**Detailed Project Report**

**Of**

**Vulnerb.**

**By**

**Abhijeet Adani**

**Table of Contents Page No**

Abstract

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 5

2. General Description 6

## 2.1 Product Perspective 6

3. Specific Requirements 7

3.1 External Interface Requirements 7

3.1.1 User Interfaces 7

### **3.1.2 Hardware Interfaces 8**

### **3.1.3 Software Interfaces 8**

3.2 Functional Requirements 9

1

4. Analysis Models 10

4.1 Data Flow Diagrams (DFD) 12

4.2State Transition Diagrams (STD) 13

5. Literature survey 14

References

**ABSTRACT**

Rising technologies overall the world is increasing day by day. The Internet is a massive interconnection that connects the global wide area network throughout the world. The network connection is used for various activities such as emails, downloading files, etc., and also new techniques propages through different filed in different activities.

With the increasing concern for security in the network, many approaches are laid out that try to protect the network from unauthorized access. New methods have been adopted in order to find the potential discrepancies that may damage the network. Most commonly used approach is the vulnerability assessment. By vulnerability, we mean, the potential flaws in the system that make it prone to the attack. Assessment of these system vulnerabilities provide a means to identify and develop new strategies so as to protect the system from the risk of being damaged.

Keywords:- Vulnerability scanner , web application penetration testing, WAP , OWASP Top 10, Port scanner, Hacking web Application, DevSecOps.

**1 Introduction**

**1.1 Introduction**

Vulnerb (Vulnerability/Port Scanner) is a free utility for network exploration and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts.

**1.1** **Purpose**

                  Port scanning is a method of determining which ports on a network are open and could be receiving or sending data.

   It is also a process for sending packets to specific ports on a host and analysing responses to

   identify vulnerabilities.

This scanning can’t take place without first identifying a list of active hosts and mapping those hosts to their IP addresses. This activity, called host discovery, starts by doing a network scan.

The goal behind port and network scanning is to identify the organization of IP addresses, hosts, and ports to properly determine open or vulnerable server locations and diagnose security levels. Both network and port scanning can reveal the presence of security measures in place such as a firewall between the server and the user’s device.

After a thorough network scan is complete and a list of active hosts is compiled, port scanning can take place to identify open ports on a network that may enable unauthorized access.

It’s important to note that network and port scanning can be used by both IT administrators and cybercriminals to verify or check the security policies of a network and identify vulnerabilities — and in the attackers’ case, to exploit any potential weak entry points. In fact, the host discovery element in network scanning is often the first step used by attackers before they execute an attack.

As both scans continue to be used as key tools for attackers, the results of network and port scanning can provide important indications of network security levels for IT administrators trying to keep networks safe from attacks.

**1.2 Scope**

      Port scanning is a popular reconnaissance technique which is used to discover the open ports and services available on a particular host. It can be used by the network administrator to check the open ports; it can be used by penetration tester during the security audit to check for vulnerabilities or it can be used by an attacker or a hacker to discover vulnerable service that they can exploit to break into the system. Every host connected to the LAN or internet run many services that listen on some port. By running the port scan, we can get the information like what all ports are open, what service is running on each port, what is the OS and MAC address of the target host, etc. We can configure the port scanner according to our requirement to get the maximum information from the target system.

Port scanners send a request to connect to each port sequentially and based on the response it decides whether the port is open, closed or filtered.

* Open port: The remote host sends a response to accept the connection.
* Closed port: The remote host sends a response indicating the connection is denied.
* Filtered port: There is no reply from the remote host.
* There are total 65536 ports each for TCP and UDP protocol which are divided into three ranges:
* Well known ports: These ports are in the range of 0-1023.
* Registered ports: These ports are associated with certain protocols or application. These ports lie in the range of 1024-49151.
* Dynamic /private ports: Ports>49151

**1.3 Definitions, Acronyms, and Abbreviations**

## Database:

## A database is an organized collection of [data](https://en.wikipedia.org/wiki/Data_(computing)). It is the collection of schemes, tables, queries, reports, views and other objects. The data is typically organized to model aspects of reality in a way that supports [processes](https://en.wikipedia.org/wiki/Process_(computing)) requiring information, such as modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

## Python: -

Python is a general-purpose, interpreted, high-level programming language popularly used for website development, data analytics and automation.

Python is a general-purpose language which means it is versatile and can be used to program many different types of functions. Because it is an interpreted language, it precludes the need for [compiling](https://www.techopedia.com/definition/540/compile) code before execution and because it is a high-level programming language, Python is able to abstract details from code. In fact, Python focuses so much attention on abstraction that its code can be understood by most novice programmers.

Python code tends to be short and when compared to compiled languages like C and C++, it executes programs slower. Its user-friendliness makes it a popular language for [citizen developers](https://www.techopedia.com/definition/30968/citizen-developer) working with [machine learning](https://www.techopedia.com/definition/8181/machine-learning-ml) algorithms in low-code no-code ([LCNC](https://www.techopedia.com/definition/33512/low-codeno-code-development-lcnc-development)) software applications.

Python has a simply [syntax](https://www.techopedia.com/definition/3959/syntax) and is known for having a large community that actively contributes to a growing selection of software modules and libraries. Python’s initial development was spearheaded by Guido van Rossum in the late 1980s. Today, Python is managed by the [Python Software Foundation](https://www.python.org/psf/).

|  |  |
| --- | --- |
| 1.SRS | Software Requirement Specification. |
| 2.SQL | Structured Query Language. |

## 1.5 Overview

## Ports are simply a software abstraction, used to distinguish between communication channels. Similar to the way IP addresses are used to identify machines on networks, ports identify specific applications in use on a single machine. For example, your web browser will by default connect to TCP port 80 of machines in HTTP URLs. If you specify the secure HTTPS protocol instead, the browser will try port 443 by default.

## Vulnerb works with two protocols that use ports: TCP and UDP. A connection for each protocol is uniquely identified by four elements: source and destination IP addresses and corresponding source and destination ports. All of these elements are simply numbers placed in the headers of each packet sent between hosts. The protocol is an eight-bit field, which specifies what type of packet is contained in the IP data (payload) section. For example, TCP is protocol number six, and UDP is 17. IPv4 addresses have a length of 32-bits, while ports are 16-bits long. IPv6 addresses are 128-bits in length. Further IP, TCP, and UDP header layout details can be found in [the section called “TCP/IP Reference”](https://nmap.org/book/tcpip-ref.html).

## 2. General Description

## 2.1 Product Perspective

## Intruders routinely search the Internet for servers that allow connections to exploitable inbound services. These services are exploitable generally because they contain some weakness such as a [buffer overflow](https://www.sciencedirect.com/topics/computer-science/buffer-overflow) condition that can be tripped to gain privileged access. Once privileged access is obtained, the intruder can perform administrative tasks such as changing system files, installing [malware](https://www.sciencedirect.com/topics/computer-science/malware), and stealing [sensitive information](https://www.sciencedirect.com/topics/computer-science/sensitive-informations). All good system administrators understand the importance of *hardening* servers by disabling all exploitable and unnecessary services. The problem is that hardening is a complex process that is made more difficult in environments where the operating system is proprietary and less transparent. Amazingly, most software and server vendors still deliver their products in configurations that include most services being default enabled.

## The deliberate insertion of [open service ports](https://www.sciencedirect.com/topics/computer-science/open-service-port) on an Internet-facing server is the most straightforward of all deceptive computing practices. The deliberately open ports are connected to back-end honey pot functionality, which is connected to monitoring systems for the purpose of observation and analysis. The result is that servers would thus present adversaries of national infrastructure with three different views of open service ports: (1) valid open ports one might expect, such as HTTP, DNS, and SMTP; (2) open ports that are inadvertently left open and might correspond to exploitable software; and (3) open ports that are deliberately inserted and connected to bogus assets in a honey pot. As long as it is generally understood that deception could *potentially* be deployed, there could be some uncertainty on the part of the adversary about which open ports are deliberate and which are inadvertent

## Security managers who use port scanners as part of a normal program of enterprise network protection often cringe at this use of deception. What happens is that their scanners will find these open ports, which will result in the generation of reports that highlight the presumed vulnerabilities to managers, users, and auditors. Certainly, the output can be manually cropped to avoid such exposure, but this might not scale well to a large enterprise. Unfortunately, solutions are not easily identified that solve this incompatibility between the authorized use of port scanners and the deliberate use of open ports as traps. It represents yet another area for research and development in deceptive computing.

# 3. Specific Requirements

## 3.1 External Interface

### 3.1.1 User Interface

## For the UI the user should using python version 3.7+ version with Tkinter

## If not, user can install it via command pip install Tkinter.

## User Gives their target host IP address in input field and able to specify their Port number to scan and then perform scan

## After completion of scan user can if he want to take it for future reference he can save the report in his machine.

### 3.1.2 Hardware Interfaces

* Intel Pentium 4 ( 2.8 GHz ) Processor and Above
* RAM 1 GB and Above
* System Type 32-bit and above
* HDD 10 GB Hard Disk Space and Above

### 

### 3.1.3 Software Interfaces

**Operating System: Windows XP or later:**

Microsoft Windows is a meta family of [graphical](https://en.wikipedia.org/wiki/Graphical_user_interface) [operating systems](https://en.wikipedia.org/wiki/Operating_system) developed, marketed, and sold by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It consists of several families of operating systems, each of which cater to a certain sector of the computing industry. Active Windows families include [Windows NT](https://en.wikipedia.org/wiki/Windows_NT), [Windows Embedded](https://en.wikipedia.org/wiki/Windows_Embedded) and [Windows Phone](https://en.wikipedia.org/wiki/Windows_Phone); these may encompass subfamilies, e.g. [Windows Embedded Compact](https://en.wikipedia.org/wiki/Windows_Embedded_Compact) (Windows CE) or [Windows Server](https://en.wikipedia.org/wiki/Windows_Server). Defunct Windows families include [Windows 9x](https://en.wikipedia.org/wiki/Windows_9x) and [Windows Mobile](https://en.wikipedia.org/wiki/Windows_Mobile).

**Database Server: Microsoft SQL Server**

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

**Microsoft Internet Explorer**

Internet Explorer is a series of [graphical](https://en.wikipedia.org/wiki/Graphical_user_interface) [web browsers](https://en.wikipedia.org/wiki/Web_browser) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) and included as part of the [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) line of [operating systems](https://en.wikipedia.org/wiki/Operating_system), starting in 1995. It was first released as part of the add-on package [Plus! for Windows 95](https://en.wikipedia.org/wiki/Microsoft_Plus!) that year. Later versions were available as free downloads, or in [service packs](https://en.wikipedia.org/wiki/Service_pack), and included in the [Original Equipment Manufacturer](https://en.wikipedia.org/wiki/Original_Equipment_Manufacturer) (OEM) service releases of Windows 95 and later versions of Windows.

Internet Explorer is one of the most widely used web browsers, attaining a peak of about 95% [usage share](https://en.wikipedia.org/wiki/Usage_share_of_web_browsers) during 2002 and 2003.[[6]](https://en.wikipedia.org/wiki/Internet_Explorer#cite_note-9) This came after it managed to win the [first browser war](https://en.wikipedia.org/wiki/First_browser_war) against [Netscape](https://en.wikipedia.org/wiki/Netscape), which was the dominant browser in the 1990s. Its usage share has since declined with the launch of [Firefox](https://en.wikipedia.org/wiki/Firefox) (2004) and [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome) (2008), and with the growing popularity of operating systems such as [OS X](https://en.wikipedia.org/wiki/OS_X), [Linux](https://en.wikipedia.org/wiki/Linux), [iOS](https://en.wikipedia.org/wiki/IOS) and [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) that do not run Internet Explorer.

## 3.2 Functional Requirements

## Ports are simply a software abstraction, used to distinguish between communication channels. Similar to the way IP addresses are used to identify machines on networks, ports identify specific applications in use on a single machine. For example, your web browser will by default connect to TCP port 80 of machines in HTTP URLs. If you specify the secure HTTPS protocol instead, the browser will try port 443 by default.

## Nmap works with two protocols that use ports: TCP and UDP. A connection for each protocol is uniquely identified by four elements: source and destination IP addresses and corresponding source and destination ports. All of these elements are simply numbers placed in the headers of each packet sent between hosts. The protocol is an eight-bit field, which specifies what type of packet is contained in the IP data (payload) section. For example, TCP is protocol number six, and UDP is 17. IPv4 addresses have a length of 32-bits, while ports are 16-bits long. IPv6 addresses are 128-bits in length. Further IP, TCP, and UDP header layout details can be found in [the section called “TCP/IP Reference”](https://nmap.org/book/tcpip-ref.html).

## Because most popular services are registered to a well-known port number, one can often guess what services open ports represent. Nmap includes an [nmap-services](https://svn.nmap.org/nmap/nmap-services" \t "_top) file, containing the well-known service for registered port and protocol numbers, as well as common ports for trojan backdoors and other applications that don't bother registering with the Internet Assigned Numbers Authority (IANA). Nmap prints this service name for reference along with the port number.

## Because the port number field is 16-bits wide, values can reach 65,535. The lowest possible value, zero, is invalid. The Berkeley sockets API, which defines how programs are usually written for network communication, does not allow port zero to be used as such. Instead, it interprets a port zero request as a wildcard, meaning that the programmer does not care which is used. The system then chooses an available port number. For example, programmers rarely care what source port number is used for an outgoing connection. So they set it to zero and let the operating system choose one.

## While port zero is invalid, nothing stops someone from specifying it in the header field. Some malicious trojan backdoors listen on port zero of compromised systems as a stealthy way to offer illegitimate access without appearing on most port scans. To combat this, Nmap does allow scanning of port zero when it is specified explicitly (e.g. -p0-65535).

## The first class of valid ports, numbers one through 1,023, are known as reserved ports. Unix systems (unlike Windows) require that applications have special (root) privileges in order to bind to and listen on these ports. The idea is to allow remote users to trust that they are connecting to a valid service started by an administrator and not by some wicked, unprivileged user. If the registered port for SSH was 2,222 instead of 22, a malicious user could start up a rogue SSH daemon on that port, collecting passwords from anyone who connects. As most common server applications listen on reserved ports, these are often the most fruitful to scan.

## The ephemeral port range is another class of ports. This pool of ports is made available by the system for allocation as needed. When an application specifies port zero (meaning “any port”), the system chooses a port from this range. The range varies by operating system, and is usually configurable. It should contain at least a couple thousand ports to avoid running out when many concurrent connections are open. The Nmap connect scan can use hundreds at a time as it scans every specified port on each target machine. On Linux, you can view or set the range using the file /proc/sys/net/ipv4/ip\_local\_port\_range.  shows that on my Linux system, the range is 32,768 to 61,000. Such a large range should be sufficient in almost all cases, but I expand it just to demonstrate how to do so.

# 4. Analysis Models

**Spiral Model**

Here in the for Vulnerb the spiral model will use as analysis model.

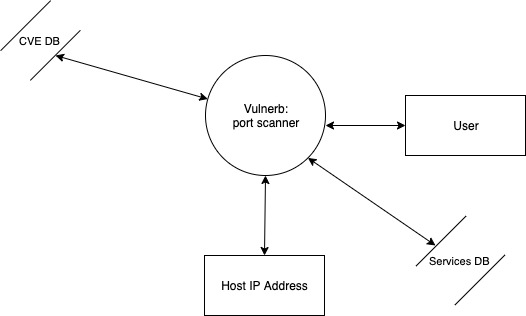
The steps for Spiral Model can be generalized as follows:

* The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
* A preliminary design is created for the new system.
* A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
* A second prototype is evolved by a fourfold procedure:

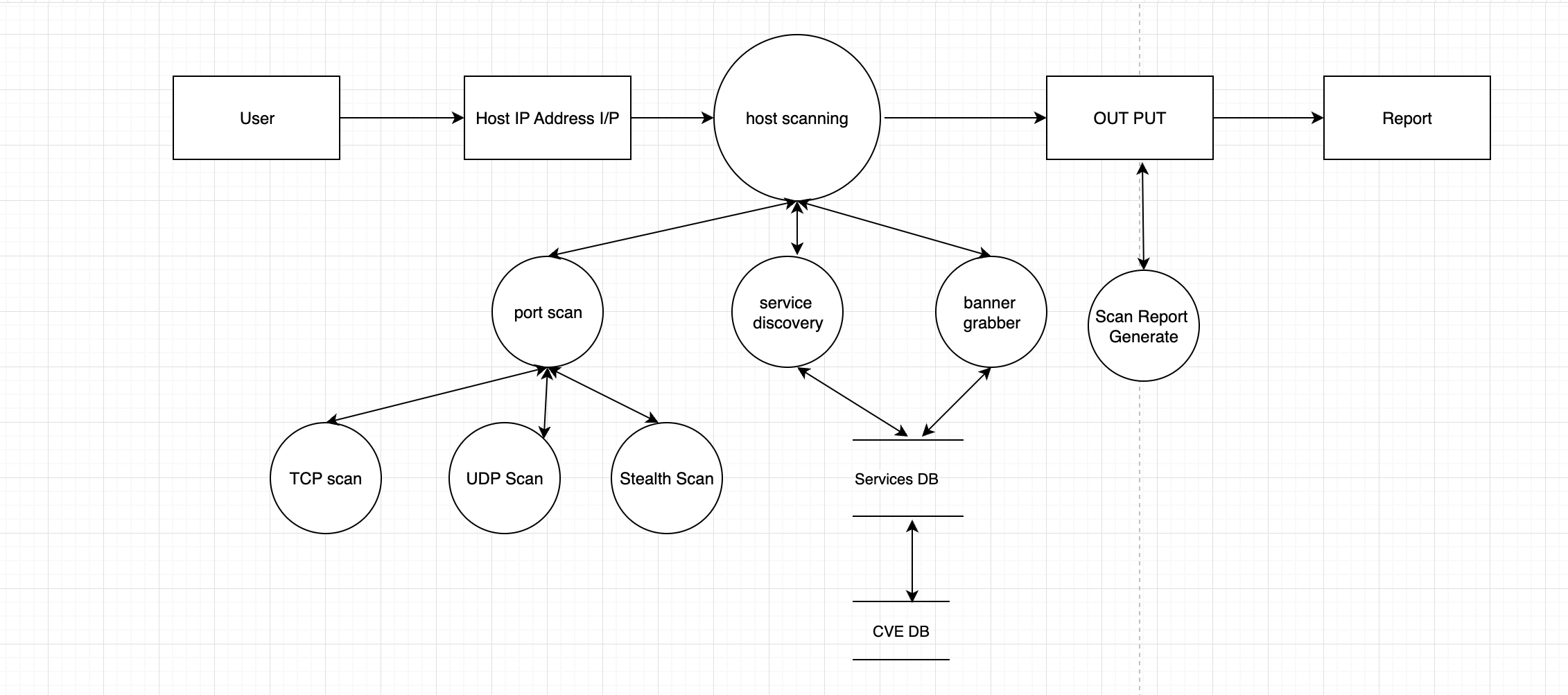
1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning an designing the second prototype.
4. Constructing and testing the second prototype.

* At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.
* The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
* The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
* The final system is constructed, based on the refined prototype.
* The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.
  1. **Data Flow Diagrams (DFD)**

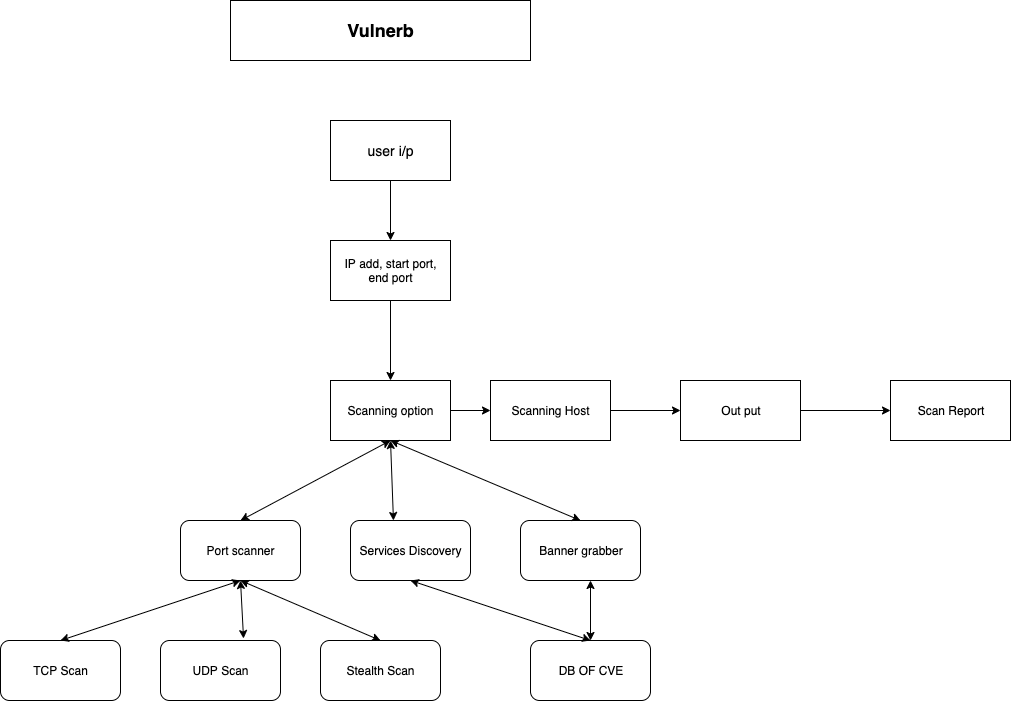
DFD Level 0 :-



DFD Level 1:-



* 1. **State Transition Diagram**

****

**5. LITERATURE SURVEY**

Port scanning permits a hacker to deduce what services are running on the systems that have been pointed out. If vulnerable or insecure services are tracked down, the hacker may be able to exploit these to gain unauthorized access. We have a total of 65,535 \* 2 ports (TCP & UDP). While a complete scan of all these ports may not be feasible, analysis of popular ports should be performed. By port scanning, one is able to find out which ports are accessible. Factually, a port scan consists of sending a message to each port, one at a time and analyzing the response received. If the port is in use, it can then be examined further for weakness. Port Scanning is one of the most favoured reconnaissance techniques which attackers use. By port scanning, one discovers which ports are available (i.e. being listened to by a service). Essentially, a port scan consists of sending a message to each port, one at a time and examining the response received. If the port is in use, it can then be probed further for weakness. Port Scanning is one of the most popular among the reconnaissance techniques attackers use.

In [1] Fyodor has suggested many techniques used to discover what ports (or similar protocol abstraction) of a host are listening for connections. These ports typify potential communication channels. Mapping their existence smooth`s the exchange of information with the host, and thus it is very useful for anyone who wants to investigate their networked environment, including hackers.

 In [2] Marco de Vivo, Eddy Carrasco, Germinal Isern and Gabriela O. de Vivo have set forth that TCP port scanners are distinctive programs used to discover what TCP ports of a host have processes listening on them for viable connections. Since these ports specify, in part, the amount of manifestation of the hosts to potential external attacks, knowing their existence is a elementary matter for network and/or security administrators.

In [3] Pete Herzog has suggested that Port scanning is an invasive examining of system ports on the transport and network level. The paper also includes the validation of system reception to encapsulated, tunneled or routing protocols. This parameter is to calculate live or accessible Internet services as well as penetrating the firewall to discover additional live systems. Testing for different protocols will depend on the system type and services it provides.

In [4] Roger Christopher has described that Port Scanning is one of the most favorable techniques attackers use to find services that they can enslave to break into systems. All systems connected to a LAN or the Internet with a modem run services that listen to the ports which are well-known and not so well-known. By port scanning, the attacker can gather the following information about the targeted systems: what services are executing, under what users those services run, whether anonymous logins are supported or not, and whether certain network services require authentication or not.

In [5] Brenden Claypool have described that Port scanning is a skillful and efficient way which is used by attackers, curious individuals, and administrators to gather information from computers on a network. System and network administrators take the help of port scans to find out open ports to a system so that they may be able to access those ports, or shut them off fully. The way attackers and administrators use port scanning is the same but the only difference lies in their purpose. The attackers use port scanning for malicious purpose. There are many techniques which are used in stealth scanning, ranging from those that prevent their detection by logging systems, identity concealment, to confusing the server with invalid information. All of these techniques are interesting in their implementation and execution.

In [6] Harry Anderson has described that Port scanning appears simple on the surface but is actually a very complicate subject. One factor which makes port scanning tough is the response system. Accuracy, stealth and speed are the principal factors to stabilize when scanning the ports. The factors which affect these are timeouts, the type of scan and what ports to scan. The two most often used types of scans are the SYN scans and connect (). There is disparity of both in Nessus and in the optional NMap component.

In [7] Nazar El-Nazeer and Kevin Daimi have put the light on network port scanning tools. A port is an application noticeable software construct acting as an endpoint in many communications. The Transmission Control Protocol (TCP) and the User Diagram Protocol(UDP) of the Transport Layer mainly uses the ports. Ports are recognized by numbers. For example, Port 25 is used for Simple Mail Transfer, and port 80 is reserved by HTTP. A port scan is an attack that tries to discover known vulnerabilities of a service running on active ports. Both network administrators and attackers use port scanner tools to examine servers/hosts for open ports, but with different purposes.

In [8] Gadge, J. Patil, and A.A. have proposed that Port scanning is a phase in foot printing and scanning; this comes in reconnaissance which is regarded as the first phase of a computer attack. Port scanning aims at finding open ports in a system. These open ports are taken as an advantage by attackers to carry out attacks and exploits. There are a number of tools which are used for scanning open ports. However, very few tools are present to detect port scanning attempts.

 In [9] Zhang and Fang have proposed a new port scan detection approach known as time-based flow size distribution sequential hypothesis testing (TFDS) for transit networks which are having high speed where only unidirectional flow information is available. TFDS makes use of the foremost ideas of sequential hypothesis testing to detect scanners that exhibit abnormal access patterns in terms of flow size distribution entropy.

In [10] Monowar H Bhuyan, D K Bhattacharyya and J K Kalita, have described that the Scanning of ports on a computer occur habitually on the Internet. An attacker conducts port scans of IP addresses to discover vulnerable hosts so as to compromise them. However, it is also helpful for system administrators and other network defenders to discover port scans as possible preparatory measures to more serious attacks. It is a very tough task to recognize instances of malicious port scanning. Port scanning is designed to examine a network host for open ports and other services accessible. From the attacker‟s viewpoint, a port scan is helpful for collecting relevant information for initiating a successful attack. Thus it is of appreciable interest to attackers to determine whether or not the defenders of a network are scanning ports frequently. Defenders do not often conceal their identity during port scanning while attackers do.

 In [11] Mehiar Dabbagh, Ali J. Ghandour, Kassem Fawaz, Wassim El Hajj and Hazem Hajj have suggested that port scanning is generally divided into two main parts, horizontal and vertical. In horizontal scans, the same port is scanned on the multiple hosts. This is helpful for attackers who want to gain access on victim hosts by exploiting a known vulnerability of a definite service running on that port. While in vertical attacks, multiple ports are scanned over the same host. This is common for attackers who are collecting information to attack a particular target host. Port scanning is the most favorable reconnaissance technique which attackers use to determine services they can exploit. Port scanning detection has got a lot of attention by researchers.Nevertheless, a slow port scan attack can defraud most of the existing Intrusion Detection Systems (IDS).

**References**

**Internet Sources:**

**[1]**[**https://nmap.org/**](https://nmap.org/)

**[2]**[**https://www.sciencedirect.com/**](https://www.sciencedirect.com/)

**[3]**[**https://www.techopedia.com/**](https://www.techopedia.com/)

**[4]**[**https://www.python.org/doc/essays/blurb/**](https://www.python.org/doc/essays/blurb/)

**[5] https://www.tenable.com**