



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David



School of
Applied Computing

Investigate, Analyse and Demonstrate the Importance of Network Security and Penetration Testing

Author: Sophie Slaughter (1909705) BSc Applied Computing
University of Wales Trinity St. David SA1 Campus
Email: 1909705@student.uwtsd.ac.uk

1. Introduction

The idea of the project is based on a realistic ongoing issue within the network security sector. A Network Vulnerability Scanner will highlight the importance of network security and is worth an investigation into as security plays a crucial part within networks today.

2. Aims and Objectives

The aim of the project is to design a syntax prototype network scanner that helps users successfully run basic scans to uncover network vulnerabilities

Objectives:

- Conduct a literature review on network vulnerability scanners.
- Design the prototype based on a syntax code that's discovers vulnerabilities.
- Testing within a configured simulation where the results of different programs are produced.
- Evaluation of what the users will benefit from the vulnerability scanner.

3. Research

The key points of the research can be seen below:

1. An in-depth understanding of Vulnerability Scanners
2. Different scans that can be completed [1]
3. Protection against recent attacks [2]
 - Adware
 - Backdoor
 - Botnet
4. Configuring a Scanner
 - Researched Content contributed to the development process
 - Resolve Mistakes Made

4. Design and Implementation

A syntax code prototype was designed and implemented to produce the results for the vulnerability scan. The Implementation included the following.

- VirtualBox
 - Kali Linux operating system
 - Metasploitable operating system
 - Python

5. Results

The results can be seen within figures 1-4. Figure 1 identifies open ports within the tested device. Figure 2 discovers specified port with coloured results. Figure 3 identifies the vulnerable content contained from a specified file. Figure 4 discovers any vulnerabilities contained on a specified open port.

6. Conclusion

Clearer understanding of Vulnerability Scanners has been achieved from the Literature Review Section
Improved development skills when creating the programs

- Structuring the code

Improved understanding of Python and the use of Virtual Environments
Understand where this project can be taken further for future work.

- Further Development

List of Figures

1. Port Scan
2. Specified amount of Port Scanning
3. Banner Retriever
4. Vulnerability Scanning

```
#!/usr/bin/python
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
socket.setdefaulttimeout(3)

host = input("Enter the Host to Scan: ")
port = int(input("Enter the Port to Scan: "))

def portscanner(port):
    if sock.connect_ex((host,port)):
        print ("Port %d is closed" % (port))
    else:
        print ("Port %d is opened" % (port))

portscanner(port)
```

Figure 1

```
#!/usr/bin/python
import socket
from termcolor import colored

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

host = input("[*] Enter the Host to Scan: ")

def portscanner(port):
    if sock.connect_ex((host,port)):
        print (colored("[*] Port %d is closed" % (port), 'red'))
    else:
        print (colored("[*] Port %d is open" % (port), 'green'))

for port in range(1024):
    portscanner(port)
```

Figure 2

```
#!/usr/bin/python
import socket

def retBanner(ip, port):
    s = socket.socket()
    s.connect((ip, int(port)))
    s.settimeout(3)
    print (s.recv(1024))

def main():
    ip = input("Please enter the IP: ")
    port = int(input("Please enter the port: "))
    retBanner(ip, port)
```

Figure 3

```
#!/usr/bin/python
import socket
import os
import sys

def main():
    if len(sys.argv) == 2:
        filename = sys.argv[1]
        if not os.path.isfile(filename):
            print ("[-] File Does Not Exist!")
            exit(0)
        if not os.access(filename, os.R_OK):
            print ("[-] Access Denied!")
            exit(0)
    else:
        print ("[-] Usage: " + str(sys.argv[0]) + " <vulnerable filename>")
        exit(0)
    for x in range(99,101):
        ip = "192.168.1." + str(x)
        port = str(input("Please enter the port: "))
        banner = retBanner(ip, port)
        checkVuln(banner, filename)
```

Figure 4

Result

Result

Result

[1] 'Vulnerability Scanner: what is it and how does it work? | Snky'. <https://snyk.io/learn/vulnerability-scanner/> (accessed Jan. 18, 2022).
[2] 'Live Cyber Threat Map | Check Point'. <https://threatmap.checkpoint.com/> (accessed Feb. 25, 2022).