Analysing Wireless Sensor Network Trafficking and Ways to Minimise It

To

Vellore Institute of Technology

SENSE

by

Sanhita Kundu

20BEC0215

sanhita.kundu2020@vitstudent.ac.in

Sabyasachi Mohanty

20BEC0401

sabyasachi.mohanty2020@vitstudent.ac.in

Debabrath Sarkar

20BEC0430

debabrath.sarkar2020@vitstudent.ac.in

Vaishnavi Kolichala

20BEC0469

vaishnavi.kolichala2020@vitstudent.ac.in

Faculty In Charge:

Sriharipriya K C

sriharipriya.kc@vit.ac.in

Abstract— This report presents an analysis of wireless sensor network (WSN) trafficking, which is a critical aspect of WSN security. The study focuses on the traffic patterns and characteristics of WSNs, comprising the categories of traffic, the amount of traffic, the origins, and destinations of traffic. The report also examines the security implications of WSN trafficking, including the potential for attacks and the effectiveness of existing security measures. The analysis is based on a comprehensive literature review and a series of experiments conducted on a WSN testbed like wireshark to get the packet sniffing and then using JWT token authorization to encrypt the network layer and microsoft azure to create virtual network. The results of the study provide insights into the nature of WSN traffic and the challenges of securing WSNs against attacks. The report concludes with recommendations for improving WSN security, including the development of new security protocols and the use of dev ops techniques to detect and prevent attacks.

INTRODUCTION:

Packet sniffing is a method that gathers packets passing via a network on which it is placed. A packet sniffer is a monitoring tool for all network traffic. Moreover, it can collect and record both incoming and outgoing network data. Packets are the format of information sent through a network. In a network, for example, when a packet is transferred from one computer to another, it is divided into smaller segments with the source and destination addresses and other vital data attached. But, network performance may be analysed and network bottlenecks can be identified by installing a packet sniffer on any network node (source or destination). Packet sniffers are utilised by network administrators primarily because they assist with network troubleshooting, network intrusion detection system to monitor attackers, locating bottlenecks in networks, and converting binary network data to human readable form, such as collecting clear

usernames and passwords, VoIP communications, mapping the network, etc. These are prohibited uses of a packet sniffer unless your organization's network administrators provide permission. In addition to network analysis and protocol analysis, packet sniffing is also referred to as sniffing. There are both active and passive packet sniffers. Passive packet sniffers are undetectable since they do not respond; they just collect data. Passive sniffers have uses in industries outside of telecommunications.

LITERATURE SURVEY:

The network interface card (NIC) is subsequently configured for promiscuous mode. According to the dictionary, the definition of promiscuous mode is a lack of discrimination. All packets traversing a network reach the network interface card (NIC) of each node, which then verifies the IP addresses of the source and destination nodes. Therefore, when promiscuous mode is enabled, the network interface card (NIC) accepts all packets regardless of their destination address. Protocol interpretation is the retrieval of data for protocols such as TCP, IP, UDP, and ICMP, among others. [1]

Look for machines operating in a NIC mode that enables all traffic to pass without verifying the destination address. Isolated sniffers are difficult to detect since they do not give off any telltale signs by broadcasting their data flow. The use of a reverse DNS query may reveal multi-function sniffers. Promiscuous mode may be detected using any number of tools, the most popular of which being Nmap. Check the router's MAC address and other computers to see if it has changed with the help of an intrusion detection system. Network sniffing may be uncovered by an IDS. When malicious actions like sniffer or MAC spoofing are detected, the administrator is notified.[2]

Numerous studies have been conducted on packet sniffing for LAN or WAN monitoring ; numerous monitoring tools are available. In this paper, the behaviour of various tools is examined Free and open-source

packet analyser is Wireshark. Wireshark is utilised for network troubleshooting and analysis, but it does not offer intrusion detection and requires more memory for installation. Tcpdump is a common packet analyzer that can be run from the CLI. It records and displays TCP/IP and other network packets in real time as they are sent and received. Several additional tools are analysed; they exhibit a variety of problems, including memory and functionality issues . Therefore, we must design a tool that addresses the aforementioned issues and occupies minimal space.[3]

The primary objective of the thesis is to evaluate any network for enhanced performance and security. This necessitates less use of system resources such as memory and processor, and less packet loss in comparison to other systems. This section contains a number of tests conducted on network captured data, based on a number of parameters. The outcome is analysed and compared to other outcomes. In result analysis, data play a central role. Network data collected in real time. Different sites' packets are captured from the live environment. Data collected for analysis are referred to as datasets.[4]

Data is delivered by a computer over a network in the form of packets. These packets, which are really just data chunks, are meant for a certain system. Every message that is sent has a destination in mind. As a result, a particular computer has rapid access to all the data. Even if a system in a network is frequently designed to receive packets and only read the data that is intended for it, the sniffing process still needs both hardware and software to work properly. The procedure can be broken down into three phases. A packet sniffer is used to collect the wire's raw binary packets. Switched networks typically do these three tasks when a promiscuous network interface is selected: conversion of the collected binary data into readable form; analysis of the combined data. Using the network data it has gathered, the packet sniffer first checks its protocol using that data before beginning to analyse the specific components of that protocol.[5]

The majority of end users are not familiar with the fundamental security ideas behind SSL and encryption. Similar to this, a large portion of the

data offered by a conventional sniffer software is essentially meaningless. The seven layers provided by conventional tools can be used to scan a simple cookie or password in network traffic. It is challenging when collectors are spending a lot of time researching hidden price values, filter. As a result, network and security professionals may find packet sniffers to be a beneficial tool for both problem-solving and sniffing out networks where we know our children or children's children are present.[6]

Hacking is not the only useful application for packet sniffers; network traffic analysis, packet/traffic monitoring, and troubleshooting are a few others. Because a packet may contain sensitive information like clear-text passwords or user identities, the packet sniffer was designed to capture packets. Networks that are switched and un-switched can both be sniffed. We can utilise a few tools that are also used by researchers to record network traffic. We can assume that packet sniffers can be used for intrusion detection. [7]

PROPOSED METHODOLOGY:

To encrypt network layer using JSON Web Tokens (JWT) in a network communication system, you can follow these steps:

- Identify the data that needs to be transmitted securely over the network layer, such as user credentials or sensitive information.
- Generate a JWT token with a secure secret key using a cryptographic algorithm such as HMAC-SHA256 or RSA. The token should contain the data to be transmitted, as well as any additional information such as the issuer, audience, and expiration time.
- Attach the JWT token to the network layer data, typically as a header or a query parameter, depending on the network protocol being used.
- Send the encrypted data and the JWT token over the network layer to the intended recipient.
- Once the data has been decrypted, it can be processed and used as needed by the recipient application.

Microsoft Azure provides several options for encrypting the network layer of your applications and services. Here are some steps you can take to use Azure to encrypt your network layer:

1. Choose a Virtual Network (VNet): create a VNet in Azure, which is a logical isolation of your network in the cloud. You can choose the region, address space, and subnet range for your VNet.
2. Create a Network Security Group (NSG): An NSG is a set of security rules that you can apply to a subnet or network interface. You can use NSGs to allow or deny traffic to and from your resources.
3. Use Azure Firewall: Azure Firewall is a managed firewall service that you can use to filter network traffic. You can use it to create application and network rules to control traffic flow.
4. Use Virtual Private Network (VPN): You can use VPN to create a secure connection between your on-premises network and your Azure VNet. Azure provides several VPN options, including Azure VPN Gateway, Site-to-Site VPN, and Point-to-Site VPN.

By following these steps, you can ensure that the data transmitted over the network layer is encrypted and can only be accessed by authorized parties with the correct secret key and verification process.

BLOCK DIAGRAM:

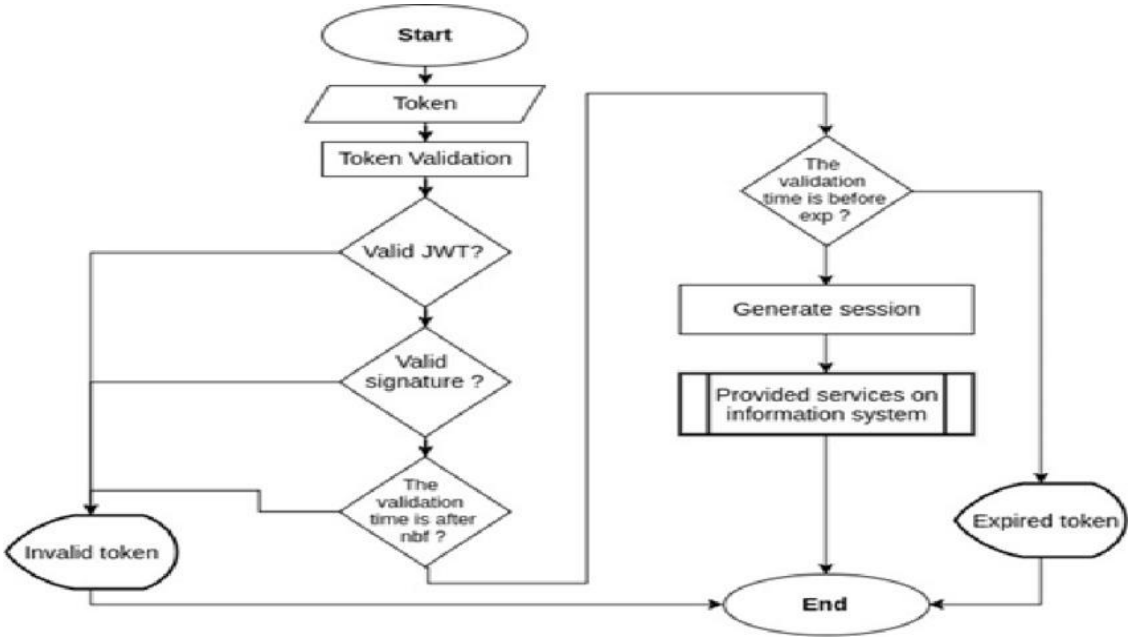

Fig. 1(a) Block diagram for proposed model

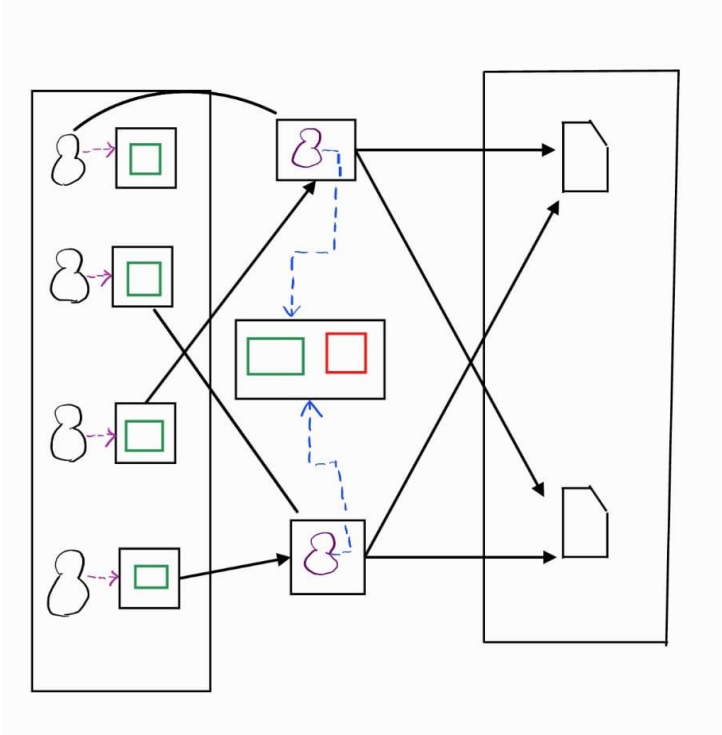

Fig. 1(b) Proposed Model Architecture

RESULTS:

Microsoft Azure provides several options for encrypting the network layer of applications and services. These include creating a Virtual Network (VNet), creating a Network Security Group (NSG), using Azure DDoS Protection, enabling Transport Layer Security (TLS), using Virtual Private Network (VPN), and using Azure ExpressRoute. By implementing these steps, Microsoft Azure can provide a secure and reliable environment for your business.

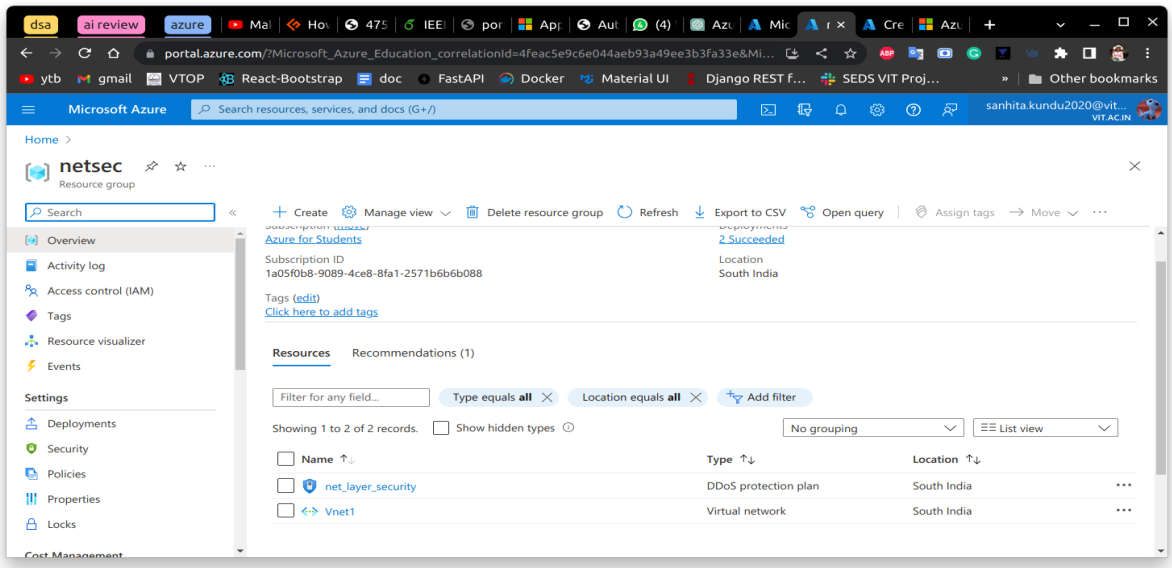

Fig. 2

Virtual Network with Microsoft Azure

Virtual Network has been created using Microsoft Azure

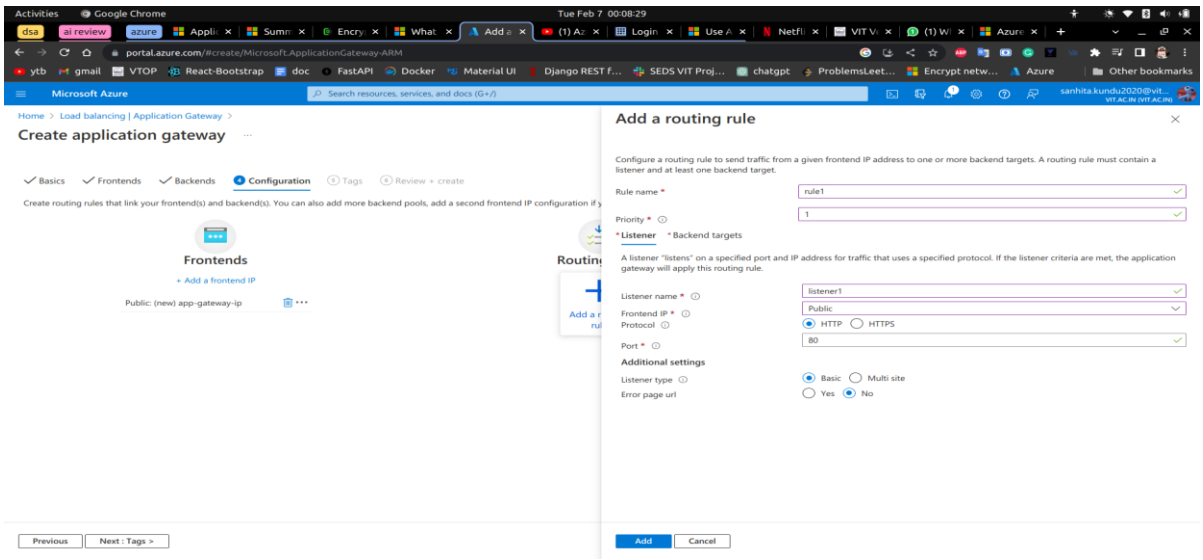

Fig.3 Routing

Routing for the above network shown in figure 2 has been done using Microsoft azure

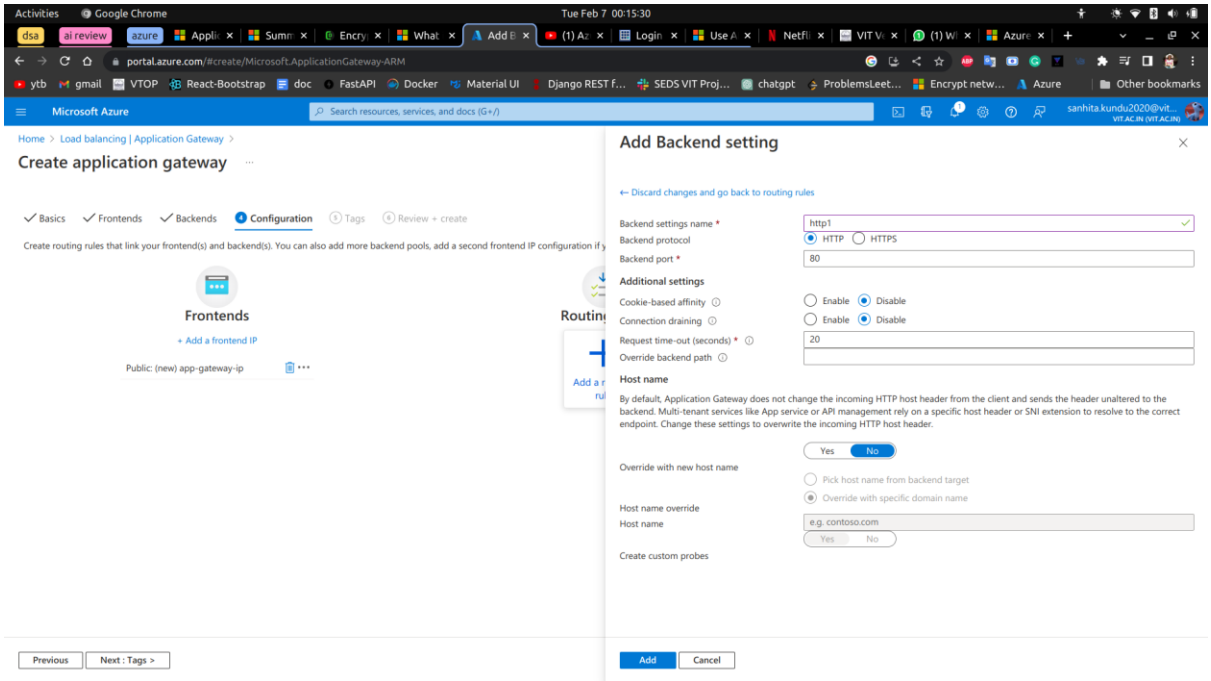

Fig. 4 Backend Pulling

This figure shows that backend Pulling for the virtual network shown in Fig. 2

Before Encryption:

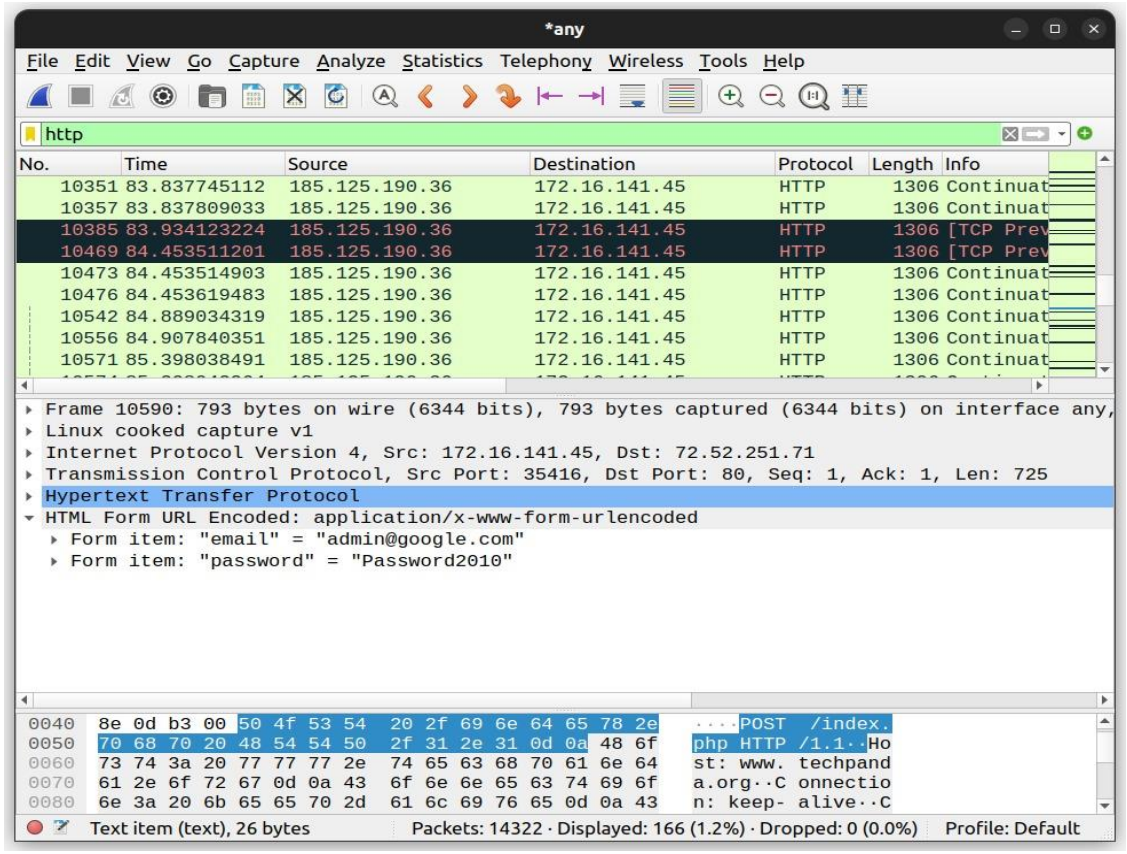

Fig. 5 Wireshark Terminal before encryption

Before we apply data encryption, we can see that our data is being exposed to packet sniffing and that our username and password are visible and our data is extremely vulnerable.

After Encryption:

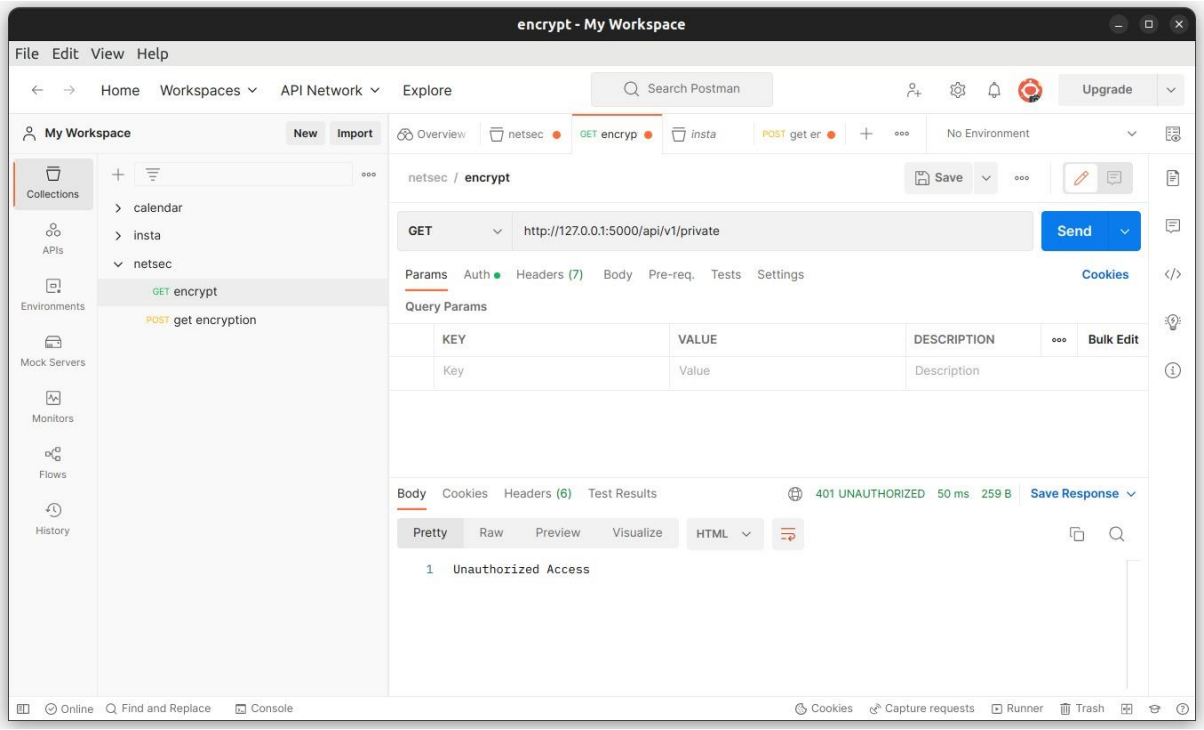

Fig. 6 Wireshark Terminal after encryption

After encryption it can be seen that the username and the password are getting masked and it shows a message that says “unauthorized” thus no one can hack or extract our data.

JSON Web Tokens (JWT) offer several advantages over using plain HTTP for transmitting data and authentication. Some of these benefits include:

- Stateless design: JWT tokens are designed to be stateless, which means that they do not require the server to maintain session information for each user. This makes them easier to scale and reduces the load on the server.
- Reduced server load: Because JWT tokens are self-contained and include all the necessary information for authentication and authorization, they reduce the load on the server. This is because the server does not need to query a database or perform additional calculations to verify the user's identity.
- Improved security: JWT tokens use a digital signature to verify the authenticity of the data, which makes them more secure than

plain HTTP. This reduces the risk of data tampering and makes it more difficult for attackers to intercept or modify the data in transit.

- Cross-platform compatibility: JWT tokens are compatible with any programming language or framework that supports JSON. This makes them a good choice for building web and mobile applications that need to communicate securely with APIs.
- User identity verification: JWT tokens include information about the user's identity, which allows the server to verify the user's identity and determine what resources they are authorized to access.

Overall, using JWT tokens for encryption and authentication provides a more secure and scalable way to transmit data and authenticate users than plain HTTP. They offer several advantages over plain HTTP, including improved security, reduced server load, and cross-platform compatibility.

CONCLUSION:

In conclusion, Azure Virtual Network (VNet) provides several built-in features to encrypt network traffic, such as Transport Layer Security (TLS), Secure Sockets Layer (SSL), and Virtual Network encryption. You can enable these features through the Azure portal or APIs, depending on your specific requirements.

In addition to encryption, you can also use JSON Web Tokens (JWT) to secure your network traffic by providing secure identity and access management. Azure Active Directory (Azure AD) provides a centralized identity management solution that you can use to issue and verify JWT tokens for your applications and services.

By combining encryption and JWT tokens, you can secure your network layer in Azure and protect your applications and data from unauthorized access and attacks. However, it's crucial to keep in mind

that security and encryption are ongoing processes that call for constant oversight and modifications to maintain their efficacy over time.

REFERENCES:

[1] Dr. Charu Gandhi, Gaurav Suri, Rishi P. Golyan, Pupul Saxena, Bhavya K. Saxena “Packet Sniffer – A Comparative Study,”

[2] Ruchi Tuli, Packet Sniffing and Sniffing Detection.

[3] Pallavi Asrodia, Mr. Vishal Sharma, “Analysis of Various Packet Sniffing Tools for Network Monitoring and Analysis,”.

[4] Hemlata Patel, “Network Monitoring and Analysis by Packet Sniffing Method ,”.

[5] Dr.Aruna Varanasi, P.Swathi, ”Comparative Study of Packet Sniffing tools for HTTP Network Monitoring and Analyzing”

[6] Shivam Kumar, Suryansh Jigyasu, Vikas Singh, ”Network Traffic Monitor and Analysis Using Packet Sniffer”

[7] Palak Girdhar, Vikas Malik, ”A Study on Detecting Packet Using Sniffing Method”

[8] Ruwaidah Fadhil Albadri, ”Development of a network packet sniffing tool for internet protocol generations”

[9] Otisule Oluwabukola, Awodele Oludele, A.C. Ogbonna, Ajeagbu Chigozirim, Anyeahie Amarachi, ”A Packet Sniffer (PSniffer) Application for Network Security in Java”

[10] Jiaqian Li; Chengrong Wu; Jiawei Ye; Jiong Ding; Qinwei Fu; Jiatao Huang, The Comparison and Verification of Some Efficient Packet Capture and Processing Technologies

[11] Michael J. Jipping, Agata Bugaj,Liliyana Mihalkova, Donald E. Porter,” Using Java to teach networking concepts with a programmable network sniffer”

[12] Muhammad Syaffiq Abdul Malek, Ahmad Roshidi Amran, ”A Study of Packet Sniffing as an Imperative Security Solution in Cybersecurity”

[13] Sourabh Saroha, ”Restraining Packet Sniffing Security: A Brief Overview”

[14] S. H. Brahmanand, N. Dayanand Lal, D. S. Sahana, G. S. Nijguna Parikshith Nayak ,”A Systematic Approach of Analysing Network Traffic Using Packet Sniffing with Scapy Framework”

[15] Usha Banerjee, Ashutosh Vashishtha, Mukul Saxena, ”Evaluation of the Capabilities of WireShark as a tool for Intrusion Detection”.