

A
Mini Project
On
OSFINDER
(Submitted in the Partial Fulfilment of
the Requirements for the Award of
the Degree)

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by
BANDARU SWAPNIL (197R1A05J7)
Under the Guidance of
B. LAXMAIAH
(Associate Professor)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CMR TECHNICAL CAMPUS
UGC AUTONOMOUS
(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by
AICTE, NewDelhi) Recognized Under Section 2(f) & 12(B) of the UGC Act. 1956,
Kandlakoya (V), Medchal Road, Hyderabad-501401.
2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project entitled “**OSFINDER**” being submitted by **BANDARU SWAPNIL(197R1A05J7)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. B. Laxmaiah
(Associate Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Dr.B.LAXMAIAH**, Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. Shilpa, Dr.M . Subha Mastan Rao & J. Narasimharao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju**, Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

BANDARU SWAPNIL (197R1A05J7)

ABSTRACT

Network consists of many devices or systems which are connected to the network. When many devices are connected to the network, each device may have its different own operating system, IP Address, Mac Address, unique device name. In an organization, if there are huge number of systems connected to the network, this script makes the process of keeping the information of all the systems into an CSV format. I developed a script which automates the result of Nmap and filters the results and displays the result in CSV Format. This script helps in searching for a particular operating system in the network and also all the devices information in the network.

Keywords: Nmap, IP Address, Mac Address, CSV Forma

LIST OF FIGURES

Figure 4.2: USE CASE DIAGRAM FOR OSFINDER	9
Figure 4.3: ACTIVITY DIAGRAM FOR OSFINDER	10
Figure 4.4: SEQUENCE DIAGRAM FOR OSFINDER	11
Figure 4.5: CLASS DIAGRAM FOR OSFINDER	12

LIST OF TABLES

Table 1 : TEST CASES

25

LIST OF SCREENSHOTS

Screenshot 6.1: COMMAND PROMPT	16
Screenshot 6.2: CHANGING DIRECTORY OF CMD	17
Screenshot 6.3: COMPILING THE SCRIPT	18
Screenshot 6.4: EXECUTING THE OSFINDER SCRIPT	19
Screenshot 6.5: INPUTTING IP ADDRESS RANGE	20
Screenshot 6.6: SCRIPT EXECUTING	21
Screenshot 6.7: RESULTS IN CSV FORMAT	22

TABLE OF CONTENTS

ABSTRACT	i
LIST OF FIGURES	ii
LIST OF TABLES	iii
LIST OF SCREENSHOTS	iv
1. INTRODUCTION.....	1
1.1 PROJECT SCOPE.....	1
1.2 PROJECT PURPOSE.....	1
1.3 PROJECT FEATURES	1
2. LITERATURE SURVEY	2
2.1 INTRODUCTION.....	2
2.2 EXISTING SYSTEM.....	2
2.2.1 NMAP (Network Mapper).....	2
2.3 PROPOSED SYSTEM.....	3
2.3.1 ADVANTAGES OF PROPOSED SYSTEM.....	3
3. SYSTEM ANALYSIS	4
3.1 INTRODUCTION.....	4
3.2 FEASIBILITY STUDY	4
3.2.1 ECONOMIC FEASIBILITY	5
3.2.2 TECHNICAL FEASIBILITY	5
3.2.3 BEHAVIORAL FEASIBILITY.....	5
3.3 HARDWARE AND SOFTWARE REQUIREMENTS	6
3.3.1 HARDWARE REQUIREMENTS	6
3.3.2 SOFTWARE REQUIREMENTS.....	6
4. ARCHITECTURE	7
4.1 PROJECT ARCHITECTURE	7
4.2 DESCRIPTION.....	8
4.3 USE CASE DIAGRAM	9
4.4 ACTIVITY DIAGRAM	10
4.5 SEQUENCE DIAGRAM	11
4.6 CLASS DIAGRAM.....	12
5. IMPLEMENTATION.....	13
5.1 SAMPLE CODE	13
6.RESULTS.....	16

6.1 COMMAND PROMPT	16
6.2 CHANGING DIRECTORY OF CMD	17
6.3 COMPILING THE SCRIPT	18
6.5 INPUTTING IP ADDRESS RANGE	20
6.6 SCRIPT EXECUTING.....	21
6.7 RESULTS IN CSV FORMAT	22
7. TESTING	23
7.1 INTRODUCTION TO TESTING	23
7.2 TYPES OF TESTING.....	23
7.2.1 UNIT TESTING.....	23
7.2.2 INTEGRATION TESTING	24
7.2.3 FUNCTIONAL TESTING.....	24
7.3 TEST CASES	25
7.3.1 CLASSIFICATION	25
8. CONCLUSION	26
8.1 CONCLUSION	26
8.2 FUTURE RECOMMENDATIONS	26
9. BIBLIOGRAPHY	27
9.1 REFERENCES.....	27
9.2 GITHUB LINK	27

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The goal of the project is to filter the results of NMAP scan. Nmap scans the network and fetches all the information of the systems that are connected to the network. The project mainly focuses on filtering the scanned information of the Nmap and filtering the important information based on the operating system and also it formatting it into an order into an CSV Format. In this project we will be using the script which automates the process of fetching the information and filters the information and saves into an CSV format.

1.2 PROJECT PURPOSE

The project has been developed to identify the details of the devices connected to the same network. Network consists of many devices that are connected, in which each device has its own unique IP Address, MAC Address, Device Name. This Script also automates the network scanning and formats the results in CSV Format.

1.3 PROJECT FEATURES

This project has many features of this project are that the script that is developed that detects all of the information like IP Address, MAC Address, Device Name. This script allows the feature of scanning the devices in the network based on the user input of IP Address range and scans the network using NMAP tool. This script also fetches the information of each device and formats the result into CSV Format. This script also retrieves the hostname of each system connected in the network. This script take the input as the range of IP Address and scans the network at a time.

2. LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 INTRODUCTION

In this chapter, we will give an overview of NMAP tool which is a network mapper tool used to fetch the details of the systems that are connected to the network. Nmap is used almost by every organization which helps in monitoring the changes of the systems in the network and also it fetches various other details like vulnerability status, firewall status and various other details.

2.2 EXISTING SYSTEM

2.2.1 NMAP (Network Mapper)

Nmap, short for Network Mapper, is a free and open source tool used for vulnerability checking, port scanning and, of course, network mapping. Despite being created back in 1997, Nmap remains the gold standard against which all other similar tools, either commercial or open source, are judged.

The heart of Nmap is port scanning. How it works is that users designate a list of targets on a network that they want to learn information about. Users don't need to identify specific targets, which is good because most administrators don't have a complete picture of everything that is using the potentially thousands of ports on their network. Instead, they compile a range of ports to scan. It's also possible to scan all network ports, although that would potentially take a lot of time and eat up quite a bit of available bandwidth. Plus, depending on the type of passive defenses that are in use on the network, such a massive port scan would likely trigger security alerts.

2.2.2 DISADVANTAGES OF EXISTING SYSTEM

Nmap does not provide the feature to filter the information based on its operating system and it doesn't allow the scanned information to be save in the CSV Format (Excel file). Nmap doesn't allow to filter a particular system on the version of the operating system. The process of formatting all the necessary information of devices in the network is very difficult process as it needs each and every record of the scanned results in Nmap to manually copy all of it into excel sheet (CSV FILE).

2.3 PROPOSED SYSTEM

This project presents a script that automates the result of network scan done using NMAP tool. This script initiates the scan through the network based on user input of IP Addresses. This system also formats the scan results in CSV format and it only formats the important information of devices like IP Address, MAC Address, Device Name, Hostname, System name. It formats it into a excel sheet (CSV format) and the name of the excel file is automatically assigned with the user input IP Address. The script also allows the feature to filter the results based on the operating system provided at the time of execution.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

The advantages of the proposed system are:

- The script automates the result of scan and formats it into CSV
- The script saves the scanned result in an ordered format
- The script filters the result based on the operating system

3. SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 INTRODUCTION

Following the analysis of the task's requirements, the next stage is to study the problem and comprehend its context. The analysis of the old system is the initial activity in this phase, followed by an understanding of the new system's needs and domain. Both tasks are equally vital, but the first one acts as a foundation for providing functional specifications and subsequently designing the proposed system successfully. Knowing the characteristics and needs of a new system is more complex and demands creative thinking, and knowledge the properties and requirements of an existing operating system is equally challenging; incorrect understanding of the current system might lead to a solution diversion.

3.2 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis:

- o Economic Feasibility
- o Technical Feasibility
- o Social Feasibility

3.2.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication that the system is economically possible for development.

3.2.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.2.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

3.3 HARDWARE AND SOFTWARE REQUIREMENTS

For execution of this project, we require few hardware and software interfaces for implementing the script. This project script is developed using Java in a Windows environment. The project's major goal is to filter the information of the NMAP results. The different hardware and software components are required to execute this script.

3.3.1 HARDWARE REQUIREMENTS

Hardware Requirements that are required to execute osfinder script are:

System Architecture	: 64-bit X86 with Windows or Linux
CPU	: Intel core 2 CPU Q6600 @2.40GHz or greater
RAM	: 4GB or greater

3.3.2 SOFTWARE REQUIREMENTS

Software Requirements that are required to execute the osfinder script are:

Operating System: Windows 7 or greater

Java Standard Edition (SE) Development Kit (JDK): Version 7 or greater

Nmap (Network Mapper): Version 3.70 or greater

4. ARCHITECTURE

4. ARCHITECTURE

4.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

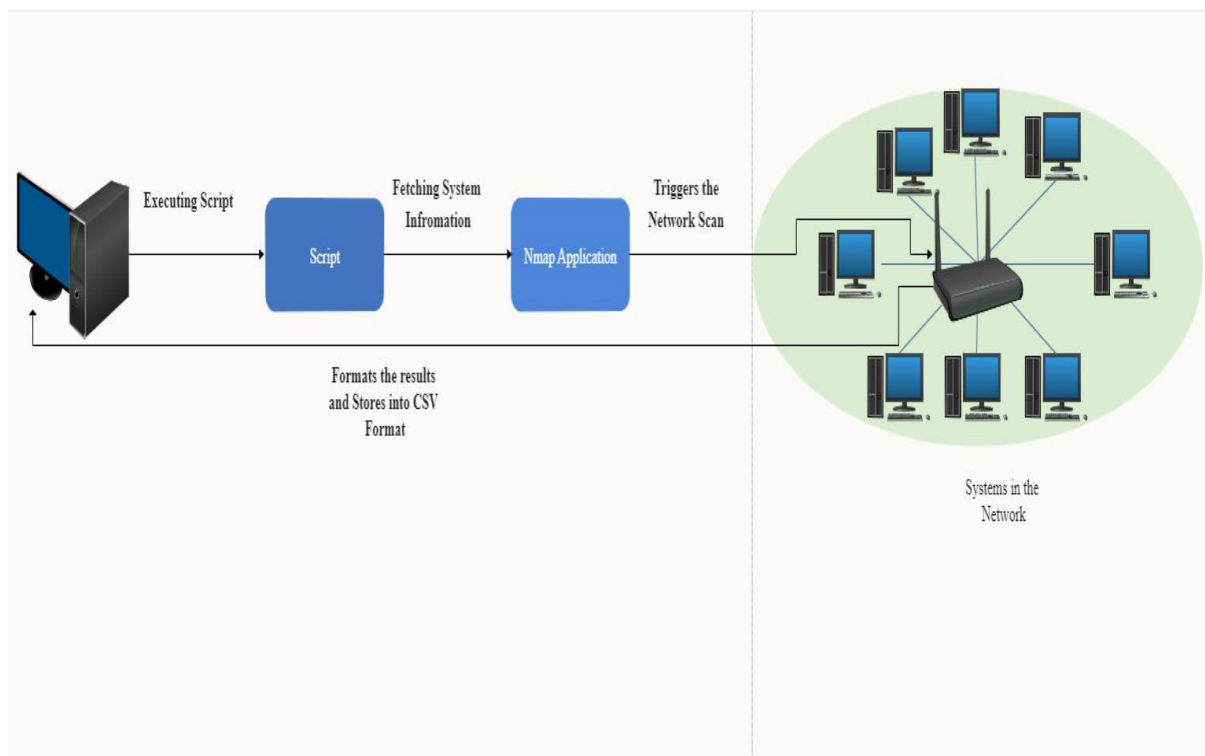


Figure 4.1: ARCHITECTURE DIAGRAM FOR OSFINDER

4.2 DESCRIPTION

The script mainly works on the NMAP Application. The script uses the java **process builder class** to run the NMAP application. The process builder class is used to for run operating system processes.

The script uses the NMAP command which is “ **nmap -p 445 --script smb-os-discovery ###.###.###.0/24** ”. This command is used to execute the default script which is present in **NMAP**. The **smb-os-discovery script** attempts to determine the operating system, computer name, domain, workgroup, and current time over the **SMB protocol (ports 445 or 139)**.

This is done by starting a session with the anonymous account (or with a proper user account, if one is given; it likely doesn't make a difference); in response to a session starting, the server will send back all this information.

Retrieving the name and operating system of a server is a vital step in targeting an attack against it, and this script makes that retrieval easy. Additionally, if a penetration tester is choosing between multiple targets, the time can help identify servers that are being poorly maintained

After the Nmap scan is completed using the above command, then the results which are obtained after the scan are unordered. Only the important details of the system like **IP Address, MAC address, Hostname, Operating System name** etc. are formatted and parsed into a CSV Format.

4.3 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model. A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures

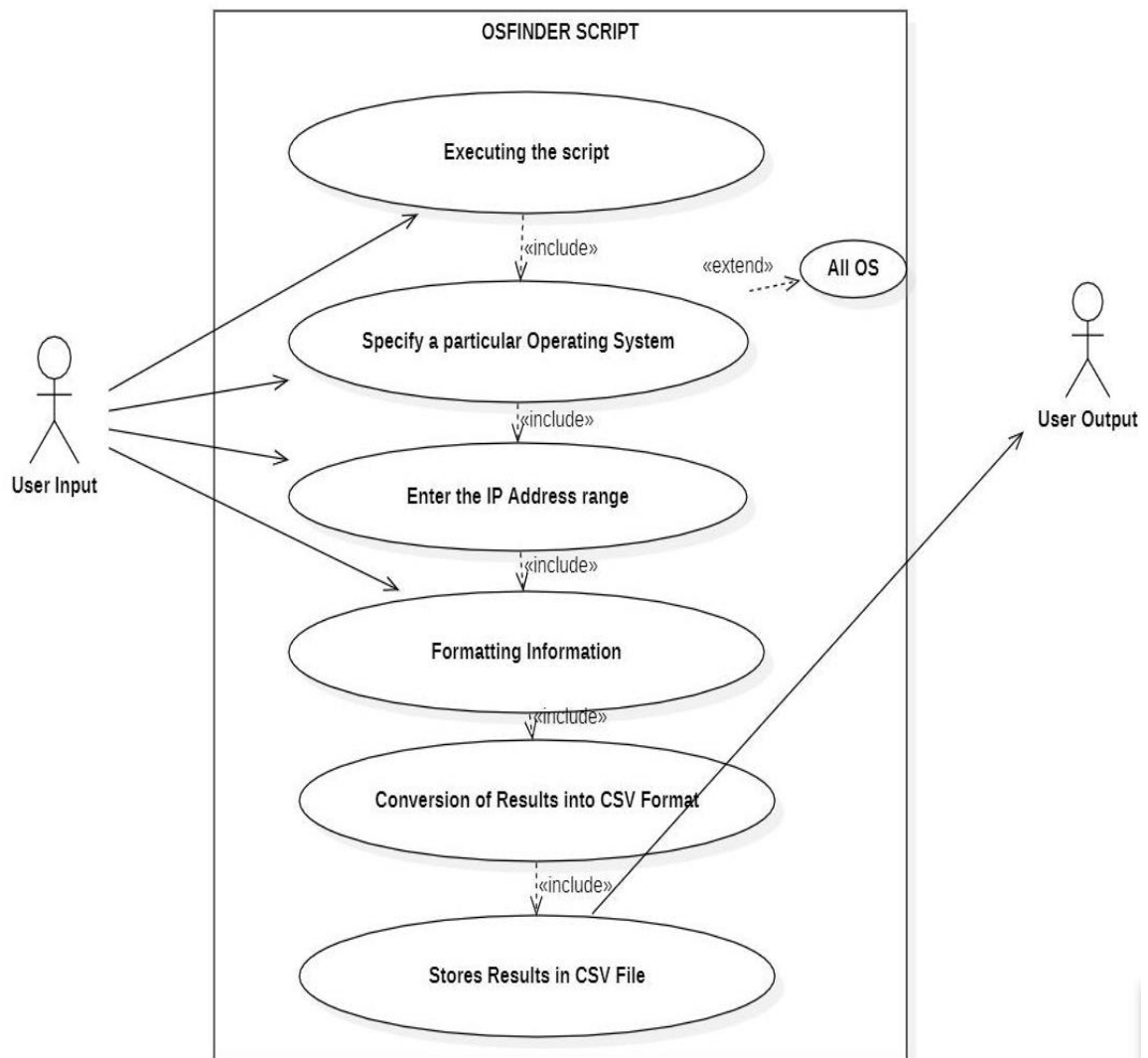


Figure 4.1: USE CASE DIAGRAM FOR OSFINDER

4.4 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control..

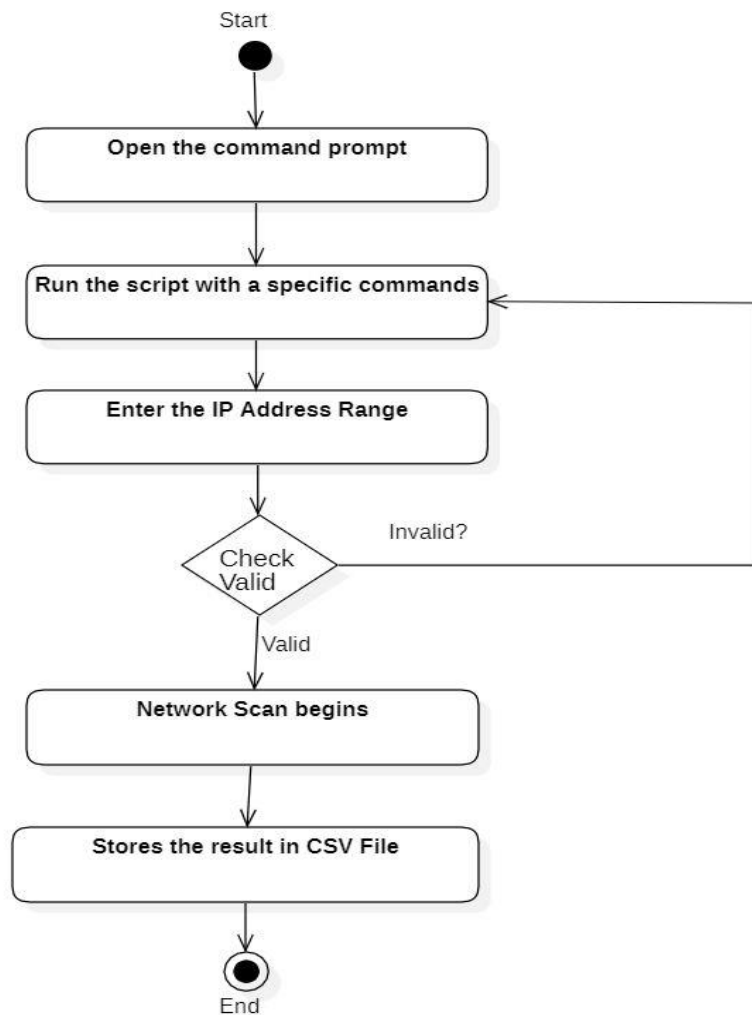


Figure 4.2: ACTIVITY DIAGRAM FOR OSFINDER

4.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

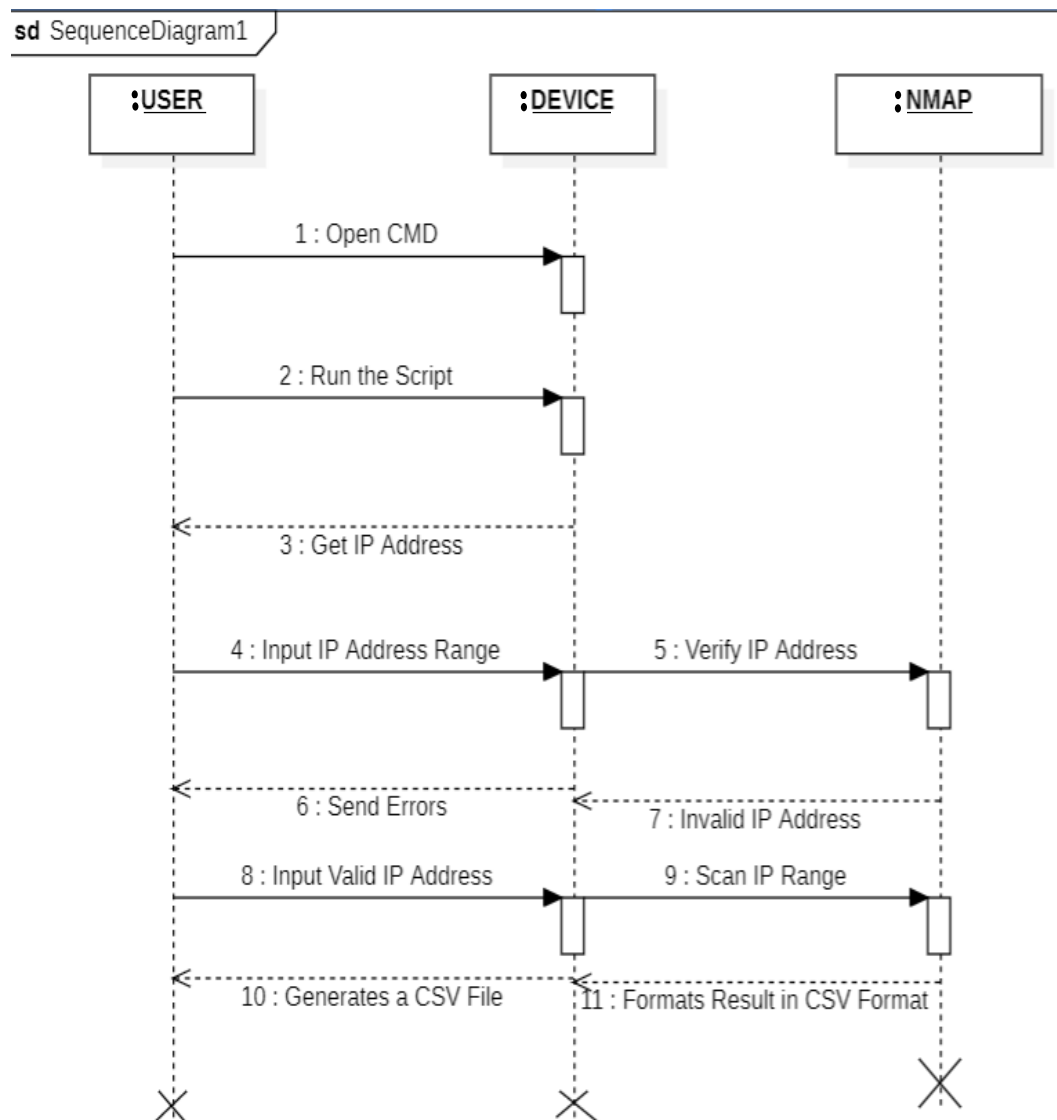


Figure 4.3: SEQUENCE DIAGRAM FOR OSFINDER

4.6 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages

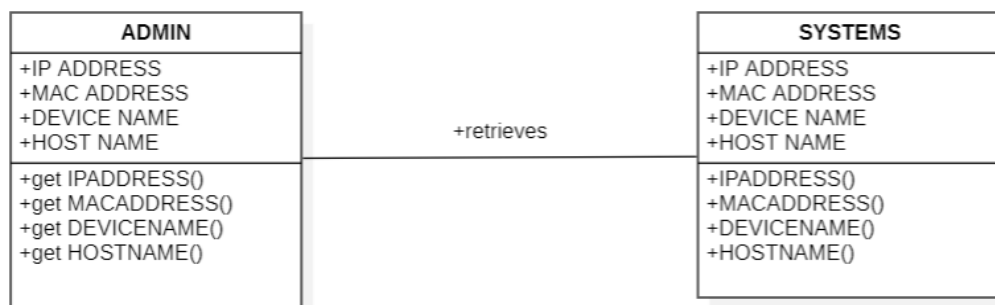


Figure 4.4: CLASS DIAGRAM FOR OSFINDER

5. IMPLEMENTATION

5. IMPLEMENTATION

5.1 SAMPLE CODE

```

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.*;

public class OSFinder {
    // ProcessBuilder is used to actually build and run applications (ipconfig and Nmap)
    ProcessBuilder pb = new ProcessBuilder();
    InputStream inputStream;
    InputStreamReader inputStreamReader;
    BufferedReader bufferedReader;
    // Indicates whether the program will output extra stuff
    // Operating System that is being searched for
    static String searchOS = "Windows XP";
    /**
     * The main method. Creates an instance of this class that it then uses to find the subnet
     * for later use in the Nmap OS detection command.
     *
     * @param args An array of strings
     */
    public static void main(String[] args) {

```

```

// Loop over the arguments
for (String arg : args)
{
    // Check if the user wants extra output
    if (arg.equals("-eo"))
    {
        extraOutput = true;
    }

    // Check if the user wants to look for Windows 7 computers instead
    else if (arg.equals("-os7"))
    {
        searchOS = "Windows 7";
    }
    else if (arg.equals("-os8")){
        searchOS= "Windows 8";
    }
    else if (arg.equals("-os10")){
        searchOS="Windows 10";
    }
    else if (arg.equals("-os11")){
        searchOS="Windows 11";
    }
    else if (arg.equals("-osunix")){

        searchOS="Unix";
    }
}

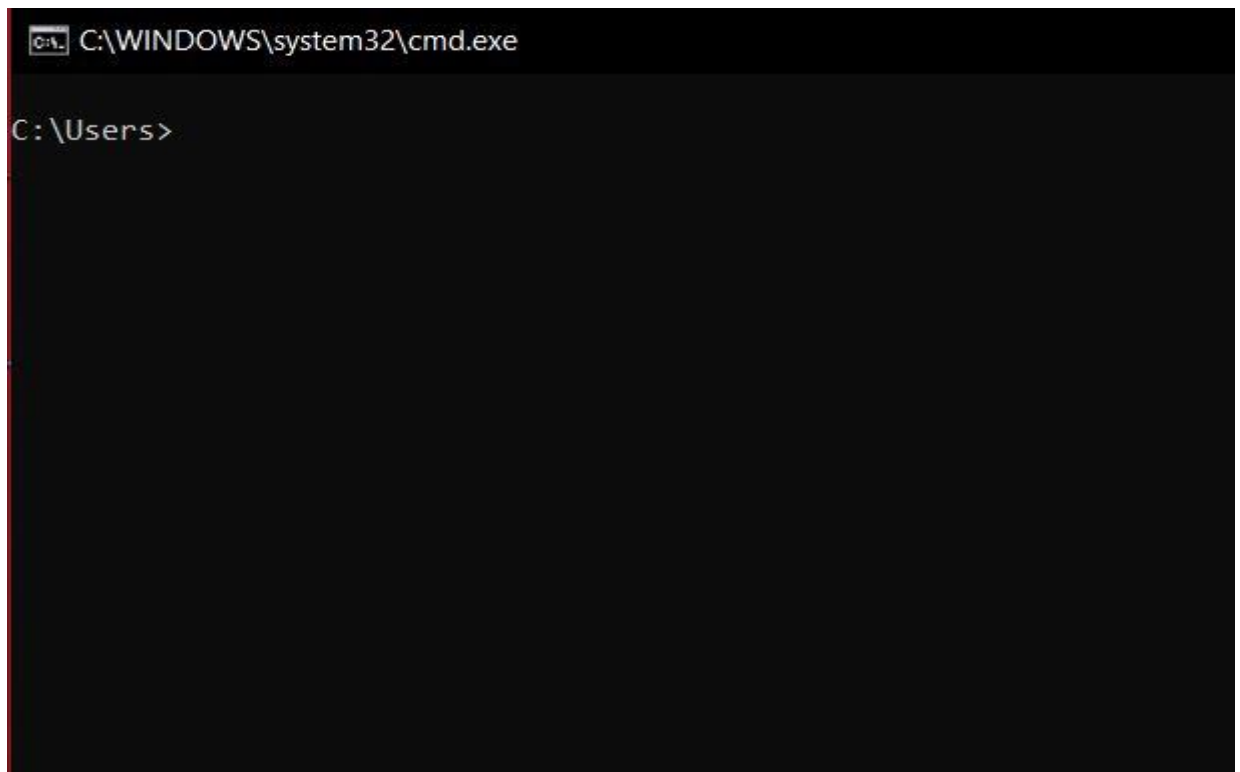
```

```
else if(arg.equals("-osall")){  
  
    searchOS="";  
}  
// Argument not found  
else  
{  
    printInvalidArgument();  
    System.exit(1);  
}}
```

6. RESULTS

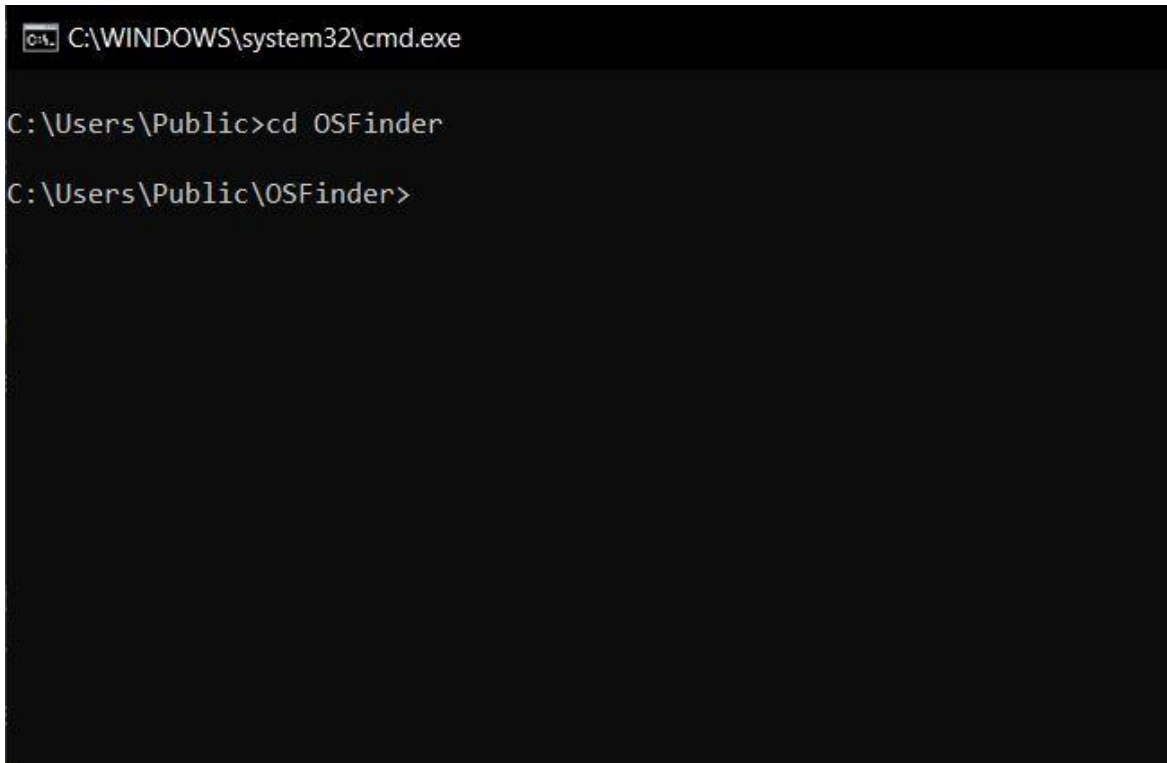
6.RESULTS

6.1 COMMAND PROMPT



Screenshot 6.1: COMMAND PROMPT

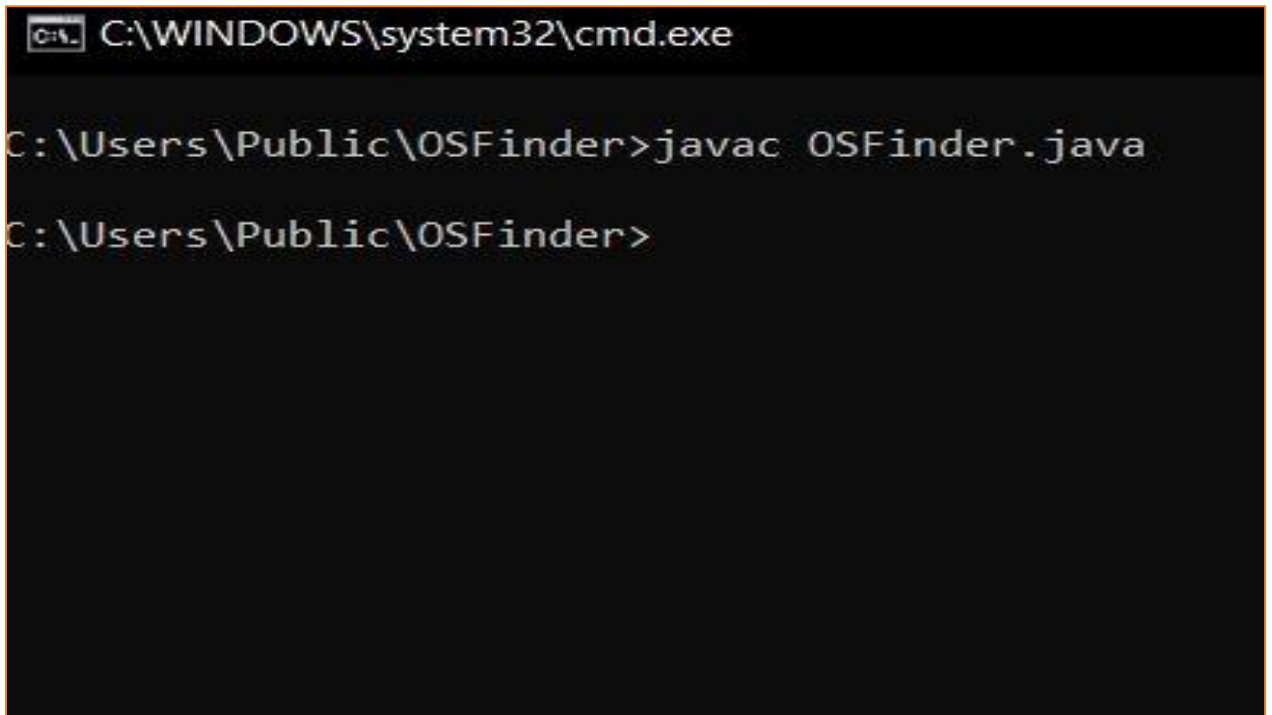
6.2 CHANGING DIRECTORY OF CMD

A screenshot of a Windows Command Prompt window. The title bar at the top shows the icon for Command Prompt and the text "C:\WINDOWS\system32\cmd.exe". The main area of the window is black with white text. The first line shows the prompt "C:\Users\Public>" followed by the command "cd OSFinder". The second line shows the prompt "C:\Users\Public\OSFinder>" indicating the directory has been successfully changed.

```
C:\WINDOWS\system32\cmd.exe  
  
C:\Users\Public>cd OSFinder  
C:\Users\Public\OSFinder>
```

Screenshot 6.2: CHANGING DIRECTORY OF CMD

6.3 COMPILING THE SCRIPT



```
C:\WINDOWS\system32\cmd.exe

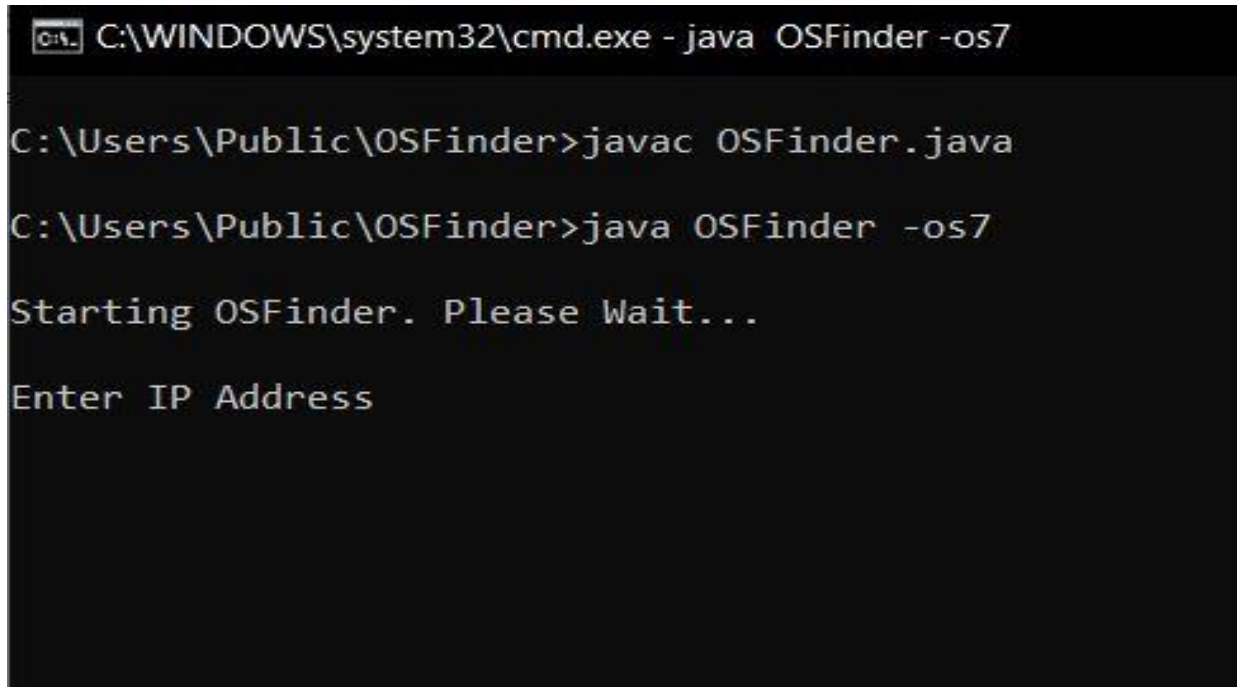
C:\Users\Public\OSFinder>javac OSFinder.java

C:\Users\Public\OSFinder>
```

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\WINDOWS\system32\cmd.exe". The command prompt shows the current directory as "C:\Users\Public\OSFinder". The user has entered the command "javac OSFinder.java" to compile the script. The prompt is now waiting for the next command.

Screenshot 6.3: COMPILING THE SCRIPT

6.4 EXECUTING THE OSFINDER SCRIPT



```
C:\WINDOWS\system32\cmd.exe - java OSFinder -os7

C:\Users\Public\OSFinder>javac OSFinder.java

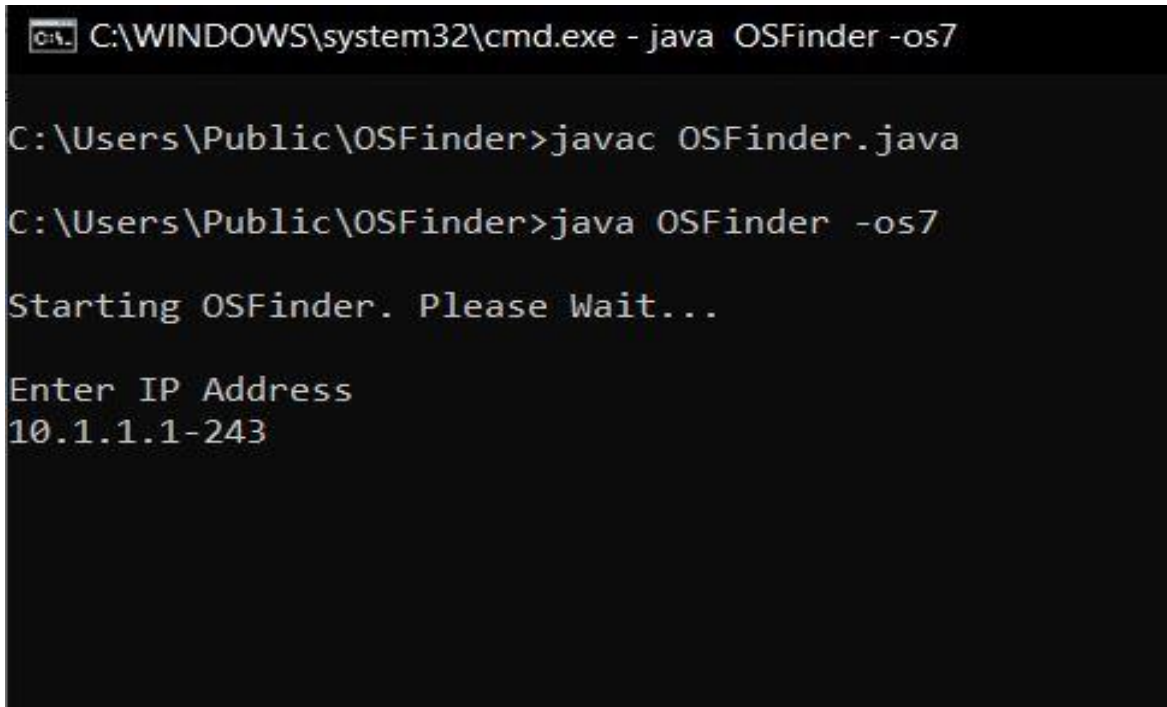
C:\Users\Public\OSFinder>java OSFinder -os7

Starting OSFinder. Please Wait...

Enter IP Address
```

Screenshot 6.4: EXECUTING THE OSFINDER SCRIPT

6.5 INPUTTING IP ADDRESS RANGE



```
C:\WINDOWS\system32\cmd.exe - java OSFinder -os7

C:\Users\Public\OSFinder>javac OSFinder.java

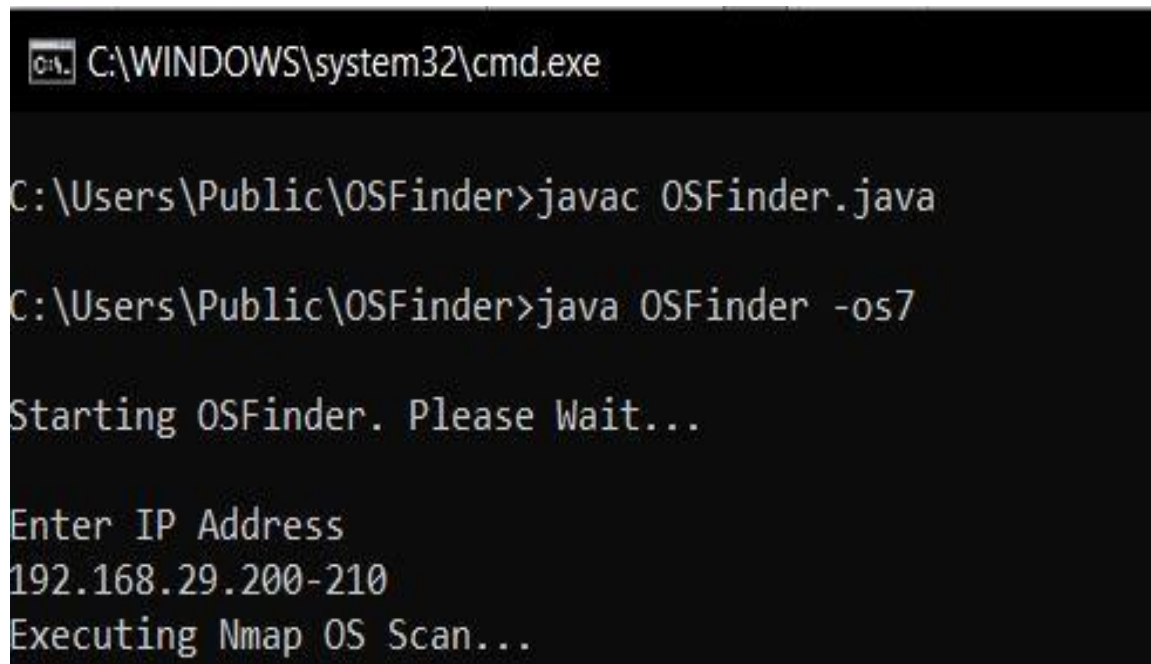
C:\Users\Public\OSFinder>java OSFinder -os7

Starting OSFinder. Please Wait...

Enter IP Address
10.1.1.1-243
```

Screenshot 6.5: INPUTTING IP ADDRESS RANGE

6.6 SCRIPT EXECUTING



```
C:\WINDOWS\system32\cmd.exe

C:\Users\Public\OSFinder>javac OSFinder.java

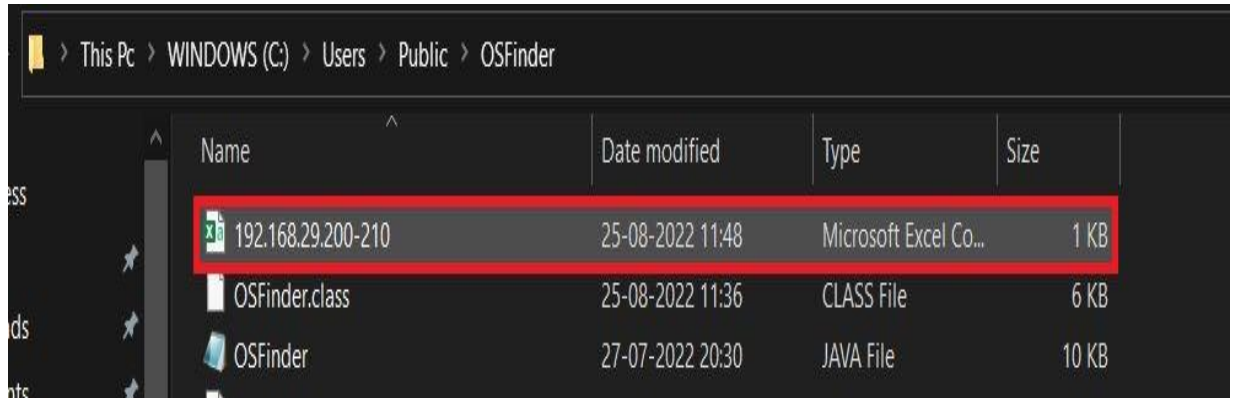
C:\Users\Public\OSFinder>java OSFinder -os7

Starting OSFinder. Please Wait...

Enter IP Address
192.168.29.200-210
Executing Nmap OS Scan...
```

Screenshot 6.6: SCRIPT EXECUTING

6.7 RESULTS IN CSV FORMAT



Name	Date modified	Type	Size
192.168.29.200-210	25-08-2022 11:48	Microsoft Excel Co...	1 KB
OSFinder.class	25-08-2022 11:36	CLASS File	6 KB
OSFinder	27-07-2022 20:30	JAVA File	10 KB

Screenshot 6.7: RESULTS IN CSV FORMAT

7. TESTING

7. TESTING

7.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.2 TYPES OF TESTING

7.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : Identified classes of valid input must be accepted.

InvalidInput : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test

7.3 TEST CASES

7.3.1 CLASSIFICATION

Table 1 : TEST CASES

Test case ID	Test case name	Purpose	Input	Output
1	OS DETECTION	To detect OS	The user gives the input in the range of IP Address.	An output is formatted into CSV File
2	OS DETECTION	To detect OS	The user gives the input in the range of IP Address.	An output is formatted into CSV File

8. CONCLUSION

8. CONCLUSION

8.1 CONCLUSION

The problem of fetching device's information from the network and filtering the fetched data is solved. The Script eases the process of fetching the detailed information like IP Address, Mac Address, System Name, Hostname, Operating system of all the devices present in the network and it provides the filtered and ordered data. The script formats the data into the CSV format, which makes easier for any network administrator to keep track of all the devices connected to the network.

This Project makes easy for any network engineer to keep track of all the information of the systems in an ordered format and also makes easier for one to save information in an excel sheet.

8.2 FUTURE RECOMMENDATIONS

In future many other features can be added to the script for fetching other information like vulnerability status, Antivirus status of each system in the network.

The future work is to improve the other features and also the fetch the information of the systems even with the Firewall on status.

9. BIBLIOGRAPHY

9. BIBLIOGRAPHY

9.1 REFERENCES

[1] Nmap Scripts:

<https://nmap.org/nsedoc/scripts/smb-os-discovery.html>

[2] Nmap Version Scanning:

https://www.youtube.com/watch?v=AKWUllE9MOg&t=59s&ab_channel=HackerSploit

[3] Nmap Tutorial :

https://www.youtube.com/watch?v=4t4kBkMsDbQ&t=924s&ab_channel=NetworkChuck

9.2 GITHUB LINK

<https://github.com/SwapnilBandaru/OSFINDER>