VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ISAA PROJECT

TOPIC-PORT SCANNER

Submitted by

Paras Dang 19BIT0378

Aman Gupta 19BIT0306

DECLARATION

We hereby declare that the project report entitled **"Port Scanner"** submitted by us to Vellore Institute of Technology University, Vellore in partial fulfilment of the requirement for the award of the course **ISAA** is a record of bonafide project work carried out by us under the guidance of **Prof. Priya V.**

Title

A port scanner is a piece of software designed to search the network host for open ports. This is often used by administrators to check the security of their networks and hackers to reduce them. An online port scanner will scan your computer for open ports. Systems use boats (like we use doors) to visit and connect with the outside world (net). Viruses now reside in ports that explore the Internet looking for smart computers with open ports; if they find out, they disable your software or worse, remain anonymous and report your personal activity and information to another system. This project is basically about how to scan a local website for services of any remote system connected to the Server with the help of the IP / TCP Address of the system connected to that server. The scanner also integrates with the connection test module (Connect page) and allows the management of the local service site (Resources page). After the hacker has used the port scanner on your system they know which services you have that accept the connection.

Abstract

Port scanning is a method of determining which port in a network are open and which may receive or transmit data. It is also a process of sending packets to specific ports in the host and analyzing responses to determine risk. This scan would not be possible without first scanning the list of active hosts and mapping the host to their IP addresses. This operation, called host detection, begins with a network scan. The goal of port and network scanning is to identify the organization of IP addresses, hosts, and ports in order to accurately identify open or endangered server locations and to identify security levels. Both network and port scanning may reveal the presence of security measures in a firewall-like environment between a server and a user's device. A port scanner enables you stumble on a potential protection breach by using identifying the hosts linked on your network and the services strolling on them, consisting of the report transfer protocol (FTP) and hypertext transfer protocol (HTTP). an advanced port scanner offers you with an in-depth view into your network ports. This consists of details like, linked IP, DNS, and MAC, together with name and information about the interface related to a particular

port. A network port scanner also permits you to identify and near all the open ports on your networks. Port scanning guarantees your community hosts are configured to run only accepted network services, and prevents ports from permitting any unauthorized traffic.

TITLE	ADVANTAGES	DRAWBACKS	METRICS USED TO EVALUATE WORK
TCP Port scanners	Since these ports characterize, in part, the amount of exposure of the hosts to potential external attacks, knowing their existence is a fundamental matter for network and/or security administrators.	As scanners are also used by hackers, administrators need to know how they work and what possible weakness they exploit to be able to prevent unwanted scanning or at least to record each scanning attempt.	Accuracy
Linux platform for Port Scan	Regardless of the limitation of processing power, the system performance on the embedded platform is at par with other port scanners running on a much better performance PC.	The findings indicate that low end embedded Linux platform is suitable for network security application and it is marketable at a lower cost with the extra benefit ofportability.	Sensitivity

Automated Port and Vulnerability Scanner	The purpose of this report is to design and document an automated network scanning and vulnerability assessment tool using the Network Mapper (Nmap) Scripting Engine (NSE) integrated with VulScan and Nmap_vulners modules databases written in BASH scripting language.	May cause difficulties switching from a tool to the other. The main goal of MDXploit is to bridge the gap by automating the network scanning process and vulnerability assessment of small or large networks	Precision
Embedded port scanner	It is interesting enough to find that regardless of the limitation of power, the system performance on the embedded platform is at par with other port scanners running on a much better performance PC	There is limitation of processing power in this system.	Accuracy
Half-connection port scanner	For its security reliability,it is widely used in information detection of network penetration tests.	Not very reliable	Sensitivity

Port Scanner	responds in 15.6 seconds which is much better than the proposed models. the choice of port it limits to hosts of 1024 with UDP or some TCP and scan finishes in 15 seconds	It maximum scan, sending rate used is 100 packets per second	Takes less Time
ZMap	The advantage of the half-open scan is relevant to the situations when some hosts were unreachable or had closed ports. An exchange of three packets to unreachable hosts or closed ports would have been inefficient and would have wasted the packets.	There needed to be a compromise between the scan speed and the speed that the local network and machine hardware could handle ZMap supports multiple network probes for the same target host, it was decided to use the default of one probe per host.	Takes less Time
System for Detecting a Port Scanner in 4G WCDMA Mobile Networks	High speed	Data Trafficking	High Accuracy High Speed

Rule-Based Network Intrusion Detection System for port Scanning with efficient Port Scan Detection Rules Using Snort	This rule-based Intrusion Detection System we will match the signature with our Efficient Port Scan Detection Rules (EPSDR) from captured packet. As a definition of signature based IDS this new EPSDR based IDS will be useful to reduce the false positive alarm.	This rule-based Intrusion Detection System we will match the signature with our Efficient Port Scan Detection Rules (EPSDR) from captured packet. As a definition of signature based IDS this new EPSDR based IDS will be useful to reduce the false positive alarm.	False positive alarm

Dual port scanning	The developed system achieved high accuracy and reliability	Only for EM characterstics	Accuracy
Port scanner in 3g WCDMA Mobile networks	The results showed that this proposed system is practical and effective	Unlike the traditional wired systems ,they have limited wireless resources and signalling procesured	Positive alarm
High Speed port scanner	This thesis research showed that the scan speed was substantially faster and more accurate than previous Web	Not much of limitation in this system	Less time taker

	census techniques.		
Port scanner Embedded	Performance on the embedded platform is at par with other port scanners running on a much better performance PC	There is limitation of processing power in this system.	Accuracy

Summary(Germinal Isern)

This paper reports on the most important techniques used by TCP port scanners. TCP port scanners are special systems used to determine which TCP host ports have processes in them to connect potential. As these ports point out, in part, the amount of hackers' exposure to potential external attacks, knowing their presence is an important issue for network and / or security controllers. In addition, since scanners are also used by hackers, managers need to know how they work and what potential vulnerabilities they use in order to prevent unwanted scanning or at least record every scanning attempt.

Summary(Moona Olakara Mohammed)

In a computer network, an attack is an attempt to destroy or steal unauthorized information or to use information as an asset. Another attack is a conscious attack which is considered the first step in a computer attack. This type of attack is mostly done by a black hat, an expert editor, by scanning internal network devices and collecting vulnerability information. In this paper, it shows the identification of open ports and services over the network and the IP available on the network can be attacked.

Summary(Li Jirong,Zeng Aiguo)

A new pressure measurement system, which will deal with pressures of up to 46 models, has been installed in 12-in. Transonic tunnel. The system consists of a Scanivalve, two 24-port cutoff valves, a pressure transducer, a strip chart recorder, and an electronic control unit. The 46 compressed models were recorded in sequence on the strip chart recorder immediately after the 7-

second test was hit. This report describes the scanner system and the associated electrical control unit

Summary(LS)

This study demonstrates the development of a two-port free measuring system to measure full-field electromagnetic (EM) features. The program contains a free space measuring system with two antennas, a two-axis line platform, and a mechatronic synchronization that allows automatic measurement of full-field EM structures, such as loss of return and input (), clearance (), and accessibility.

Summary(LS)

In the field of network security, researchers have used different models to protect the network. The Intrusion Detection System is also one of them and Snort is an open source tool for Intrusion Detection and Prevention System. Today The Intrusion Detection System is a growing technology in network security and especially researchers have focused on this field, some of them using a legal signature or process and some strategies that are uniquely based to improve network security. In this paper we propose a Legal Entity Acquisition Principle for which we have developed the New port Scan Detection Act (EPSDR). These rules will be used to detect port scan naive attacks on a real-time network using the Snort and Basic Analysis Security Engine (BASE).

Summary(LS)

This paper introduces our effort and identifies possible uses of the embedded Linux platform for access (Port Scan). The method used was to develop software that enables port scanning using slow opening method and udp. The software is then used on Linux based Single Board Computer (SBC) using the TS-Linux 2.4.23 kernel developed by Technology System (TS). It is interesting enough to find that despite the limitations of processing power, the performance of the system in embedded space is similar to other port scanners running on a PC that works much better. The findings show that the low-cost Linux embedded platform is suitable for secure network use and is sold at low cost with the added portable benefit.

Summary(LS)

A high-speed port scanner, ZMap, was used to perform extensive Internet scanning Port 80 in an attempt to determine the size of the Web. The size of the Web was estimated from the number of web servers detected during high-speed IPv4 scanning addresses. The focus of the study was to determine whether the census was accurate in a enough time using ZMap. Metrics to determine the success of this research has been a reduction in the time required to perform extensive internet scanning, in comparison historical experiments, and comparisons between scanning data collected in this study by data collected by Rapid7 Labs to check the accuracy of ZMap scanning.

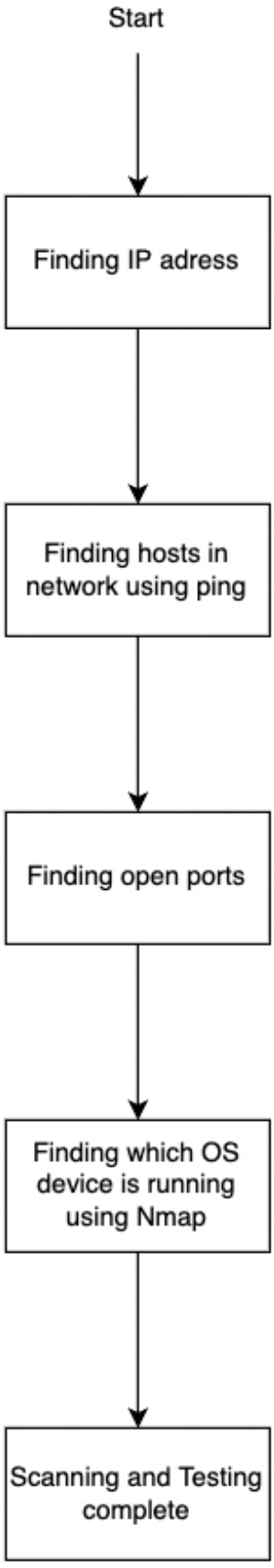
Summary(LS)

Workshop hosted by the University Computer Institute and Chair of the Computer Network and Distributed Programs (Faculty of Computer Science) at Chemnitz University of Technology. Workshop topic: ¨Digital University infrastructure¨ Port scanners are software tools used to determine information about computer systems online. Such tools are often used by attackers to detect compromised computer systems. The talk introduces the function and techniques used for free port scanners. In addition, tools are introduced when port scanning processes can be detected.

Summary(LS)

The main purpose of this program is to build a system automatic port scanning process to scan the wport the network ports of the target information system about target hosts, listening ports, and services running through ports. There are too many network scans network development and remediation tools. these communication tools that are publicly disclosed could used by purposeful criminals and attackers for a vicious reason. One of the tools is analyzed in my system is the Nmap tool. As a result of scanning the host, point to the open ports again services and related IP.

SYSTEM ARCHITECTURE


Start
Finding IP adress
Finding hosts in network using ping
Finding open ports
Finding which OS device is running using Nmap
Scanning and Testing complete

PSEUDO CODE

START
First we will find the IP address
2)Then we will be starting our scan
3)We will find all the hosts in network using ping
4)Then we will Find open ports in these active hosts
5)Then we will look for malicious ports and find which OS device is running using NMAP

IMPLEMENTATION CODE

```
import os
import socket
import threading
import subprocess

ping_lock = threading.Lock()
def is_host_active( host, all_active_hosts):
  ping_resp = os.popen("ping -c 1 -t 2 " + host)

  for line in ping_resp.readlines():
    if (line.count("ttl")):
      with ping_lock:
        all_active_hosts.append(host)
      break

nmap_lock = threading.Lock()
def find_os(host, all_os):
  scanv = subprocess.Popen(["nmap", "-PR", "-O", str(host)],
stdout=subprocess.PIPE, stderr=subprocess.PIPE).communicate()[0]
  scanlist = scanv.split()

  with nmap_lock:
    if 'printer' in scanlist:
      all_os[host] = 'Printer'
    elif 'Linux' in scanlist:
      all_os[host] = 'Linux'
```

```
    elif 'Windows' in scanlist:
      all_os[host] =  'Windows'
    elif 'Apple' in scanlist:
      all_os[host] = 'Apple'
    elif 'IOS' in scanlist:
      all_os[host] = 'IOS'
    else:
      all_os[host] = 'Unknown'

tcp_scan_lock = threading.Lock()
def tcp_scan(host, all_open_ports):
  open_ports = list()
  for port in range(1, 10000):
    try:
      print("Looking at " + host + " for port" + str(port))
      tcp_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
      tcp_socket.connect((host, port))
      open_ports.append(port)
      tcp_socket.close()

    except Exception:
      pass

  with tcp_scan_lock:
    all_open_ports[host] = open_ports

# Find my IP address.
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.connect(("8.8.8.8", 80))
my_ip_address = s.getsockname()[0]
print("My IP address: " + my_ip_address)
s.close()

print("Starting scan...")
ip_split = my_ip_address.split('.')
network_prefix = ip_split[0] + '.' + ip_split[1] + '.' + ip_split[2]

# Find all hosts in the network using ping
all_active_hosts = list()
all_ping_threads = list()
```

```
for host_ip in range (2,255):
  host_addr = network_prefix + '.' + str(host_ip)
  ping_thread = threading.Thread(target=is_host_active, args=(host_addr,
all_active_hosts,))
  ping_thread.start()
  all_ping_threads.append(ping_thread)

for _, thread in enumerate(all_ping_threads):
  ping_thread.join()

all_active_hosts.sort()

# Find open ports in these active hosts
all_host_open_ports = {}
all_tcp_threads = list()
for _, host in enumerate(all_active_hosts):
  tcp_thread = threading.Thread(target=tcp_scan, args=(host,
all_host_open_ports,))
  tcp_thread.start()
  all_tcp_threads.append(tcp_thread)

for _, thread in enumerate(all_tcp_threads):
  thread.join()

#Find OS device is running using NMAP
all_host_os = {}
all_nmap_threads = list()

for _, host in enumerate(all_active_hosts):
    nmap_thread = threading.Thread(target=find_os, args=(host, all_host_os,))
    nmap_thread.start()
    all_nmap_threads.append(nmap_thread)

for _, thread in enumerate(all_nmap_threads):
  thread.join()

print("Scan complete, Stats below:")
for _, host in enumerate(all_active_hosts):
  print ("Address: {0} OS: {1} Open ports: {2}".format(host, all_host_os[host],
all_host_open_ports[host]))
```

```
import os
import socket
import threading
import subprocess

ping_lock = threading.Lock()
def is_host_active( host, all_active_hosts):
  ping_resp = os.popen("ping -c 1 -t 2 " + host)

  for line in ping_resp.readlines():
    if (line.count("ttl")):
       with ping_lock:
         all_active_hosts.append(host)
       break

nmap_lock = threading.Lock()
def find_os(host, all_os):
  scanv = subprocess.Popen(["nmap", "-PR", "-O", str(host)],
stdout=subprocess.PIPE, stderr=subprocess.PIPE).communicate()[0]

  scanlist = scanv.split()
  print(str(scanlist))
  with nmap_lock:
    if 'printer' in scanlist:
      all_os[host] = 'Printer'
    elif 'Linux' in scanlist:
      all_os[host] = 'Linux'
    elif 'Windows' in scanlist:
      all_os[host] =  'Windows'
```

```
    elif 'apple' in scanlist:
      all_os[host] = 'apple'
    elif 'IOS' in scanlist:
      all_os[host] = 'IOS'
    else:
      all_os[host] = 'Unknown'

tcp_scan_lock = threading.Lock()
def tcp_scan(host, all_open_ports):
  open_ports = list()
  for port in range(1, 10000):
    try:
      print("Looking at " + host + " for port" + str(port))
      tcp_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
      tcp_socket.connect((host, port))
      open_ports.append(port)
      tcp_socket.close()

    except Exception:
      pass

  with tcp_scan_lock:
    all_open_ports[host] = open_ports

# Find my IP address.
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.connect(("8.8.8.8", 80))
my_ip_address = s.getsockname()[0]
print("My IP address: " + my_ip_address)
```

```
my_ip_address = s.getsockname()[0]
print("My IP address: " + my_ip_address)
s.close()

print("Starting scan...")
ip_split = my_ip_address.split('.')
network_prefix = ip_split[0] + '.' + ip_split[1] + '.' + ip_split[2]

# Find all hosts in the network using ping
all_active_hosts = list()
all_ping_threads = list()
for host_ip in range (2,255):
  host_addr = network_prefix + '.' + str(host_ip)
  ping_thread = threading.Thread(target=is_host_active, args=(host_addr,
all_active_hosts,))
  ping_thread.start()
  all_ping_threads.append(ping_thread)

for _, thread in enumerate(all_ping_threads):
  ping_thread.join()

all_active_hosts.sort()

# Find open ports in these active hosts
all_host_open_ports = {}
all_tcp_threads = list()
for _, host in enumerate(all_active_hosts):
  tcp_thread = threading.Thread(target=tcp_scan, args=(host,
all_host_open_ports,))
```

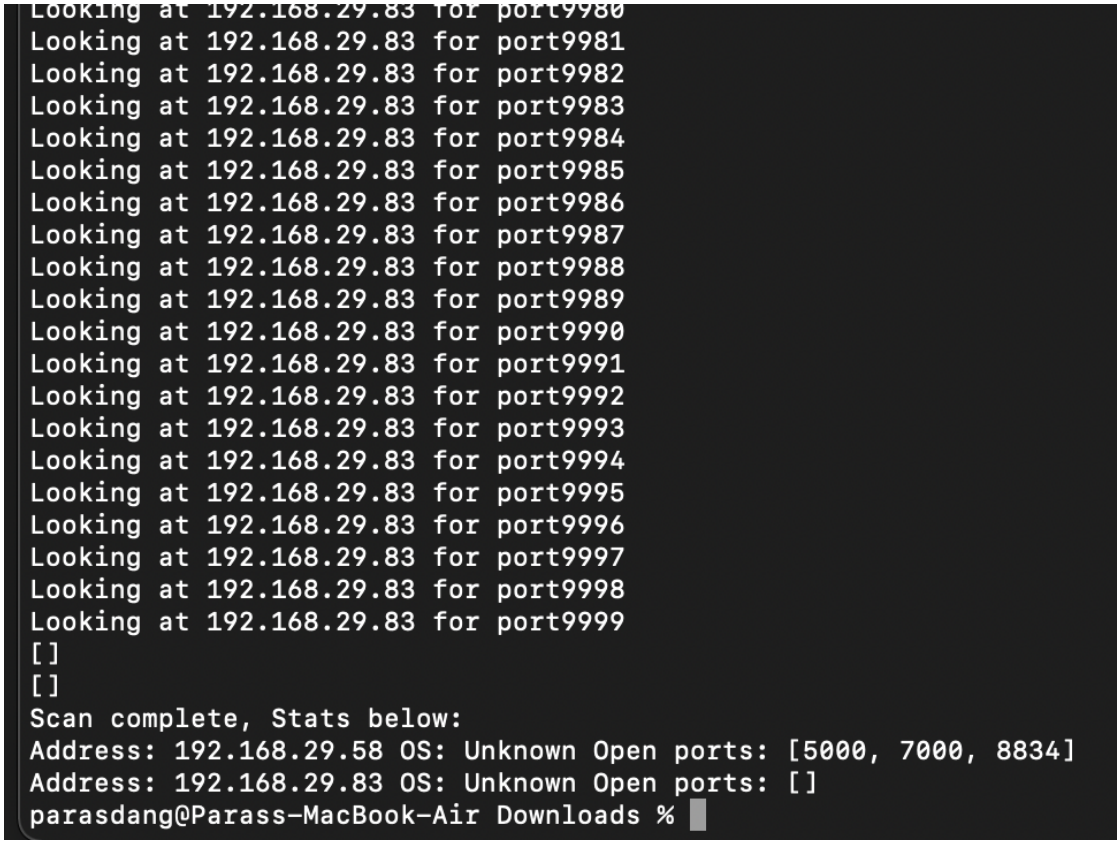
```
#Find OS device is running using NMAP
all_host_os = {}
all_nmap_threads = list()

for _, host in enumerate(all_active_hosts):
    nmap_thread = threading.Thread(target=find_os, args=(host,
all_host_os,))
    nmap_thread.start()
    all_nmap_threads.append(nmap_thread)

for _, thread in enumerate(all_nmap_threads):
  thread.join()

print("Scan complete, Stats below:")
for _, host in enumerate(all_active_hosts):
  print ("Address: {0} OS: {1} Open ports: {2}".format(host,
all_host_os[host], all_host_open_ports[host]))
```

```
parasdang@Parass-MacBook-Air Downloads % python host_scanner.py
My IP address: 192.168.29.58
Starting scan...
ping: sendto: No route to host
Looking at 192.168.29.58 for port1
 Looking at 192.168.29.83 for port1
Looking at 192.168.29.58 for port2
Looking at 192.168.29.58 for port3
Looking at 192.168.29.58 for port4
Looking at 192.168.29.58 for port5
Looking at 192.168.29.58 for port6
Looking at 192.168.29.58 for port7
Looking at 192.168.29.58 for port8
Looking at 192.168.29.58 for port9
Looking at 192.168.29.58 for port10
Looking at 192.168.29.58 for port11
Looking at 192.168.29.58 for port12
Looking at 192.168.29.58 for port13
Looking at 192.168.29.58 for port14
Looking at 192.168.29.58 for port15
Looking at 192.168.29.58 for port16
Looking at 192.168.29.58 for port17
```



Calculations

Total addresses = A
Time to discover H hosts = A/T

Discovered hosts = H
Number of ports checked = P
Time to discover H hosts
Time to connect to P port = H * P

If we execute the program serially, the time complexity would be O(H *P).

But because of parallel execution on T threads, time complexity is O((H * P)/ T
H = O(A)

Total time = A/T + (H*P)/T
= A/T + (A*P)/T
= (A*P)/T

SUMMARY AND CONCLUSION

Port scanner is an application designed to detect a server or host for open ports. Such an application can be used by administrators to verify the security policies of their networks and attackers to identify network services that work on hackers and exploit the risk.
In our project we first identified the IP address of various devices and then we found all the host in network using ping and we used threading so as to detect fast OS .Without a port scanner Some ports might remain continually open, presenting a potential network vulnerability. An intruder can access an open port to create difficulties in the normal flow of network operations. Network ports should be closely monitored by an effective advance network port scanning tool to avoid any data leakage. This also helps secure communications between the computing entities in the network.

FUTURE WORK

Our project was telling all the open ports and dangerous ports so we are planning to improve the system that it automatically highlights all the dangerous ports and closes them. Running port scans without authorization can be considered an aggressive action, and if we are on a shared network, we might scan a system that isn't under our control, which isn't good. Port scans are a critical part of building a good defence from cyberattacks. Attackers are using port scans, as well. We need to beat them to the punch and close down possible attack vectors and make their lives as difficult as possible.Thanks Maam for giving us this wonderful opportunity to work at this great project .

REFERENCES

https://www.researchgate.net/publication/220195237_A_review_of_port_scanning_techniques

https://www.researchgate.net/publication/345959645_Automatic_Port_Scanner

https://www.researchgate.net/publication/251873688_Embedded_Port_Scanner_EPSS_System_using_linux_and_Single_Board_Computer

https://www.researchgate.net/publication/350819163_MDXploit_An_Automated_Port_and_Vulnerability_Scanner

https://en.cnki.com.cn/Article_en/CJFDTotal-JZCK201007063.htm
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.638.1039&rep=rep1&type=pdf#page=33

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.638.1039&rep=rep1&type=pdf#page=33

https://www.osti.gov/biblio/4338629

https://iopscience.iop.org/article/10.1088/1361-6501/aac7fd/meta

https://www.earticle.net/Article/A280274

https://www.earticle.net/Article/A280274

https://ieeexplore.ieee.org/abstract/document/4786717

https://scholar.acadiau.ca/islandora/object/theses:393/datastream/PDF/file.pdf

https://monarch.qucosa.de/landing-page/?tx_dlf[id]=https%3A%2F%2Fmonarch.qucosa.de%2Fapi%2Fqucosa%253A17604%2Fmets

https://ijisrt.com/assets/upload/files/IJISRT20SEP503.pdf