**Auto locate servers in a network**

**Table of Content**

1. **Background**

**1.1. Purpose**

To develop a management software/application that can locate the server’s identity and display Server Vendor, Server Model, Server Serial Number IP in the web GUI.

**1.2. Scope**

1.2.1 System Requirements:

* CentOS operating system running on all servers.
* All servers connected on the same network as the central DHCP server.
* The central server and client machines need to be configured suitably.

1.2.2 Objectives:

* Develop a software/application that can locate the identity of servers on a network.
* Display the vendor information, model, serial number, and IP address of the servers in a user-friendly GUI.
* Implement the DORA process of DHCP to collect client-specific information from the server machines. Log the collected information in the DHCP server's log files.
* Build a Python application to query the DHCP log files for information about network devices. Develop a GUI frontend to display information about the servers on the network.
* Enhance network management and monitoring. Improve network security by enabling administrators to monitor and control the devices connected to the network.
* Test and validate the application to ensure it is reliable, efficient, and secure.

**1.3. References**

* <https://linux.die.net/man/5/dhclient.conf>
* <https://linux.die.net/man/5/dhcp-options>
* <https://flask.palletsprojects.com/en/2.0.x/>
* <https://datatracker.ietf.org/doc/html/rfc2131>
* <https://manpages.ubuntu.com/manpages/bionic/man8/dmidecode.8.html>

2. **Functional Specification**

**2.1. Summary**

This project aims to develop a comprehensive solution for efficient management of servers in a network by implementing the DORA process of Dynamic Host Configuration Protocol (DHCP). The project utilizes a central DHCP server to assign IP addresses to devices joining the network. When a new device enters the network, it sends a lease request to the DHCP server, which assigns it an IP address and receives information about the client device, including its vendor details in the discover packet sent by the client device.

To collect this information, the client device is configured to run the commands at the time of boot-up where a script file runs and appends the client-specific information to the dhclient.conf file and is logged by the DHCP server in the DHCP log files.

To make this information readily available to network administrators, a Python, Go, Java application is buil. This application queries the DHCP log files for information about the network devices and displays it in a user-friendly graphical user interface (GUI). The GUI provides details such as server vendor, model, serial number, enabling network administrators to easily manage and monitor the devices connected to the network.

**2.2. Requirements Details**

| **ID** | **Description** |
| --- | --- |
| REQ\_1 | CentOS server when starts booting should have DHCP enabled. |
| REQ\_2 | Install and run more than 4-5 centos Server. Receive DHCP request on 6th VM (can be widows/Linux which has a proper IP address. |
| REQ\_3 | Each CentOS server should have VendorInfo, Model-Info and SerialNumber info. This can be part of any internal files. Note Serial Number is a unique number [example phone number is unique] |
| REQ\_4 | DHCP request (broadcast) should carry VendorInfo, Model-Info and SerialNumber. |
| REQ\_5 | Write JAVA Application to read the DHCP Request. And display the each centos servers VendorInfo, Model-Info and SerialNumber in a table format. Let this application acts like HTTP application. |
| REQ\_6 | Write Python Application to read the DHCP Request. And display the each centos servers VendorInfo, Model-Info and SerialNumber in a table format. Let this application acts like HTTP application. |
| REQ\_7 | Write C++ Application to read the DHCP Request. And display the each centos servers VendorInfo, Model-Info and SerialNumber in a table format. Let this application acts like HTTP application. |

**2.3. Assumptions**

The systems need to be configured such that every new system joining the network sends a new lease request with specific information to the central server.

**2.4. Caveats**

Systems should be configured to have required files for enabling the DORA communication process.

**2.5. Platforms Supported**

CentOS/ Linux systems.

**2.6. Not in Scope**

**2.6. Update on the requirements met:**

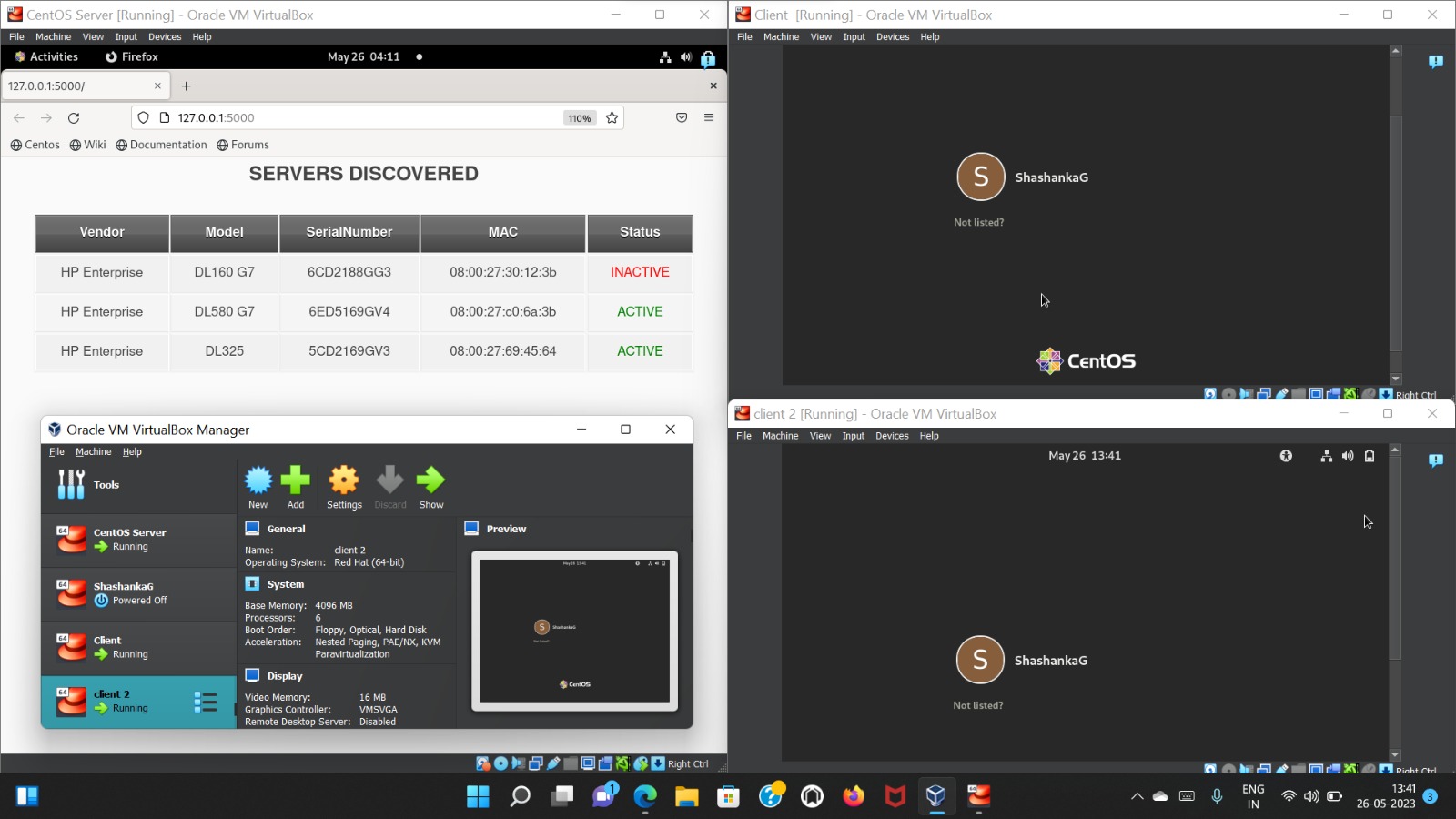
The DORA process of DHCP is used to collect client-specific information from the server machines. Every time a new device enters the network, it sends a lease request to the central DHCP server for the assignment of an IP address. The DHCP server responds with an IP address and other configuration information in the form of a lease offer.

The vendor information from the client is sent to the server machine through dhcp discover packet. The client system-specific information is copied from a file(details.txt) , which gets appended in the dhclient.conf file. This information is then logged in the DHCP server's log files.

3. **Customer Experience**

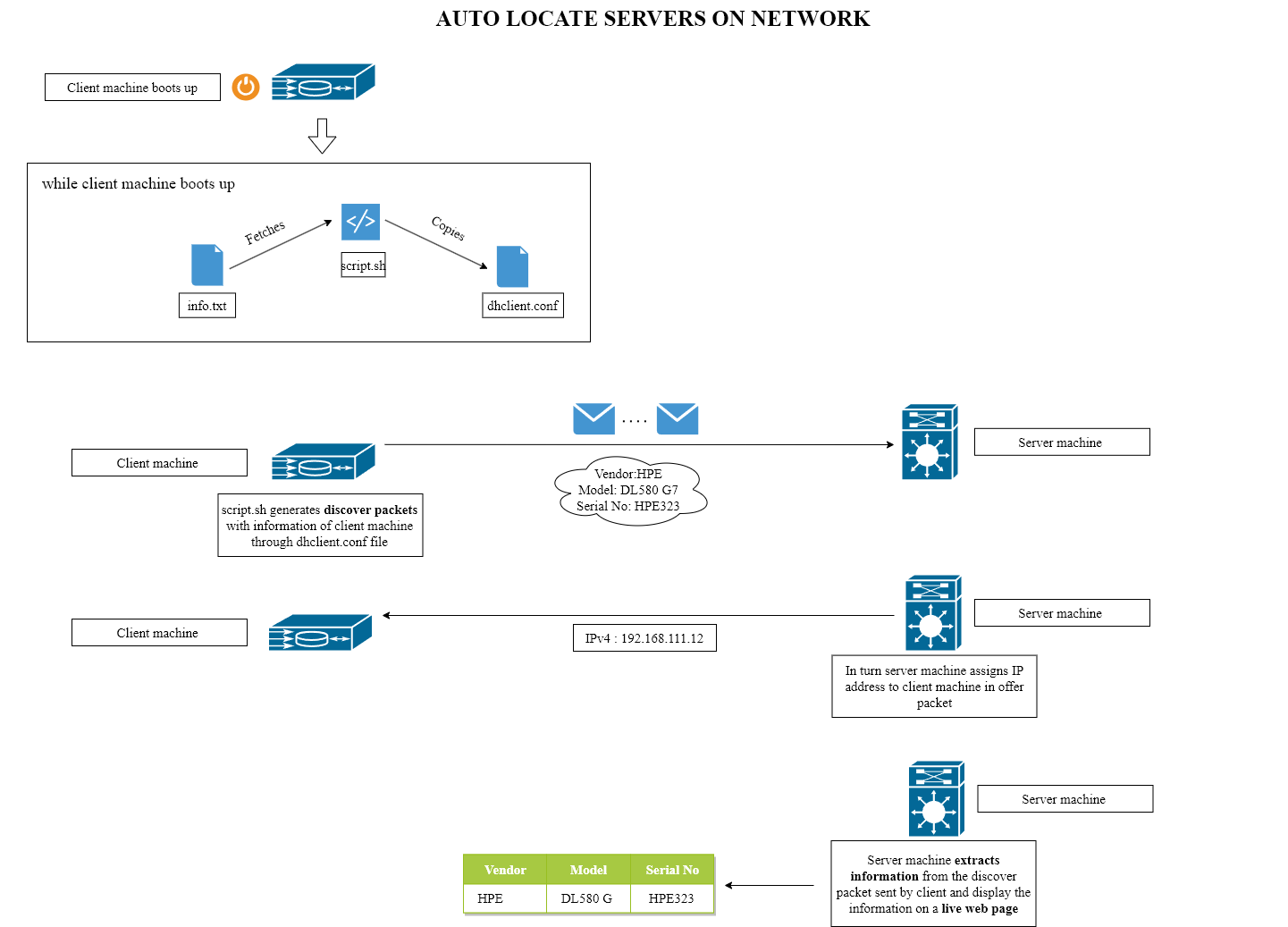
The DHCP central server is responsible for assignment of IP address to client machine and receiving client information through the DORA process. The user is provided with a friendly GUI to view the information from all the machines on the network, which shows the active devices in the network.

**3.1. Customer Visible Behavior Changes**

4. **External Dependencies**

| **ID** | **Description** | **Category** |
| --- | --- | --- |
| 1 | Scapy | Python module  for tracing the packet |

5. **Architecture**

6. **Performance**

Yes the application can scale for thousands of servers provided all the network and system requirements are taken into consideration.

7. **Security**

8. **Feature Risks**

NA

9. **Quality**

We have tested the server with 3 client machines booting at once and successfully fetching the client-specific details on the server machine’s web interface.

10. **Appendix**

**10.1. Failure Modes**

Failure or downtime of DHCP server could affect the functioning of the application.

.

**Acronyms Table**

| **Abbreviation** | **definition** |
| --- | --- |
| DHCP | Dynamic host configuration protocol |
| HTTP | Hyper text Transfer protocol |
| DORA | Discover, Offer, Request, Acknowledgement |