Journal of Computing Research and Innovation (JCRINN) Vol. 6 No. 4 (2021) (pp90-101)
https://jcrinn.com : eISSN: 2600-8793

SafeSearch: Obfuscated VPN Server using Raspberry Pi for Secure Network

Mohd Faris Mohd Fuzi[1]*, Mohamad Ridzuan Mohd Alias[2], Naginder Kaur[3], Iman Hazwam Abd Halim[4]
[1,2,] *Faculty of Computer and Mathematical Sciences,*
Universiti Teknologi MARA Perlis Branch, Arau Campus, 02600 Arau, Perlis, Malaysia
[3]*Academy of Language Studies,*
Universiti Teknologi MARA Perlis Branch, Arau Campus, 02600 Arau, Perlis, Malaysia

*Corresponding author: *farisfuzi@uitm.edu.my*
Received Date: 15 August 2021
Accepted Date: 9 September 2021
Published Date: 20 September 2021

HIGHLIGHTS

- Obfuscated VPN Server was developed in Raspberry Pi 3 using OpenVPN protocol and obfuscated technique.
- Used open-source software and low-cost VPN Server implementation to secure the network
- VPN Traffic successfully disguised to bypass web filtering and deep packet inspection
- Network Assessment Penetration Testing proves that the SafeSearch VPN Server is able to improve network security levels

ABSTRACT

Virtual Private Network (VPN) is a private network that uses public network to tunnel the connection from the users' end to the VPN server. VPN allows users to create a secure connection to another network over the public Internet. VPNs can be used to shield users' browsing activity and encrypts data transmitted over the network to prevent sniffing attack. Nowadays, users can either pay a premium price for a good VPN service or risk their privacy using free browser-based VPN. Thus, SafeSearch is developed to address these issues in mind. With SafeSearch, users will not need to fork out a lot of money for premium VPN subscription services or expose themselves to targeted advertising when utilising free browser-based VPN. In this study, open VPN protocol was used to create the VPN server on a microcomputer called Raspberry Pi. The software used was mostly open-source except for the VPN client. Obfuscation technique was used to hide VPN traffic by disguising it as just another normal Internet traffic against Deep Packet Inspection when passing through firewall. After the VPN server was established, tests were carried out to evaluate the functionality and reliability of the VPN server in "real-world" environment. The tests conducted were network restriction penetration assessment, network performance and user acceptance test. Penetration assessment result showed that SafeSearch is capable of bypassing web filtering and deep packet inspection. Network performance during SafeSearch connection has slight latency and bandwidth decline, although it is not overly affected. The outcome of the user acceptance test was positive as the majority of participants of the study were confident that SafeSearch can secure their connection and protect their privacy when browsing the web. To conclude, both objectives of this project were fully achieved and the scope of study was followed thoroughly.

Keywords: *Virtual Private Network, Obfuscation VPN, OpenVPN, Raspberry Pi, Microcomputer, Security*

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA (https://creativecommons.org/licenses/by-nc-sa/4.0//)

INTRODUCTION

Nowadays, public wireless network (Wi-Fi) is rapidly increasing due to the demand of free wireless services. This free Wi-Fi service is available in most public places such as restaurants, hotels, cafes, and airports. However, free Wi-Fi services normally implement lower security measures which use weak encryption and authentication that produce increased risk of attacks for intercepting data (Niitsu, Sakai, & Tezuka, 2019). Although 68.02% of Wi-Fi hotspots use WPA2 which is better in terms of security, they are still vulnerable to brute-force and dictionary attacks. Recently, WPA2 encryption was found to have a severe flaw that allows attackers perform Key Reinstallation Attack (KRACK) to break the encryption (Vanhoef & Piessens, 2017).

When users are connected to the public Wi-Fi, it is difficult to validate correctness of access points and they may be connected to spoofed access points (Niitsu et al., 2019). This can occur if an Evil Twin AP is maliciously placed. As a result, the attacker(s) can do various Man-in-the-Middle (MITM) attacks including eavesdropping, falsification of contents, presenting fake log-on pages for identity theft purposes, and carrying out active attacks on users' devices (Goto, 2019). With the availability and ability of wireless attacking tools such as network scanners, attackers care able to find the most vulnerable wireless networks and plan an attack that can affect a large number of users (Akram, Saeed, & Daud, 2018). One of the popular network scanners, Vistumbler has the functionality to detect and extract information from access points such as SSID, authentication mode, encryption, network type, router manufacturer, and signal strength (Goncharov, Zareshin, Bulychev, & Silnov, 2018). This information will help attackers to launch the attack.

In effort to solve this problem, VPN is used when browsing on public Wi-Fi. The essence of VPN is to build a secure tunnel in the public network using relevant encryption technology. Thus, the data transmission is secure and protected from being sniffed (Xu & Ni, 2020). VPN services such as NordVPN and TunnelBear are very secured but come with a monthly cost. On the other hand, there exist free VPN services, but users are unable to change the port. This is a serious problem because most workplaces will only allow port https (443) and http (80). Above all, these types of VPNs do not use obfuscation technology rendering them useless against Deep Packet Inspection when passing through firewall. Obfuscation technology is used to hide VPN traffic by disguising it as just another normal Internet traffic (Boldt, Kent, & Herpers, 2020). This paper implements VPN server named SafeSearch that encompasses a small Raspberry pi microcomputer installed with OpenVPN that uses home Internet setup. This will allow users access to their own VPN Server which will effectively eliminate steep monthly cost for a viable VPN service. By greenlighting this project, users will be able to connect to their VPN anywhere, anytime on any public Wi-Fi.

LITERATURE REVIEW

Virtual Private Network

VPN is a service which provides secure web access by privately routing your connection through a VPN server and hiding the client's online actions. The VPN software will encrypt the data, even before an Internet Service Provider (ISP) or the public Wi-Fi provider sees it. The data then goes to the VPN, and from the VPN server to the client's online destination. There are four core technologies in VPN which are tunnelling, encryption, identity authentication and key exchange management (Xu et al., 2020). VPN protocols define how the service handles data transmission over a VPN. The most common protocols are Point-To-Point Tunnelling Protocol (PPTP), Layer 2 Tunnelling Protocol (L2TP), Secure Socket Tunnelling Protocol (SSTP), Internet Key Exchange, Version 2 (IKEV2), and OpenVPN (Jaha, 2015). OpenVPN takes the best in the above protocols and does away with most of the flaws. It is based on

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA
(https://creativecommons.org/licenses/by-nc-sa/4.0//)

SSL/TLS protocol, and is an open-source project, which means that it is constantly being improved by hundreds of developers (Aung & Thein, 2020). It secures the connection by using keys that are known only by the two participating parties on either end of the transmission.

VPN Obfuscation Technique

Obfuscation is a technique used in 'Stealth VPN' which is simply a VPN server or protocol that can disguise VPN traffic as regular web traffic, even when subjected to deep packet inspection by the network administrator or firewall (Bodis, 2017). A stealth VPN is designed to be difficult to detect by firewalls and applications intended to block VPN traffic. VPN-blocking firewalls are common in countries that restrict or censor access to the Internet.

The way Stealth VPN works is it starts with a regular OpenVPN encrypted data. A typical OpenVPN data packet consists of two parts: 1) the header that contains packet identification and routing information, and 2) the payload which contains encrypted portion of the data packet, which will be forwarded by the VPN server to the correct web address (Aung et al., 2020). The stealth VPN then uses Obfuscation technique to remove all meta data from the packet header that identifies the data as belonging to a VPN protocol. Since the source packet has been obfuscated, the final step is for the stealth VPN to cloak it as regular HTTPS encrypted web traffic. The OpenVPN data packet is wrapped inside a second layer encryption, using SSL or TLS protocol which is the same encryption used by HTTPS. The data is then assigned to port 443, also known as HTTPS. With these two steps, the data packet is virtually indistinguishable from regular https data and is nearly impossible to block. In this research, Obsf64 has been chosen. It has the capability to disguise flow signatures by offering protection against some protocol such as deep packet inspection and fingerprint attacks (Boldt et al., 2020).

Related Works

Taib et al. (2020) implemented a VPN with Pi-Hole and Intrusion Prevention System (IPS) using Raspberry Pi to secure the network. The researchers named the project VPiSec where the system was developed using OpenVPN protocol and Pi-Hole application to block any known tracking Domain Name System (DNS) and advertising domain. This project also implemented Intrusion Prevention System (IPS) to prevent the brute force attack. The network performance had no significant difference while the users were connected to the VPN.

Taib et al. (2020) also developed an integrated tool that implemented OpenVPN protocol, DNS blocker and Intrusion Detection System (IDS) known as NetGuard. The researchers used Raspberry Pi to develop the system. OpenVPN was used as the protocol to provide the security and encryption for the network traffic while DNS blocker prevented unwanted advertisements. The researchers provided data that confirmed that the respondents were satisfied with the network performance and security during their connections using this system.

Pooja, Akansha, and Anurag (2018) developed their project Secure VPN Server deployed on Raspberry Pi. They conducted their research on public Wi-Fi security and concluded that users who used public Wi-Fi risk their privacy being intruded by unauthorised individuals. In accordance with the problem stated above, the researchers decided to implement a VPN Server into the Raspberry Pi. The project established the connection between VPN Server on Raspberry Pi and VPN client to provide multiple layers of protections. Once VPN session was established, the researchers implemented VPN authentication mechanism by incorporating three layers of verification. Finally, the project aimed at portability, is fully deployed on a

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA
(https://creativecommons.org/licenses/by-nc-sa/4.0//)

Raspberry Pi environment. This enables the system to become extremely portable, reusable and user friendly.

METHODOLOGY

This research methodology has several phases which include analysis requirements, design and implementation, testing, and result analysis. During Requirement Analysis, all required hardware and software for the project were determined. Table 1 and Table 2 show details of the hardware and software requirements.

Table 1: Hardware Requirements

No.	Item	Description
1	Raspberry Pi 3 B+	Acts as a VPN server. Used to implement all the networking features required for this project
2	CAT 6 Ethernet Cable	Provides a stable Internet connection to the Raspberry Pi
3	Micro SD card	Used as a storage device on the Raspberry Pi
4	Laptop	Used to configure routers and Raspberry Pi. Also used to connect to the VPN when it is operational
5	Router	Used for networking purposes such as DDNS and packet forwarding. Connects the VPN server to the world wide web.

Table 2: Software Requirements

No.	Item	Description
1	PuTTy	SSH client interfacing software
2	VNC Server and Client	Remote access software
3	OpenVPN	Open-source VPN protocol
4	Obfuscation Proxy	Disguises traffic to bypass firewall
5	Raspbian	The operating system used on the Raspberry Pi
7	Viscosity	An OpenVPN client for users to connect to their VPN server
8	Iperf	Network performance analysis tool

From the tables above, it can be seen that all the components were assembled to create a microcomputer server. In this project, Raspberry Pi 3 (RPi3) Model B+ was used because it can boast a quad core processor clocked at 1.4Ghz and with 1GB of RAM. This device is more than enough to handle OpenVPN and supports an ethernet port with speeds up to 300 Mbps which is vital for OpenVPN server operation. Furthermore, headless setup is easier on RPi3 compared to its predecessor which is the Raspberry Pi Zero

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA (https://creativecommons.org/licenses/by-nc-sa/4.0//)

due to the presence of an ethernet port. A headless interface was possible due to Raspbian pre-installed with a VNC server.

A router is needed because the VPN server must be remotely accessed through the Internet. Figure 1 displays the system architecture design for this project. Users can access the OpenVPN server by using Viscosity, OpenVPN client software. The OpenVPN client will create a tunnel through the network so that encrypted and obfuscated packet can pass through until it reaches the OpenVPN server network. The packet will then be port forwarded to the VPN server itself and reach the desired website on the Internet.

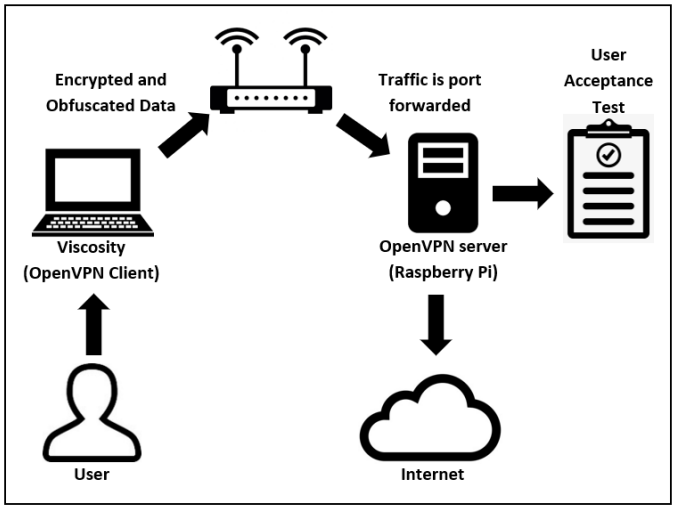

Figure 1: System Architecture

The process of installing OpenVPN in the Raspberry Pi 3 to create VPN Server started by using a shell script downloaded from the Linux repositories. After the installation was completed, an "ovpn" file was generated which contained all the necessary information that can be imported into a VPN client. Then, the VPN Server was successfully setup on our local network to bypass network restrictions on public networks.

Next, obfuscation proxy had to be installed in VPN Server. In this project, a software called Obfs4 was used. Obfs4 is a scrambling proxy that can disguise the user's Internet traffic and resembles a noise. Man-in-the-middle attacks were rendered inactive (useless), and it was not possible for perpetrators to spy on the user's online activity. Furthermore, content blocking or filtering could easily be bypassed because Firewall Deep Packet Inspection outright ignored noise signals during the connection using obfuscation proxy.

Figure 2 shows the step to install Obfs4 alongside Figure 3 and Figure 4 which shows the configuration for which the proxy will listen for new connections. After all the steps were accomplished, Figure 5 shows an example of the cert KEY.

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA
(https://creativecommons.org/licenses/by-nc-sa/4.0//)

```
apt-get update && apt-get install obfs4proxy
sudo mkdir -p /var/lib/tor/pt_state/obfs4
sudo nano /var/lib/tor/pt_state/obfs4/obfs4.config
*Configuration file* (listing 4.5)
sudo nano /etc/systemd/system/obfs4proxy.service
*SystemD service file* (listing 4.6)
sudo systemctl start obfs4proxysudo systemctl enable obfs4proxy
cat /var/lib/tor/pt_state/obfs4/obfs4_bridgeline.txt (listing 4.7)
sudo nano /var/lib/tor/pt_state/obfs4/obfs4_bridgeline.txt
```

Figure 2: Obfs4 Installation Steps

```
TOR_PT_MANAGED_TRANSPORT_VER=1
TOR_PT_STATE_LOCATION=/var/lib/tor/pt_state/obfs4
TOR_PT_SERVER_TRANSPORTS=obfs4
TOR_PT_SERVER_BINDADDR=obfs4-0.0.0.0:444
TOR_PT_ORPORT=127.0.0.1:443
```

Figure 3: Obfs4 Proxy Configuration

```
[Unit]Description=Obfsproxy Server
[Service]EnvironmentFile=/var/lib/tor/pt_state/obfs4/obfs4.configExecStar
t=/usr/bin/obfs4proxy -enableLogging true -logLevelStr INFO
[Install]WantedBy=multi-user.target
```

Figure 4: Obfs4 Listening to the New Connection

```
# obfs4 torrc client bridge line
#
# this file is an automatically generated bridge line based on
# the current obfs4proxy configuration. EDITING IT WILL HAVE
# NO EFFECT.
#
# Before distributing this Bridge, edit the placeholder fields
# to contain the actual values:
#  <IP ADDRESS>    - The public IP address of your obfs4 bridge.
#  <PORT>          - The TCP/IP port of your obfs4 bridge.
#  <FINGERPRINT> - The bridge's fingerprint

Bridge obfs4 <IP ADDRESS>:<PORT> <FINGERPRINT> cert=<YOUR
CERT> iat-mode=0
```

Figure 5: Obfs4 Certification Key

After setting up a VPN server, the connectivity was tested using a VPN client software. In this project, Viscosity software was used as VPN client because it has obfuscation functionality. Viscosity is an OpenVPN client which is supported on Windows and Mac operating systems. Viscosity provides user friendly interface for creating, editing, and controlling VPN connections. Thus, it is easy for users new to VPNs to get started because of its clean and clear interface that simplifies menu such as creating, configuring, or importing VPN connections. Viscosity also caters for power and expert users by allowing full control over VPN connections, powerful routing options, custom scripts, and has the ability to use obfuscator as a transport method. VPN clients with obfuscation transport method are more secured and can bypass firewall.

TESTING AND RESULT ANALYSIS

Several tests were conducted throughout the project to ascertain the viability of SafeSearch when used on a public network. These three tests were network restriction penetration assessment, VPN server network performance and user acceptance test.

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA
(https://creativecommons.org/licenses/by-nc-sa/4.0//)

Network Restriction Penetration Assessment

VPNs are effective at securing a way to bypass geo-blocking and censorship, but government institutions and corporations that seek to control Internet access are always one step ahead. Various Anti-VPN technologies can detect and block VPN users from websites and local networks that violate the country's laws and regulations. This assessment was carried out to evaluate how effective SafeSearch is at penetrating network restrictions, where the results were obtained through passive observation.

The test carried out was simply a passive observation. In this case, the test was carried out at UiTM Perlis Branch, Arau campus. This test required the user to access a blocked website in the web browser for example, a torrent website. Then user had to connect to the VPN server and reload the page. If the website was accessible, the VPN worked. This meant that web filtering and deep packet inspection was easily bypassed.

Moving on to port blocking, users can simply try to torrent files through peer-to-peer connection. If torrent connections worked, the VPN can bypass port blocking. In the case of strict NAT, this can be tested through hosting online games. If the user could successfully host a game server, the test was a success. Finally, the user could check if the IP is blacklisted using a website called "Whatismyipaddress.com" and navigate through the website to find the IP blacklist check tool. Table 3 and Table 4 show the assessment score guideline alongside the network penetration test results.

Table 3: Assessment Score Guideline

Score Value	Score Meaning
1	Poor
2	Average
3	Excellent

Table 4: Network Penetration Test Result

Obstructions.	Description	Score (1-3)
Web filtering	A proxy firewall may block certain websites based on category tags.	3
Port blocking	Routers can allow only specific port to be forwarded, blocked ports will drop the packet.	3
Deep packet inspection	Deep packet inspection can drop a packet on the fly if it determines that the packet is going to a blocked destination address.	3
Strict NAT	Specifically, for hosted game servers Players from outside the LAN will not be able to join the server.	2
IP block	Certain ranges of IP addresses are blocked from a web server.	1
Total Score		12/15

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA (https://creativecommons.org/licenses/by-nc-sa/4.0//)

VPN Server Network Performance Assessment

Latency is a term used when delay happens in data communication over a network. It is time interval between the user's request and server response (Kwon, 2015). There are many factors that affect latency. Transmission mediums such as WAN or fiber optic cables have limitations that can affect latency simply due to their nature. Different routers have different approaches to analyse the header information of routed packets as well as in some cases, add additional information. Every hop a packet taken from router to router increases latency. Additionally, distance can also affect latency because of light propagation speeds in fiber optic cables. High latency creates bottlenecks in any network communication. The VPN Server performance was measured based on a single VPN server and all the clients can only communicate with this server tunnel (Wu & Xiao, 2019).

This assessment was carried out to measure the latency when a user browsed the web with a VPN and without a VPN. Three websites each hosted at different countries were visited to test the latency. The tools used for this testing was Azure Latency Test. Readings were taken three times every 60 seconds to calculate the average. Table 5 shows the result of the latency assessment and Figure 6 shows the graph of average latency in different countries.

Table 5: Network Penetration Test Results

Test every 60 seconds.	Latency (ms)					
	Singapore		UK		US	
	With VPN	No VPN	With VPN	No VPN	With VPN	No VPN
1	54	33	211	186	293	248
2	52	29	214	188	310	240
3	60	37	219	189	285	256
Average	55	33	215	187	296	248

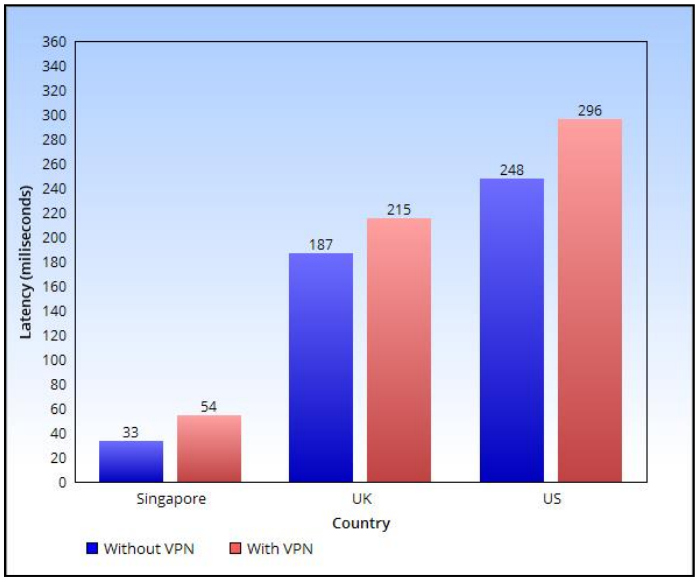

Figure 6: Average Latency in Different Countries

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA (https://creativecommons.org/licenses/by-nc-sa/4.0//)

Based on the graph above, the average latency increases when users tunnel their connection through a VPN server. One of the principal causes of network latency is distance, therefore the farther the client is from the server, the higher the latency. Therefore, latency in countries such as the UK and the US is higher. Distance coupled with the VPN server alongside the processes performed by the obfuscation proxy to hide traffic will also slightly increase latency. Although higher latency has a negative effect on a user's experience, it is a necessary trade-off to ensure their privacy is not intruded upon.

Bandwidth describes the maximum data transfer rate of a network or an Internet connection. It measures how much data can be sent over a specific connection within each duration of time. Sometimes, VPN encryption protocols can potentially cause the low bandwidth.

This assessment was carried out to measure download and upload speed using VPN and without VPN to find out how much bandwidth was lost in the process. The tests were accomplished with an ADSL Internet connection with download and upload speeds of up to 30 megabits per second. The tools used to measure bandwidth speed is a website called "speedtest.net". Both download and upload speeds were taken three times to accurately calculate the average speed. Table 6 shows the bandwidth test results and Figure 7 shows the average download and upload speed with VPN and without VPN.

Table 6: Bandwidth Test Result

Test	Bandwidth (mbps)			
	Download		Upload	
	With VPN	Without VPN	With VPN	Without VPN
1	20.80	24.82	25.56	29.63
2	25.12	27.97	20.44	29.52
3	17.00	26.97	17.22	31.13
Average	20.98	26.59	21.07	30.09

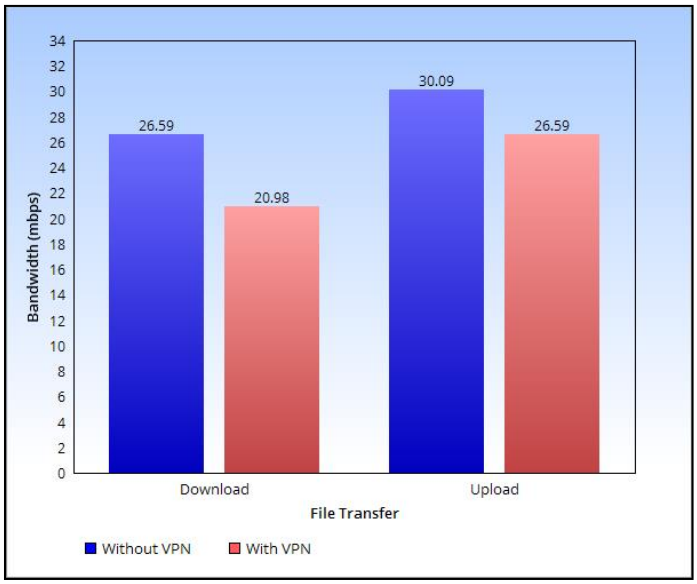

Figure 7: Average Download and Upload Speeds

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA (https://creativecommons.org/licenses/by-nc-sa/4.0//)

Based on the results above, bandwidth decreases when using VPN due to the overhead cost when going through the VPN tunnel.

User Acceptance Test

In software development, user acceptance testing is a process to evaluate whether the solution created fits the user's narrative. The goal of User Acceptance Test (UAT) is to ensure the software or hardware can both handle real-world tasks and perform up to development specifications. For this project, user acceptance testing was performed by giving 32 participants various questions to test their knowledge and awareness of cybersecurity issues such as the dangers of public Wi-Fi.

The questions were formulated using Google form because it is easy and intuitive to use. The participants were given access to the VPN Server to test its functionality. Participants were told to visit blocked websites, send any e-mail, or visit any social media website(s). The ingress and egress packet were captured by a packet sniffer software to test the VPN security (Girdhar, et. al, 2016). After hands-on experience with the VPN Server, participants were requested to offer feedback by answering the Google form questionnaire to evaluate their web browsing experience. Below are the questions used for User Acceptance Test.

Q1: By using SafeSearch, I would never have to worry about my privacy.

Q2: By using SafeSearch, information theft can be reduced.

Q3: By using SafeSearch, I can bypass network restrictions with ease.

Q4: SafeSearch does not impact my experience when browsing the web (lag, error 404, failed download).

Q5: Overall, I am satisfied with SafeSearch.

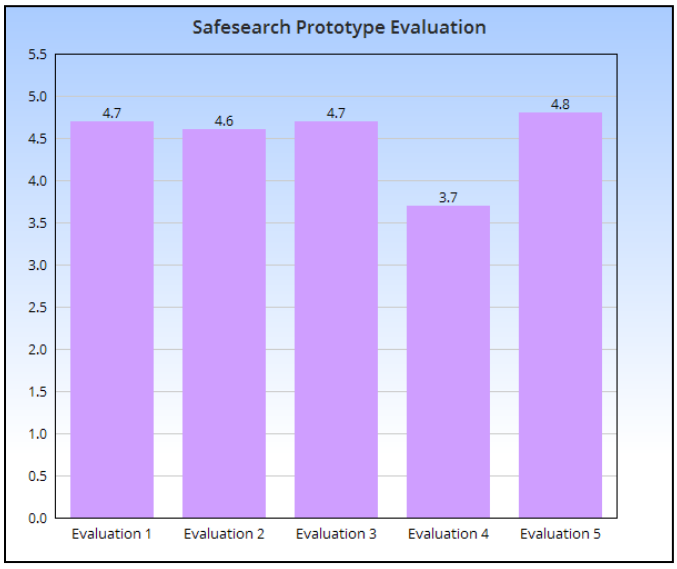

Figure 8: Mean Results of SafeSearch Prototype Evaluation

Based on Figure 8 above, the outcome was positive as the average mean result was above 4.5. The result shows that the project can perform up to development specifications. Moreover, participants were satisfied

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA
(https://creativecommons.org/licenses/by-nc-sa/4.0//)

with how SafeSearch functioned to secure their privacy by encrypting transmitted packets. This proved that the project was a success.

CONCLUSIONS AND RECOMMENDATIONS

The development for SafeSearch, Obfuscated VPN Server using microcomputer is low-cost implementation and simple to configure. This system was developed to provide secure connection and hide users' online traffic activity when browsing the web on public network. The purpose of this project was to develop a custom VPN server using a low-cost Raspberry Pi so that users can save cost by not subscribing to a premium VPN service or even free VPN. Furthermore, Safesearch is also more secure than any other VPN because it uses the OpenVPN protocol which performs great under high latency connections and provides strong encrypted connectivity to users' online activity. Based on user acceptance test that was conducted with 32 respondents, most of them were satisfied with the VPN and confident that it would protect their privacy. The network performance also declined slightly when connected to the VPN Server but has no significant difference in terms of latency and bandwidth. The SafeSearch network performance is considered to be successful as it is able to provide excellent connectivity along with strong security.

Since the VPN has been completed, there are several recommendations to improve the system for future use. One of the recommendations was to implement a DNS sinkhole inside SafeSearch to block out targeted advertisements. A DNS sinkhole sits between the user and the Internet to intercept any outgoing or incoming DNS requests to block certain domains from accessing the user's device, mainly advertisement websites. Besides that, it is also proposed that an Intrusion Detection System (IDS) be installed to further secure the traffic routed to SafeSearch network traffic which is injected with a virus that can be stopped before entering the network.

REFERENCES

Akram, Z., Saeed, M. A., & Daud, M. (2018). Real time exploitation of security mechanisms of residential WLAN access points. *Proceedings of the International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 1–5.

Aung, S. T., & Thein, T. (2020). Comparative analysis of site-to-site layer 2 virtual private networks. *Proceedings of the IEEE Conference on Computer Applications (ICCA)*, 1–5.

Bodis, M. (2017). *What is obfuscation?* https://hackernoon.com/what-is-obfuscation-30d8cc68b4d8.

Boldt, K., Kent, K. B., & Herpers, R. (2020). Investigation of encrypted and obfuscated network traffic utilizing machine learning. *Proceedings of the 30th Annual International Conference on Computer Science and Software Engineering (CASCON)*, 43–52.

Girdhar, P., Tech, M., Phool, B., Vishvidhalaya, M., & Khanpur, K. (2016). A study on detecting packet using sniffing method. *Journal of Network Communications and Emerging Technologies*, *6*(7), 45–46.

Goncharov, D. E., Zareshin, S. V., Bulychev, R. V., & Silnov, D. S. (2018). Vulnerability analysis of the wifi spots using WPS by modified scanner Vistumbler. *Proceedings of the IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 48-51.

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA
(https://creativecommons.org/licenses/by-nc-sa/4.0//)

Goto, H. (2018). Cityroam, providing secure public wireless LAN services with international roaming. *Proceedings of the Advances in Wireless and Optical Communications (RTUWO)*, 204–208.

Jaha, A. A. (2015). Performance evaluation of remote access vpn protocols on wireless networks. *International Journal of Computer and Information Technology*, *4*(2), 201–206.

Kwon, M. (2015). A tutorial on network latency and its measurements. In T. Soyata (Eds). *Enabling real-time mobile cloud computing through emerging technologies*, 272-293.

Niitsu, Y., Sakai, S. & Tezuka, K. (2019). Mutual authentication method in public wireless LAN by using BLE Beacon. *Proceedings of the Eleventh International Conference on Ubiquitous and Future Network (ICUFN)*, 449-451.

Pooja Karan, B., Akansha Santosh, M., Anurag Mohan, N., & Madhumita, C. (2018). Secure VPN server deployed on Raspberry Pi. *Journal of Network Communications and Emerging Technologies (JNCET)*, *8*(5), 31.

Taib, A.M., Zabri, M.T., Radzi, N.A.M., & Kadir, E.A. (2020). NetGuard: Securing network environment using integrated open VPN, Pi-Hole, and IDS on Raspberry Pi. *Proceedings of the Charting the Sustainable Future of ASEAN in Science and Technology*, 97-110.

Taib, A. M., Ishak, M. F. H., Kamarudin, N. K., Darus, M. Y., & Radzi, N. A. M. (2020). Securing network using Raspberry Pi by implementing VPN, Pi-Hole, IPS (VPiSec). *International Journal of Advance Trends in Computer Science and Engineering*, 9, 457-464.

Vanhoef, M., & Piessens, F. (2017). Key reinstallation attacks: Forcing Nonce Reuse in WPA2. *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 1313-1328.

Wu, Z. and Xiao, M. (2019). Performance evaluation of VPN with different network topologies. *Proceedings of the IEEE 2nd International Conference on Electronics Technology (ICET)*, 51-55.

Xu, Z. and Ni, J. (2020). Research on network security of VPN technology. *Proceedings of the International Conference on Information Science and Education (ICISE-IE)*, 539-542.

Copyright© 2021 UiTM Press. This is an open access article under the CC BY-NC-SA
(https://creativecommons.org/licenses/by-nc-sa/4.0//)