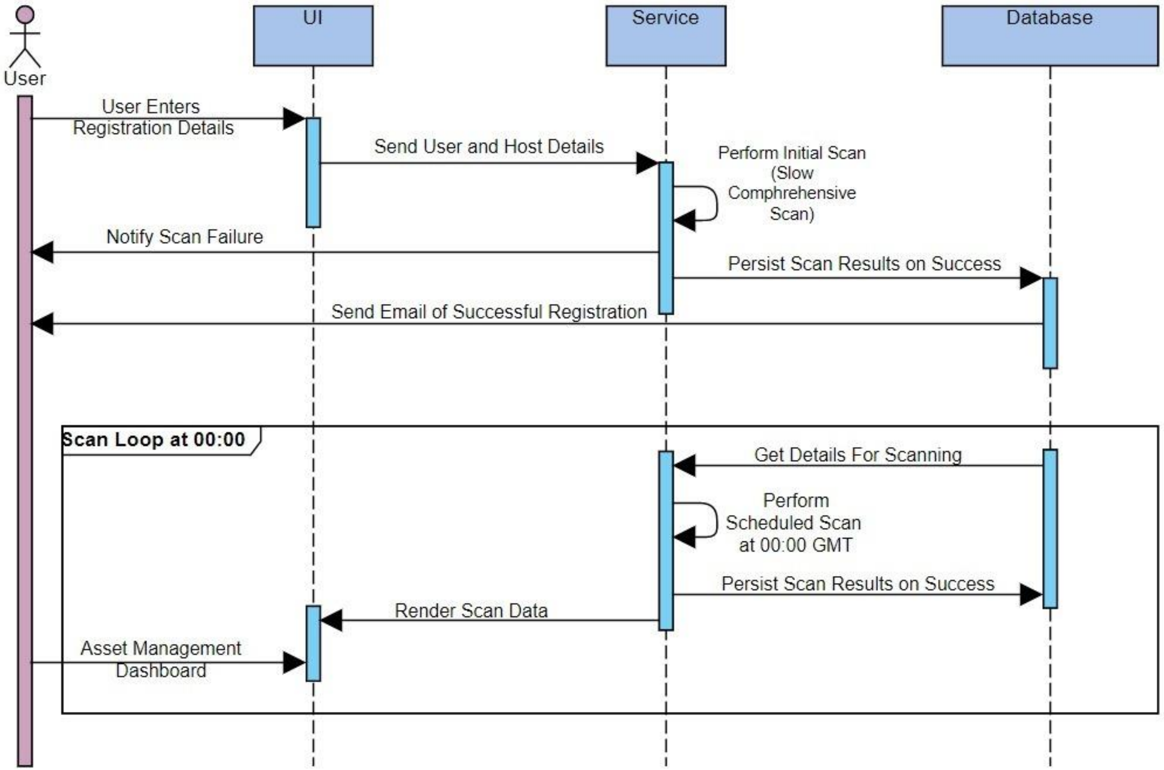
- User gets to modify the scan results as well as make decisions based on the same.

- Dashboard provides full CRUD support to the data displayed.



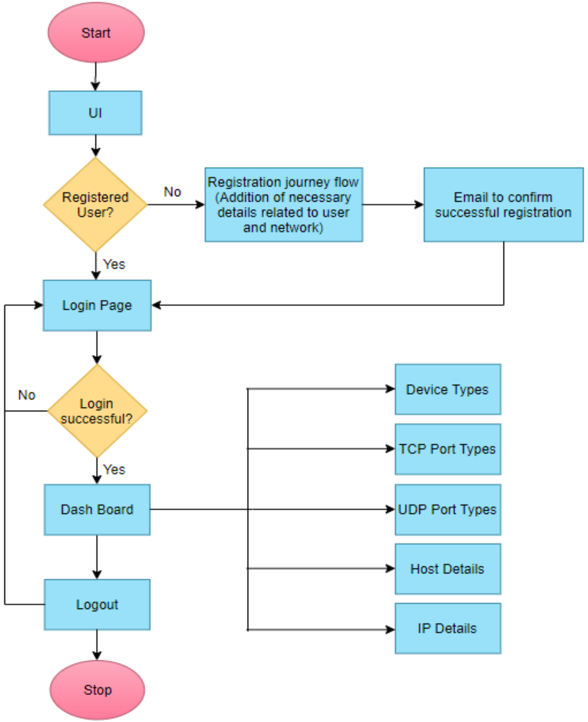
Proposed approach

SEQUENCE DIAGRAM


User
UI
Service
Database
User Enters Registration Details
Send User and Host Details
Perform Initial Scan (Slow Comphrehensive Scan)
Notify Scan Failure
Persist Scan Results on Success
Send Email of Successful Registration
Scan Loop at 00:00
Get Details For Scanning
Perform Scheduled Scan at 00:00 GMT
Persist Scan Results on Success
Render Scan Data
Asset Management Dashboard

Proposed Approach

EXPLANATION



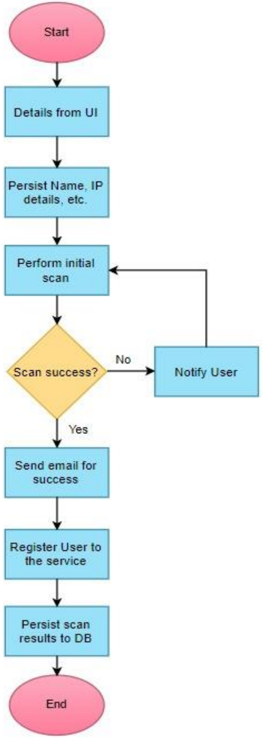
- The application uses NMAP for performing the asset discovery. It scans the network with the slow comprehensive scan preset. The application comprises of a service written in Flask, SQLite as the database, and React as the front end service. The React app has integrated login and registration flow. As a user registers the user is asked for details related to the network. These details are forwarded to the service and an initial scan is performed. As the scan is completed the user receives an email and now obtains access to the React dashboard.

- The initial scan performs OS Discovery, Device Discovery and tunnels scanner for scanning all the ports on the discovered devices. The same is scheduled to be performed everyday at 00:00:00 GMT, where the scan results are persisted to the database.

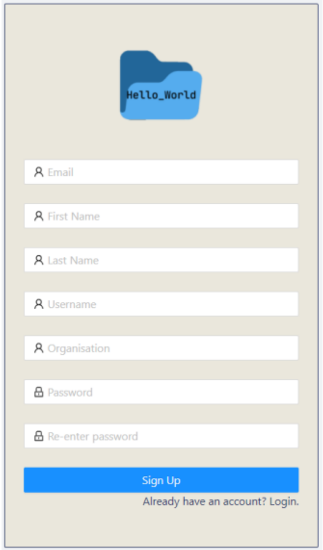
- The dashboard has options to create, read, update and delete the recorded results. It has options to view the scan results in various ways and the segregation is based on the OS Type, Workgroup and AD Domain. Charts displayed on the dashboard are direct reference from the grid corresponding to the same. The user has options to do an IP Address Scan and get the host details with the Shodan API.

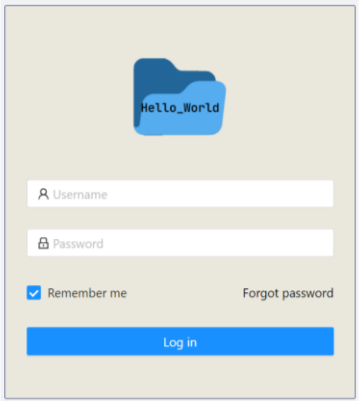
Login and Registration Flow

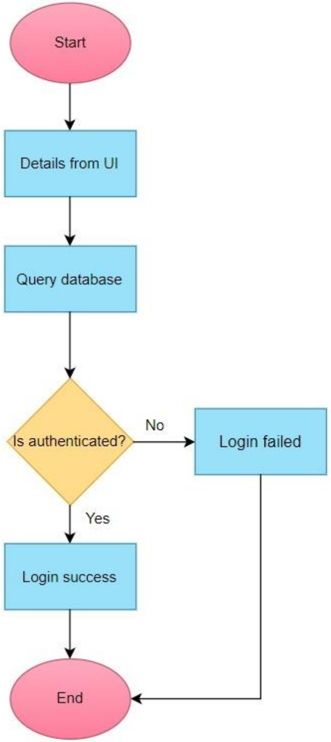
FLOW DIAGRAM



Registration flow



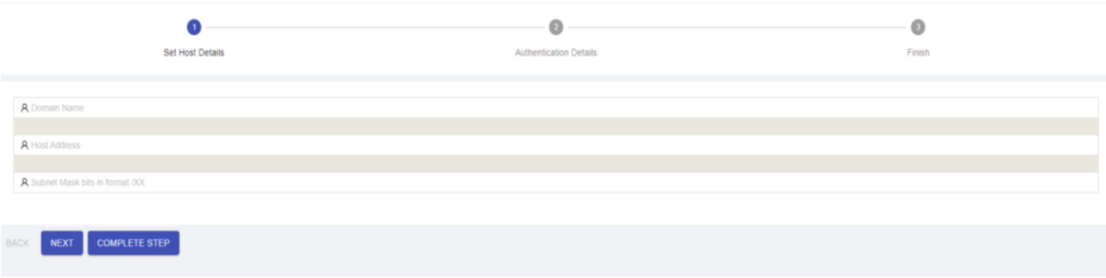


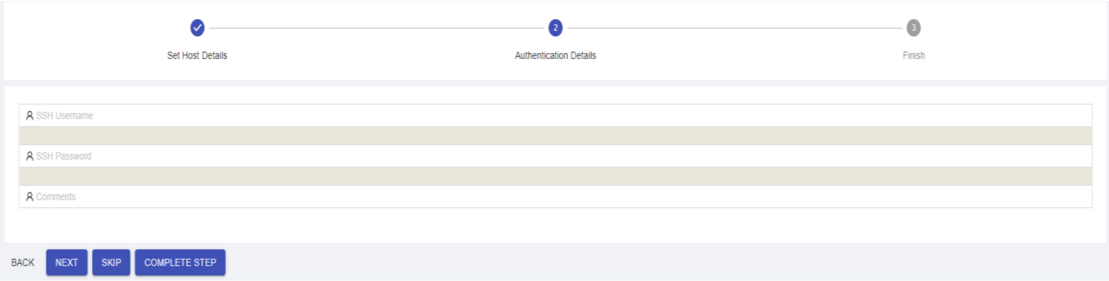


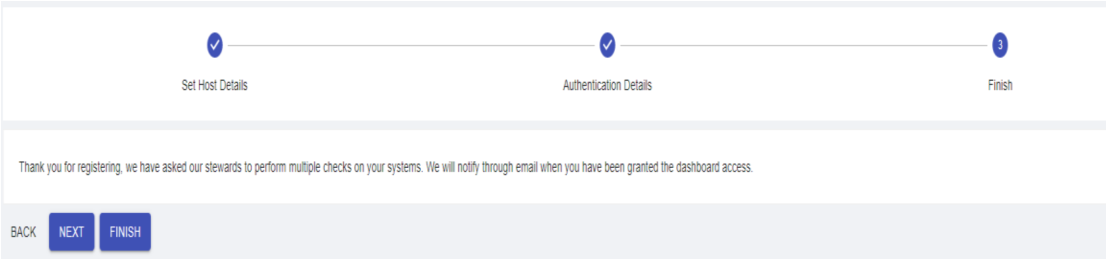
Login flow

Registration and Login Flow

EXPLANATION



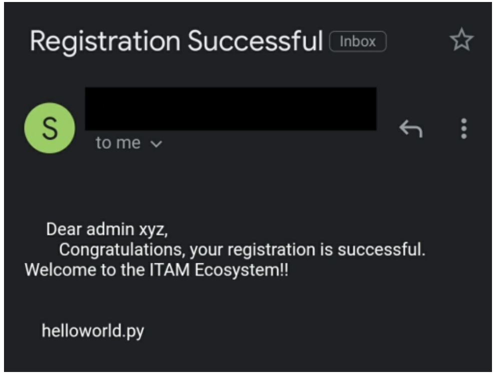




Thank you for registering, we have asked our stewards to perform multiple checks on your systems. We will notify through email when you have been granted the dashboard access.

- The registration flow is a three step process involving:
  1. Entry of basic personal and login details of the user
  2. Entry of data regarding the host and authentication of the same
  3. Email confirmation on successful registration

- In the Login flow, the user enters the details such as username and password that are validated at the server side and the result is sent back to the UI after which the corresponding actions are performed.



Registration Successful Inbox

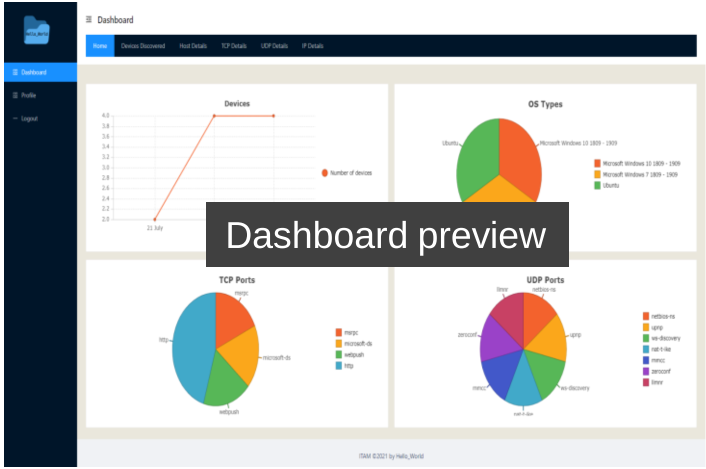
S [redacted]

to me

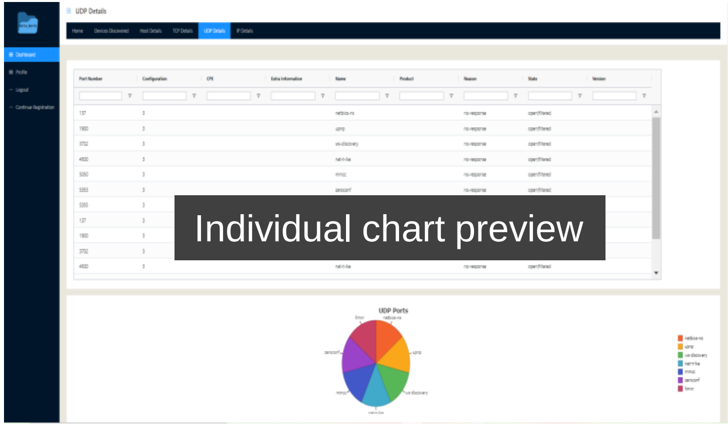
Dear admin xyz,
Congratulations, your registration is successful.
Welcome to the ITAM Ecosystem!!

helloworld.py

Sample Screenshots

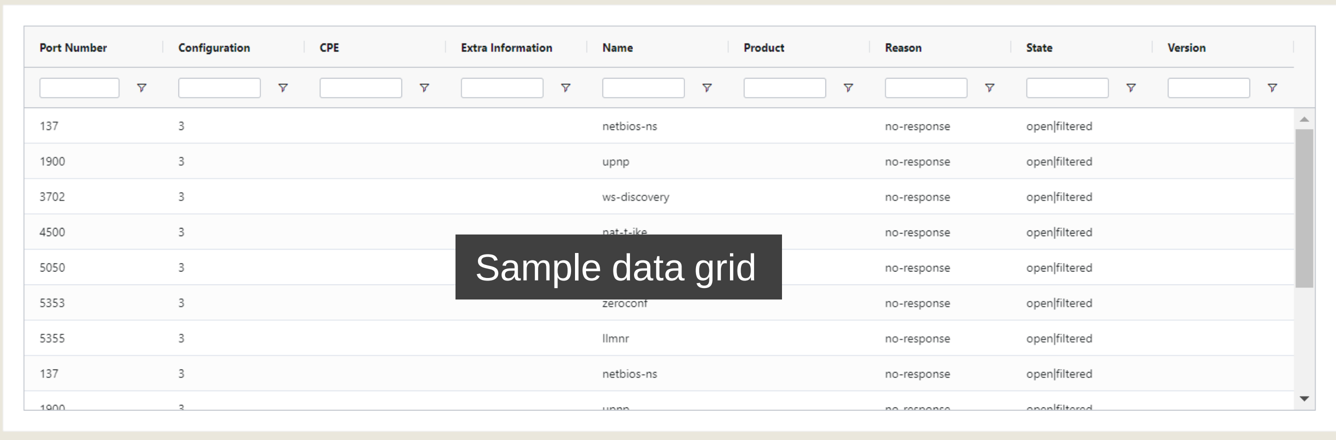


Dashboard preview



Port Number	Configuration	CPE	Extra Information	Name	Product	Reason	State	Version
137	3			netbios-ns		no-response	open|filtered	
1900	3			upnp		no-response	open|filtered	
3702	3			ws-discovery		no-response	open|filtered	
4500	3			nat-t-ike		no-response	open|filtered	
5050	3			mmcc		no-response	open|filtered	
5353	3			zeroconf		no-response	open|filtered	
5355	3			[illegible]		[illegible]	[illegible]	
137	3			[illegible]		[illegible]	[illegible]	
1900	3			[illegible]		[illegible]	[illegible]	
3702	3			[illegible]		[illegible]	[illegible]	
4500	3			nat-t-ike		no-response	open|filtered	

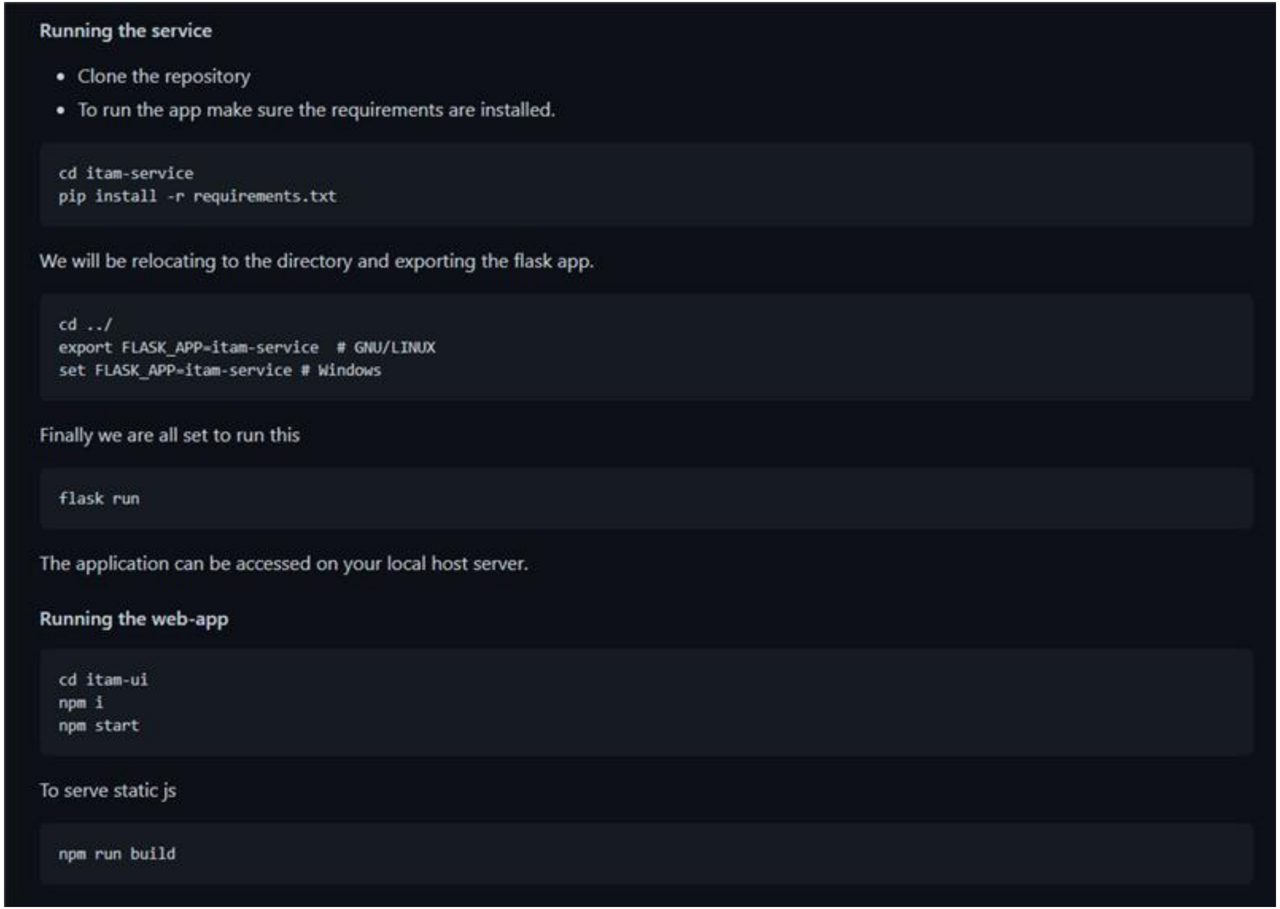
Individual chart preview



Port Number	Configuration	CPE	Extra Information	Name	Product	Reason	State	Version
137	3			netbios-ns		no-response	open|filtered	
1900	3			upnp		no-response	open|filtered	
3702	3			ws-discovery		no-response	open|filtered	
4500	3			nat-t-ike		no-response	open|filtered	
5050	3			[illegible]		no-response	open|filtered	
5353	3			zeroconf		no-response	open|filtered	
5355	3			llmnr		no-response	open|filtered	
137	3			netbios-ns		no-response	open|filtered	
1900	3			upnp		no-response	open|filtered	

Sample data grid

Starting the Application



Running the service

- Clone the repository
- To run the app make sure the requirements are installed.

```
cd itam-service
pip install -r requirements.txt
```

We will be relocating to the directory and exporting the flask app.

```
cd ../
export FLASK_APP=itam-service  # GNU/LINUX
set FLASK_APP=itam-service # Windows
```

Finally we are all set to run this

```
flask run
```

The application can be accessed on your local host server.

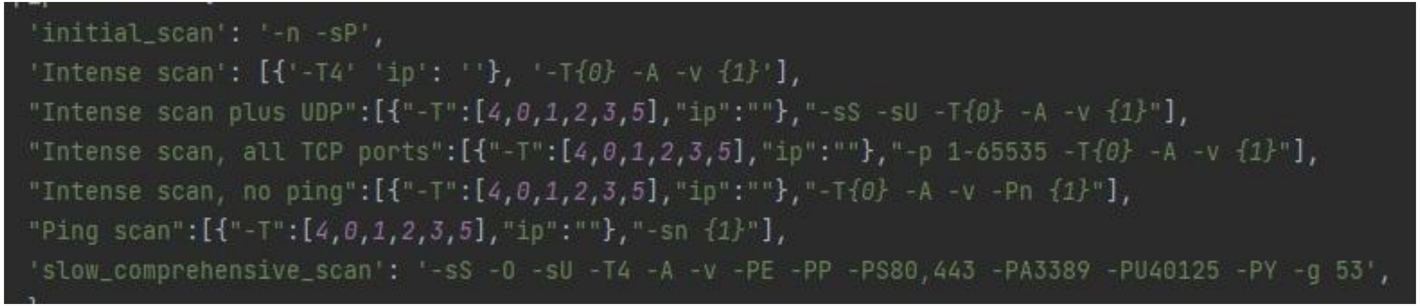
Running the web-app

```
cd itam-ui
npm i
npm start
```

To serve static js

```
npm run build
```

Scanning Commands



```
'initial_scan': '-n -sP',
'Intense scan': [{'-T4' 'ip': ''}, '-T{0} -A -v {1}'],
"Intense scan plus UDP":[{"-T":[4,0,1,2,3,5],"ip":""},"-sS -sU -T{0} -A -v {1}"],
"Intense scan, all TCP ports":[{"-T":[4,0,1,2,3,5],"ip":""},"-p 1-65535 -T{0} -A -v {1}"],
"Intense scan, no ping":[{"-T":[4,0,1,2,3,5],"ip":""},"-T{0} -A -v -Pn {1}"],
"Ping scan":[{"-T":[4,0,1,2,3,5],"ip":""},"-sn {1}"],
'slow_comprehensive_scan': '-sS -O -sU -T4 -A -v -PE -PP -PS80,443 -PA3389 -PU40125 -PY -g 53',
```

-O/A enable os detection
-sS TCP SYN scan
-sU UDP scan
-sn ping scan - disable port scan
-T<0-5> timing template
-v Increases verbosity level
-PE/PY/PU/PA ICMP echo, timestamp, and netmask request discovery servers
-PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
-Pn: Treat all hosts as online -- skip host discovery
-g to specify source port

Limitations

- The current scanning methodology introduces a performance overhead as the scan itself takes an hour or two based on the size of the network.

- The increase in data size might cause inefficiency in the storage of data in the databases as the data needs to be parsed into a required format for further processing.

- Being a proof of concept to the proposed solution, the application does not provide multiple access levels for the users under the same organization i.e each organization is allowed to have only a single account.

Future Scope

➢ Extending:

- Support for compliance management for software in the network.
- Current charts to a more detailed asset analysis which might include the license information and registry scan etc.
- Functionalities to provide API endpoints to access asset details for other application use cases.

➢ Adding functionality:

- To trigger workflows from the UI directly to resolve issues.
- To maintain audit history for the edits done on the UI.

➢ Integration of Shodan API on the service side instead of the UI to provide more powerful endpoints.

➢ Moving service implementation to a much stable framework such as Spring Boot etc.