IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

HIGH LEVEL SECURITY FOR INTERNET USAGE BY USING Q-VPN

Sayooj U[1], Vishal S[2], Rinsha[3], Priyana Rijesh[4], Jibin Joy[5]

[1] B.Sc. CS Student, [2] B.Sc. CS Student, [3] B.Sc. CS Student, [4] B.Sc. CS Student, [5]Assistant Professor
Department of Computer Science,
Yuvakshetra Institute of Management Studies, Palakkad, Kerala, INDIA

Abstract: The internet is not a secure platform. Third parties may track us easily and monitor us. At this point, a VPN is needed. The VPN helps to make data more private and secure by routing the traffic through encrypted tunnels. Rather than other VPNs, the facilities of Q-VPN are network level protection. The add-ons and malicious sites are automatically blocked. It is an open-source VPN that follows an LGPL license. Protect the user from network-level hijacking. Using WireGuard Protocol for tunnelling, Provides a TOR facility.

IndexTerms **– Virtual Private Network(VPN), Amazon Web Services (AWS)**

I.INTRODUCTION

Q-VPN is a Virtual Private Network that uses WireGuard protocol to tunnel the packets between the server and the client. It is intended to generate a user-friendly VPN without interrupting their browsing with no compromise to protecting the data and Privacy. For the users that need more security, "CAPTURE "provides the TOR along with the WireGuard. The TOR protocol helps the user to be more anonymous. With this VPN, users get a high bandwidth internet connection.

VPNs encrypt Internet traffic and disguise the online identity of the users. This makes it more challenging for outsiders or hackers to monitor internet activities and steal personal data. The majority of VPNs now use a "Pay for the service" model, and data privacy is not guaranteed. This application provides network-level protection, which means giving security to the entire network; it is free to use, and it can modify some areas of the application. The user can feel a secure atmosphere in this application because it provides a "state of the art "application by incorporating both VPN and the facility to use TOR.

II. OBJECTIVES

Q-VPN is a Virtual Private Network capable of utilizing WireGuard protocol..It aims to secure the traffic of a user over the internet.It provides network-level protection.Apart from both VPN and TOR, we also provide the facility and the built-in feature of this VPN to block all Ads and Malicious sites reported to the PI Hole community.

III. FEASIBILITY STUDY

During the system analysis, a feasibility study of the proposed system is carried out. The objective of this feasibility study is to test the developing system's technical, economic and social feasibility. This is done before developing the plan and is done by investigating the existing system in the area under investigation and generating ideas about the system. There are three aspects in the feasibility study portion of the preliminary investigation.

The system must be evaluated from the technical viewpoint first. The assessment of this feasibility must be based on an outline of the system requirement in terms of input, output, programs and procedure. Having identified the outline of the system, the investigation must go on to suggest the type of equipment required method of developing the system, and the investigation must go on to suggest the type of equipment,

required method of developing the design and the method of running the system. The system has been developed using Python. And also, the other co-factors for developing the proposed approach are readily available so that the project is technically feasible for development.

The developing system must be justified by cost and benefit. Since the design was developed as part of the project work, there is no manual cost to spend for the proposed system. All software used in developing the proposed approach is free to use. The server used to host the proposed method is AWS (Amazon Web Service server). The AWS server provides 12 months of free trial and 750 hours of CPU time. So it is enough for Project time. And also, all the resources are already available. It is indicated that the system is economically possible for development.Suppose for a moment that technical and economic resources are both judged adequate. The systems analyst must still consider the operational feasibility of the requested project. Operational feasibility depends on human resources available for the project and involves projecting whether the system will operate and be used once installed. The internet is not a secure platform, so a VPN is more needed. So CAPTURE provides more security and features than the other VPNs, making it a multiple application for users.

IV.IMPLEMENTATION

FUNCTIONAL REQUIREMENTS

Open the Connection

When the user opens the connection, the following operations are performed:

- **Connect with wireguard**

As soon as the user clicks the connect button, the VPN first checks if the device's network connection is available. If it is available, the Q-VPN sends an activate command to establish the relationship with the client and AWS server with the help of the configuration file that is pre-installed with the installation of the CAPTURE. When the wire guard server receives the command, it checks the public key of the user, and if it matches, the wireguard tunnelling is started, and the user can establish a handshake. Otherwise, the protocol is not activated.

For checking this, it only takes 2 to 3 seconds. In the AWS server, we use the default port 51820. If the user's network connection is not available, it displays an error message to the user that "Check your Internet Connectivity". After the successful establishment of the connection, the connect button changes its image to a new image so the user can easily identify that the connection is established. Also, with this, there are two functions that will be invoked.

Firstly, when the wireguard is activated the user the system provides network-level protection for the users; that is, all the advertisements, malicious websites etc, are blocked. This is achieved with the help of the PI-HOLE community. The CAPTURE sets the PI-HOLE as its DNS. Normally the traffic will be transferred to Google or Cloudflare(1.1.1.1). But when it uses PI-HOLE as its DNS, the traffic from the VPN transfers to the PI-HOLE DNS. Then the PI-HOLE DNS transmit the request to GOOGLE / Cloudflare (1.1.1.1). When it returns the requested data, the PI-HOLE filters the data and avoids other contents such as advertisements, malicious sites etc The filtering is done using the list provided by the Community members, which is created by research and testing. And returns the filtered data.

Suppose if a user loads a website, the ads on that website are also loaded with that. But in case the user connected with the CAPTURE, the system transfers the request to the Pi-Hole, and here the Pi-Hole cross-checks the data on the page with the list, and if any malicious or unwanted contents are present on the page, it will be blocked, and the page will be available to the user.

Secondly, the IP address checks whenever the user starts the CAPTURE application. And also, the user connects to the wireguard. The IP address fetched directly from the website ipecho.net in the text format, and it will be displayed within a few seconds normally. It may change according to the network availability of the user. From ipecho.net, the system will fetch the IPV4 address of the user.

Close the connection

When the user wants to close the connection, the system will send a Command to close the connection when the user clicks the disconnect button. The Server disconnects the relationship with that client. And also, the image of the button will change to the previous state. And also, the IP address is displayed.

- **Duration of the connection**

The system will display the connection duration of the user in hh:mm:ss format when the user closes the connection with the server. The Q-VPN programme minimises when the user presses the TOR button, and Firefox is opened with the ability to transmit tunnels through the TOR protocol. To be more anonymous, use the TOR protocol. In order to hide a user's location and usage from anyone performing network surveillance or traffic analysis, it routes Internet traffic over a free, global volunteer overlay network made up of more than 6,000 relays. It is more challenging to link a user's online behaviour to them when Tor is used. By preventing surveillance of its users' online activity, Tor aims to protect their privacy and freedom and capacity for secret communication. On the application layer of the communication protocol stack, it is layered like the layers of an onion; encryption is used to implement onion routing. Applications that support SOCKS can be set up to route network traffic over a Tor instance's SOCKS interface, which is accessible at localhost on TCP ports 9050 (for standalone Tor) or 9150 (for the Tor Browser bundle). To multiplex and

onion-route that traffic, Tor periodically builds virtual circuits through the Tor network. Once within a Tor network, traffic is sent around a course from router to router until it eventually reaches an exit node. At this point, the cleartext packet is made available and is relayed to its intended destination. The traffic appears to begin at the Tor exit node when seen from the final location.

Since it operates at the Transmission Control Protocol (TCP) stream level, Tor differs from most other anonymity networks in that it is application independent. Internet Relay Chat (IRC), instant messaging, and web browsing are examples of applications whose traffic is frequently anonymized using Tor.

Websites and other servers that use Tor can also maintain their anonymity. Onion services are servers set to only accept connections from Tor (formerly hidden services). An onion service is accessible using its onion address, typically through the Tor Browser, instead of disclosing a server's IP address (and hence its network location). The Tor network recognises these addresses by consulting a distributed hash table inside the web to find the appropriate public keys and introduction points. While maintaining both parties' anonymity, it can route data to and from onion services, even those housed behind firewalls or network address translators (NAT). To access these onion services, one needs Tor.

The following CAPTURE function is SpeedTest. The speed test module in Python is used to retrieve the download speed when a user clicks the speed test button. Likewise, show it in mb/s. The upload speed is then retrieved and presented in mb/s. The about page is provided in the system. On this page, the user can view the details about the developers, and this page provides the link to the GitHub repository for viewing the code.

V.NON-FUNCTIONAL REQUIREMENTS

The non-functional requirement involves those functions that are performed by the system independent of the user. It deals with the system's characteristics that cannot be expressed functionally.

- **Performance Requirement**

  - The connection can be easily established using the suggested technology we are developing.
  - The system should be easy to handle.
  - System should give expected performance results.
  - The response time should be short.

- **Security Requirement**

The suggested system will be mainly utilised for secure internet traffic. Additionally, the system offers network-level protection against nefarious websites and adverts for the user, giving them a higher level of security.

- **Maintainability Requirement**

The proposed system is designed to make it easy to maintain and release updates. The future expansion of the system, like adding more servers to the proposed system, can be easily implemented without interrupting the users, and also, the other feature updates can be easily implemented.

VI.PROTOCOL

WireGuard, a communication protocol and open-source, free software that creates encrypted virtual private networks, strives to be easy to use, quick, and low attack surface (VPNs). Compared to IPsec and OpenVPN, it aspires to perform better while using less energy. The WireGuard protocol uses UDP to transmit data. The Linux version of the programme reached a stable production release in March 2020, and some Linux distributions backported it to previous Linux kernels and the Linux 5.6 kernel. The Linux kernel's essential pieces are subject to the GNU General Public License (GPL) version 2, whereas alternative implementations are subject to GPLv2 or other free/open-source licenses.

To make it distributable, all of Q-VPN code is converted to binary files using PyInstaller:- A python module that packs the python libraries along with the binary files to simplify the installation process. PyInstaller finds all the required Python dependencies of Q-VPN and fills them with the binary files. After this, Q-VPN can be executed using the executable binary file without relying on a global Python Interpreter. This also makes sure that the Q-VPN works the same way across all of the end-users machines. To further simplify the process of distribution, the binary files are packed into an installer package using the Inno installer.The installer package automates the process of implementation and deployment in end-users computers. It copies the necessary files and makes sure that all the dependencies are satisfied. CAPTURE can be either launched using the installed shortcuts or from the actual location of the installed files located in the Program Files directory under Windows x64.

VII.FUTURE ENHANCEMENT

The Q VPN has developed with the prime objective of ensuring security over the internet with no compromise in users' data and privacy. In this Q-VPN, it achieved the major goal of providing protection to every user. Q-VPN is fully functional and performs optimally during its lifetime. The system has been designed in such a way that it can be modified with minimal effort. Since Q-VPN is an open-source project that comes under the LGPL license, its code can be reviewed by anyone, and maintenance contributions can be made through the source control system, keeping the software up to date. Below are the Future Expansions that are planned after the completion of Q-VPN.

1. Support for Android Devices and iOS.

Provide support for the users that are in Android and iOS.

2. Login for Customized details.

The users can connect to the VPN after downloading the app. If a user wants to connect the Q-VPN from their own multiple devices from the same account, Q-VPN will be able to provide a login forum for that type of user. For the login users, the system will give an individual configuration file for connections.

3. Multiple Servers.

Provide the users that have the accessibility to select which servers they want to connect to the network. The servers will be in different countries. The main advantage is that the users can choose their servers in any country without paying any amount. All of them are free for every user.

VIII.CONCLUSION

Q-VPN is developed to avoid the shortcomings of other VPNs. The Q-VPN comes under the license of LGPL. It is most helpful for others to modify the system for their own needs. I believe that the Q-VPN is user-friendly, easy to use and protects the user'suser's privacy with no compromise. I achieved all the features that were planned during the development stage. They are, provide Network level protection for the users. Free to use, and it is an Open-Source Software. The ad-ons and malicious Sites are automatically blocked. It protects the user from Network level Hi-jacking. Tracking Domains are Blocked. As the "Q-VPN "uses WireGuard protocol, the connection is much more stable and offers a high bandwidth connection. It takes less time to establish a relationship and close the connection.

IX.REFERENCES

1. Baek, Seung-Jin and Jeong, Moon-sang and Park, Jong-Tae.(1999). 'Policy-based Hybrid Management Architecture for IP-based VPN', [online] KNOM Review, 2(2) pp. 22-30,
2. Bansode, Rama and Girdhar, Dr. Anup. (2017). 'IPV6 Security Considerations', Cyber Times International Journal of Technology and Management, [online] New Delhi, India. 10(1), pp. 22-26.
3. Gokulakrishnan, Jayanthi and Bai, Dr. V. Thulasi. (2014). 'A Survey Report on VPN Security & its Technologies'. Indian Journal of Computer Science and Engineering (IJCSE). ISSN: 0976-5166, 5
4. Rahimi, Sanaz and Zargham, Mehdi. (2011). 'Security Analysis of VPN Configurations in Industrial Control Environments', [online] Chapter6, pp. 73-88,
5. Singh, Kuwar Kuldeep Veer Vikram and Gupta, Himanshu. (2016). 'A NEW APPROACH FOR THE SECURITYOF VPN'. [online] ACM, ISBN 978-1-4503-3962, Udaipur India
6. Yamansavascilar, B.; Guvensan, M.A.; Yavuz, A.G.; Karsligil, M.E. Application identification via network traffic classification. In Proceedings of the International Conference on Computing, Networking and Communications (ICNC), Silicon Valley, CA, USA, 26–29 January 2017. [**Google Scholar**]
7. Aceto, G.; Ciuonzo, D.; Montieri, A.; Pescape, A. Mobile Encrypted Traffic Classification Using Deep Learning: Experimental Evaluation, Lessons Learned, and Challenges. *IEEE Trans. Netw. Serv. Manag.* **2019**, 445–458. [**Google Scholar**] [**CrossRef**]
8. Vu, L.; Thuy, H.V.; Nguyen, Q.U.; Ngoc, T.N.; Dutkiewicz, E. Time Series Analysis for Encrypted Traffic Classification: A Deep Learning Approach. In Proceedings of the 2018 18th International Symposium on Communications and Information Technologies (ISCIT), Bangkok, Thailand, 26–29 September 2018. [**Google Scholar**]
9. Fan, Z.; He, W.; Xue, L.; Bridges, P.G. Inferring users' online activities through traffic analysis. In Proceedings of the Fourth ACM Conference on Wireless Network Security, WISEC 2011, Hamburg, Germany, 14–17 June 2011. [**Google Scholar**]
10. Platt, J. A Resource-Allocating Network for Function Interpolarion. *Neural Comput.* **1991**, *3*, 213–225. [**Google Scholar**] [**CrossRef**] [**PubMed**]
11. Wei, W.; Ming, Z.; Zeng, X.; Ye, X.; Sheng, Y. Malware traffic classification using convolutional neural network for representation learning. In Proceedings of the 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 11–13 January 2017. [**Google Scholar**]

12. Adrian, D.; Bhargavan, K.; Durumeric, Z.; Gaudry, P.; Zimmermann, P. Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. *Commun. ACM* **2015**. [**Google Scholar**] [**CrossRef**]
13. Aqniu. 2020. Available online: **Https://www.aqniu.com/industry/69880.html/** (accessed on 3 September 2020).
14. Zhao, Y.; Yang, Y.; Niu, Y.; Wu, K.; Zhao, Q. A Classification and Identification Technology of TLS Encrypted Traffic Applications. In Proceedings of the 2021 IEEE 4th International Conference on Big Data and Artificial Intelligence (BDAI), Qingdao, China, 2–4 July 2021. [**Google Scholar**]
15. Bujlow, T.; Carela-Espanol, V.; Barlet-Ros, P. Independent comparison of popular DPI tools for traffic classification. *Comput. Netw.* **2015**, *76*, 75–89. [**Google Scholar**] [**CrossRef**]
16. Mcpherson, J.A.; Ma, K.-L.; Krystosk, P.; Bartoletti, T.; Christensen, M. PortVis: A tool for port-based detection of security events. In Proceedings of the Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC 2004), Washington, DC, USA, 29 October 2004. [**Google Scholar**]
17. Amann, J.; Sommer, R. *Exploring Tor's Activity through Long-Term Passive TLS Traffic Measurement*; Springer: Cham, Germany, 2016. [**Google Scholar**]
18. Wei, W.; Ming, Z.; Wang, J.; Zeng, X.; Yang, Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July 2017. [**Google Scholar**]