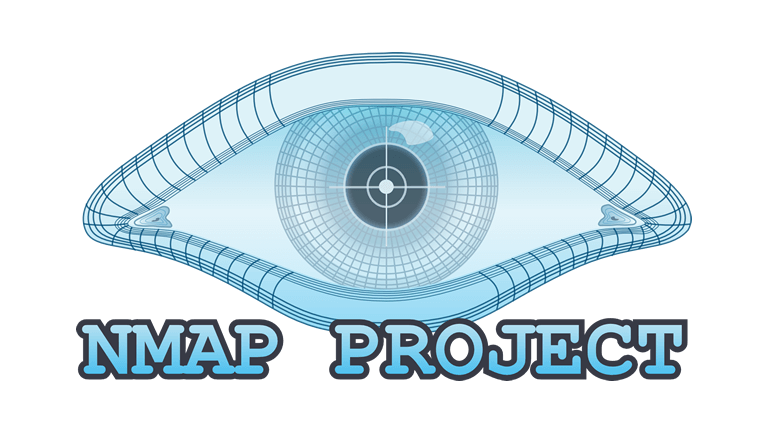
|  |  |
| --- | --- |
| KINGDOM OF SAUDI ARABIA  Ministry of Education  King Saud University  College of Computer and Information Sciences |  |

SEC505 Network Security

Project Proposal



**Auto-Nmap Analyzer**

|  |  |
| --- | --- |
| Sumaya Altamimi | Project Leader |
| Nouf Almutawa | Team member |
| Sarah Alsubaie | Team member |
| Supervised by Dr. Abdulrahman Almutairi | |

Table of Contents

[INTRODUCTION 3](#_Toc69156236)

[PROBLEM 3](#_Toc69156237)

[SOLUTION 3](#_Toc69156238)

[Project objectives 3](#_Toc69156239)

[PROJECT SCOPE 4](#_Toc69156240)

[ROLES AND RESPONSIBILITIES 4](#_Toc69156241)

[Research Methodology 4](#_Toc69156242)

[Experiment setup 6](#_Toc69156243)

[Tool description 6](#_Toc69156244)

[Data Gathering 6](#_Toc69156245)

[Data Analysis 6](#_Toc69156246)

[Anomaly Detection 7](#_Toc69156247)

[Pseudocode for anomaly detection: 8](#_Toc69156248)

[Report Generation 9](#_Toc69156249)

[Scenarios 10](#_Toc69156250)

[First Scenario: Run Nmap with two new open ports 10](#_Toc69156251)

[Data Analysis: 10](#_Toc69156252)

[Anomaly Detection: 10](#_Toc69156253)

[Second Scenario: Run Nmap with two (but not new) open ports 10](#_Toc69156254)

[Data Analysis: 10](#_Toc69156255)

[Anomaly Detection: 11](#_Toc69156256)

[Third Scenario: Run Nmap with 18 new open ports 11](#_Toc69156257)

[Data Analysis: 11](#_Toc69156258)

[Anomaly Detection: 12](#_Toc69156259)

[Performance Evaluation 12](#_Toc69156260)

[Conclusion and future work 13](#_Toc69156261)

[REFERENCES 13](#_Toc69156262)

# INTRODUCTION

Network security is one of the biggest concerns nowadays. Many different types of network threats and attacks exist, and they are increasing day by day. Several scanning tools and packet sniffing tools are available, and one of the most widely-used tools that can analyse and capture the traffic in the network is Nmap.

Nmap stands for network mapper. Nmap is a network scanner; it discovers hosts and services and therefore creates a “map” of the network [1]. Particularly in Nmap, crafted packets are sent to the destination host, and then; it analyses the responses to those packets [1]. Nmap can determine which hosts on the network are available, what services are provided by those hosts, what operating systems they are running, and many other features [2].

# PROBLEM

Although Nmap has a lot of features and is one of the most widely used tools in the field of network security, extending this tool to further enhance the work of its users is worth considering. Nmap users, mainly network administrators and security analysts, have a routine job that is time consuming as well as inaccurate. Manually checking the network and its open ports almost daily is not professional and unacceptable in the age of automation.

# SOLUTION

This project aims to give nmap users, especially network administrators a way to determine the network status automatically by extending the existing tool Nmap. We are going to automate Nmap tool to enhance the whole process.

# Project objectives

1- Facilitate the work of network administrators by **collecting** network data automatically each given period of time.

2- **Analyse** the collected network data periodically.

3- **Detect** anomaly or unusual traffic in the network.

4- Send **alarm** in case of anomaly selection.

5- Generate periodic **reports** of the network status.

## 

# PROJECT SCOPE

The application will target network administrators, security analysts, and even hackers who already used Nmap usually.

# 

# ROLES AND RESPONSIBILITIES

|  |  |
| --- | --- |
| **Role** | **Participant(s)** |
| **Project supervision** | Dr. Abdulrahman Almutairi |
| **Project Leadership** | Sumaya Altamimi |
| **Automation/Data gathering** | All team members. |
| **Data analysis and detection** | All team members. |
| **Report generation** | All team members. |

# Research Methodology

Our project contributes to some existing projects by proposing a new approach to solve one of the issues, which is automation. Our approach is also one of the future directions that could enhance the network security. In order to achieve our objectives to automate and enhance Nmap tool, different network traffic monitoring and analysis techniques should be used.

Based on [3] There are two effective ways in monitoring and analysis of network traffic which are: router-based monitoring techniques and non-router-based monitoring techniques. We will be going to use two of them to facilitate Network Administrators work.

For router-based monitoring technique, after some investigation we find that Remote Monitoring (RMON) RFC 1757 seems to be the most suitable Tanique that we can use it as we just need to announce any unusual events.

RMON: it enables various network monitors and console systems to exchange network-monitoring data. RMON is able to set alarms that will monitor the network based on certain criteria. RMON allows Administrators to manage local networks as well as remote sites from one central location. It monitors at the Network Layer and below.

Monitoring groups that obtain information provided by RMON:

* Statistics: stats for each monitored interface on this device
* History: records periodic statistical samples from a network and store it
* Alarm: periodically takes statistic samples and compares them with a set of thresholds for event generation
* Host: contains statistics associated with each host discovered on the network
* HostTopN: prepares tables that describe top hosts
* Filters: enable packets to be matched by a filter equation for capturing events
* Packet capture: captures packets after they flow through the channel
* Events: controls generation and notification of events from a device
* Token ring: supports token ring

In the other side Non-Router Based Techniques are classified as either active or passive. Besides, a combination of active and passive monitoring is available and can be used which better that using one of them.

For this section we will generate a technique called Self Configuring Network Monitor (SCNM) into our Auto-Nmap which is a monitoring tool that uses a combination of active and passive measurements to collect information at layer 3 ingress and egress routers and at other significant points within the network being monitored. The SCNM environment consists of both hardware and software components. The hardware installed at critical points in the network and it responsible for passively collecting the packet headers while the software runs on the endpoints of the network.

When a problem is detected by the passive monitoring tools, traffic can be generated using the active tools. SCNM allowing the study of only the part of network that seems to be having the problem.

We will be going to generate Comprehensive network management report of the current status of the network.

Our project contributes to some existing projects by proposing a new approach to solve one of the issues, which is automation. Our approach is also one of the future directions that could enhance the network security. In order to achieve our objectives to automate and enhance Nmap tool, different network traffic monitoring and analysis techniques should be used.

Nmap techniques and modules have been generated in out project. Specifically, we used Nmap scan for ports, one of Nmap modules called NSE scripts, it provides service scripts which run against specific services listening on a target host. The results of this process should be saved in a separate file to being the analyses phase.

# Experiment setup

We are going to create a tool that can be used to automate some tasks in Nmap. The development environment for our project would be as follows:

* Kali LINUX as a virtual machine (VMware, VirtualBox)
* Python 3
* Shell Bash Script
* Visual Studio Code and Python extensions installed.
* Metasploitable2 as the vulnerable machine with many open ports.

The program will ask the user for the IP address that they want to scan and produce a pdf report of the results.

# Tool description

We created a Bash Script to automate Nmap for port scanning. The script took an IP address as an argument, run an Nmap scan on the target, analyse the results, and produce a report of the analysis.

The report contains graphs to indicate the state of the target and a warning message if new ports are opened for the first time, that is not seen previously.

## Data Gathering

For data gathering from Nmap to the tool, we found it better to change the format to CSV files which will simplify the analysis process. List of files are stored permanently and renamed to the current date and time.

A screenshot of a computer

Description automatically generated with medium confidence

## Data Analysis

For data analysis of the currently gathered CSV file, we used Numpy, Pndas, and Matplotlib python libraries. Several statistical techniques used enhance the user experience, and maintain user engagement. Examples are shown below:

Below is the cover page of the report which include the date and the IP address currently analysed:

Diagram

Description automatically generated

Below figure shows the number of ports in each state:

Chart, shape

Description automatically generated

Below figure shows the ports that are open:

A picture containing chart

Description automatically generated

## Anomaly Detection

For anomaly detection, the tool compares all files against the currently obtained file of open ports. If there is a new port in the current file that is not exist in the previous files, the tool display a warning message in the report as what will be shown in the following sections. It is so accurate and work exactly as expected. The Pseudocode and rationale behind this Anomaly Detection feature explained in the following section.

### Pseudocode for anomaly detection:

SET susp\_list = []

FOR file IN CVS\_Files:

IF file EQUALS current\_file\_name:

continue;

ADD file TO files

with open file as myfile:

TRY:

SET df TO read\_csv(myfile)

EXCEPT:

continue;

FOR port IN current\_file['PORT']:

IF port not IN df.values:

**IF port not IN susp\_list:**

ADD port to susp\_list

SET susp\_df TO DataFrame(susp\_list)

FOR file IN CVS\_Files:

IF file EQUALS current\_file\_name:

continue;

files.append(file)

with open file as myfile:

TRY:

SET df TO read\_csv(myfile)

EXCEPT:

continue;

FOR port IN susp\_df['PORT']:

**IF port IN df.values:**

SET susp\_df TO susp\_df[susp\_df.PORT != port]

SET susp\_df TO susp\_df.drop\_duplicates()

IF len(susp\_df)>0:

SET warning\_message TO "\n WARNINIG: "+str(len(susp\_df))+" NEW PORTS DETECTED!!"

ELSE:

SET warning\_message TO 'NO NEW PORTS DETECTED.. '

RETURN warning\_message

## Report Generation

Each time the tool is executed, it produces a pdf report that contains all details of the analysis. All reports are stored in the archive and available whenever needed.

Graphical user interface, application

Description automatically generated

# Scenarios

## First Scenario: Run Nmap with **two new open ports**

For the first scenario, only Domain and mysql services are open. The result is as follow:

### Data Analysis:

Chart, bar chart

Description automatically generated

### Anomaly Detection:

Graphical user interface, text

Description automatically generated with medium confidence

## Second Scenario: Run Nmap **with two (but not new) open ports**

For the second scenario, again, only Domain and mysql services are open. But this time both of them are already seen previously and not new. So, simply run the tool again with the same IP. The result is as follow:

### Data Analysis:

Chart, bar chart

Description automatically generated

### Anomaly Detection:

Graphical user interface, text

Description automatically generated

## Third Scenario: Run Nmap **with 18 new open ports**

For the third scenario, we will have services open. The result is as follow:

### Data Analysis:

Chart, bar chart

Description automatically generated

### Anomaly Detection:

Graphical user interface, text, application

Description automatically generated

# Performance Evaluation

We need large amount of data about home networks/LANs and their properties to accurately evaluate our tool. Half of the data is about network with Auto-Nmap Analyzer installed, other half is network data without the Auto-Nmap Analyzer applied. Then, we will conduct A/B testing, with our null hypothesis as the

Auto-Nmap Analyzer will not enhance the network security while the alternative hypothesis proves that the Auto-Nmap Analyzer will enhance the security of networks, mainly local networks for this study. We will also fit a multiple linear regression model to observe the relationships between our explanatory variables and the level of security as the response variable.

We are planning to interpret our results by using visualization techniques, such as scatter plots with the help of existing tools such as Tableau. It is expected for the Auto-Nmap Analyzer to solve the randomness, inaccuracy and resource idleness which might be powerful and a good starting point to more significant tools.

# Conclusion and future work

In this research, we tried to automate Nmap tool in order to enhance network administrator work. We create a script shell and make use of Nmap techniques that scan a specific host port in order to find open ports then export the results to a file. Our future work will include the analyses and detection phase of those open ports, we will make sure that the tool have been automated properly.

# REFERENCES

1. N. Mandal and S. Jadhav, "A survey on network security tools for open source", *2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, 2016. Available: 10.1109/icctac.2016.7567330 [Accessed 16 February 2021].

1. Nmap.org, 'Obtaining, Compiling, Installing, and Removing Nmap ', 2014. [Online]. Available: https://nmap.org/book/inst-linux.html#inst- rpm.
2. A. Cecil, A Summary of Network Traffic Monitoring and Analysis Techniques. [Online]. Available: https://www.cse.wustl.edu/~jain/cse567-06/ftp/net\_monitoring/index.html. [Accessed: 24-Mar-2021].