Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org
Volume 8, Issue 5, May (2018)

Secure VPN Server Deployed on Raspberry Pi

Pooja Karan Bist
Department of Computer Engineering, Mumbai University, PCE, New Panvel, India

Akansha Santosh Mekade
Department of Computer Engineering, Mumbai University, PCE, New Panvel, India

Anurag Mohan Nair
Department of Computer Engineering, Mumbai University, PCE, New Panvel, India

Madhumita Chatterjee
Department of Computer Engineering, Mumbai University, PCE, New Panvel, India

Abstract - With the increase in data accumulation, manipulation and the need for remote access, there is also a need for a secure network route or protocol through which users can access their data stored at a location far away from their current location. VPN Server is one of the most prominent and widely used network configuration aimed at supplementing the demand of remote access. The proposed system is focused on setting up a VPN server and securing the connection between the VPN host machine and the client that is accessing it remotely. The current VPN system though has security protocols deployed on it, it fails to comprehend the more advanced and complex threats to the system. The project aims at providing multiple layers of protection in the form of authentication during the connection establishment between a vpn client and the vpn server deployed on raspberry pi. The idea is to incorporate three layers of verification into the vpn authentication mechanism and eliminate any and all flaws that maybe present during the connection stages. The multiple layers include the different modules and mechanisms like Pluggable Authentication Module (PAM), Client Specific Authentication (Private Key), Lightweight Directory Access Protocol (LDAP). In addition to deploying an advanced security mechanism, the project also focuses on converting the client machine into a mobile hotspot which will in turn, act as a Wi-Fi sources for other Wi-Fi enabled devices in the proximity, thus, extending the VPN connection to all devices and not just your desktop pc or laptop. Finally, to make this structure portable, the whole project is deployed on a Raspberry Pi environment. This enables the system to become extremely portable, reusable and user friendly, thus allowing the VPN to be set up whenever and wherever required. A user-end GUI implementation is the last stage of the proposed system where a simple and user-friendly GUI is designed to enable the user to navigate through the different actions possible on the VPN server.

Index Terms— Raspberry Pi; VPN (Virtual Private Network); OpenVPN; PAM (Pluggable Authentication Module); Client Specific; Mobile hotspot.

1. INTRODUCTION

The constant and ever-increasing need for remote access spotlighted the emerging era of VPN- a virtual remote networking module. Using VPN, users are not only able to access their data remotely but also in a secured way through a private virtual tunnel. But again, the question arises, is the existing VPN system fully secured? Although the existing system does not have any fundamental flaws, there are few minor threats and vulnerabilities that can lead to unauthorized access to the server. As a consequence, it is equally important to deal with this minor threats and vulnerabilities in order to guarantee the user their privacy.

The proposed system gears up some additional features that resolves the minor threats and vulnerabilities with existing system. These additional features are the Raspberry Pi - an all-time active device and a low power consumer; A Multi-Tier Authentication Module - a high level authentication assurance; A Hotspot Module - VPN connection extender.

2. LITERATURE REVIEW

Aparicio Carranza and Constadinos Lales, [1] give the theory of how data is insecure while accessing the public internet and how one can use the Raspberry Pi (A cheap microcomputer) as a VPN server to a home network; in order to create a VPN connection between a home network and the public internet.

Thomas Berger analyzed the current VPN technologies, [2], such as Internet Protocol Security (IPSec), Layer Two Tunneling Protocol (L2TP), and Point to Point Tunneling Protocol (PPTP). The analysis includes one significant drawback which concerns all tested technologies - the dramatic loss of performance and throughput. IPSec suffers from complex tunnel negotiation process, L2TP, when combined with IPSec, results in excessive data overhead whereas for PPTP, it’s security level is not sufficient for critical applications. Hence, to enhance the security and reliability of a VPN, a strong authentication mechanism is required on top of the traditional username and password authentication credentials.

Anupriya Shrivastava, M.A. Rizvi proposed the concept of external authentication approach for VPN using LDAP protocol [3]. The advantage of this approach is that user information is stored in a dedicated authentication server which can have a large pool of organized, directory-based user data along with greater robustness and security. Hence this approach proposes to extend the functionality of LDAP server in order to strengthen the authentication process of VPN.

L. Caldas-Calle, J. Jara Member, M. Huerta and P. Gallegos [4] surveyed that the highest throughput is for RP3 and the lowest for RPB. Values indicate dependence latency buffer each model based on the average time RTT, it is more evident in RPZ and RPB. Packages with size beyond the fragmentation point suffer QoS decrease, due the need to fragment packets. The CPU power of each Raspberry Pi model is an important factor affecting the QoS parameters of a wireless VPN. Introducing VPN to secure communication implies more complex process in communication that requires more from hardware.

3. EXISTING SYSTEM

In an existing VPN system, when a client requests for a connection the initial step taken is to match the certificate files. These certificate files contain the private key and the Signature Encryption Algorithm used to authenticate the client to the server. If the attacker is able to get this client file he can easily break into your private network and this can be a loophole for the existing VPN systems.

Where the VPN is used to protect your data transmission over the internet, there’s a security protocol namely LDAP which is used for authentication purpose while accessing the directory files. Fundamentally, LDAP using operations such as “Bind” operation authenticates the user, willing to access the directory, through the username and password included in the Bind operation.

4. PROPOSED SYSTEM

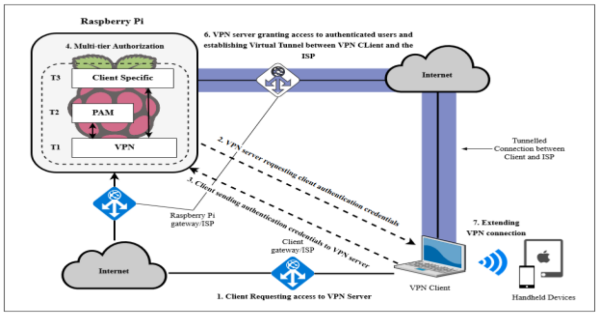

Raspberry Pi
4. Multi-tier Authorization
T3
Client Specific
T2
PAM
T1
VPN
6. VPN server granting access to authenticated users and establishing Virtual Tunnel between VPN CLient and the ISP
Internet
Tunnelled Connection between Client and ISP
2. VPN server requesting client authentication credentials
3. Client sending authentication credentials to VPN server
Raspberry Pi gateway/ISP
Client gateway/ISP
7. Extending VPN connection
Internet
VPN Client
Handheld Devices
1. Client Requesting access to VPN Server

Fig. 1. Proposed Architecture.

The proposed system leaves the basic functioning of the VPN server untouched and adds on an extra feature for better usability and security. This extra layer of security is provided by a Multi-Tier Authentication Module.

The Multi-Tier Authentication Module provides three tiers of authentication.

Tier 1: This tier comprises the basic functionalities of the VPN that includes authentication of client files.

Tier 2: This tier incorporates a PAM module - Pluggable Authentication Module, that uses low level authentication mechanisms to integrate different modules and use one simple authentication for all of them. This authentication is provided at the server side. PAM provides the same level of security as LDAP but in a more optimum way. Additionally, it also allocates a dedicated desktop for each client.

Tier 3: This tier generates a Client Specific Private Key that eliminates the possibility of multiple users using one client file to log in to the VPN server.

The connection once established, on the client machine, can be extended to other handheld devices using the Wi-Fi hotspot that is created on the client machine.

The user end machine has a simple GUI to operate and maneuver through the operations of the VPN server and its functionalities. The GUI contains three buttons: one to connect to the VPN server following up the entire authentication process, second button allows user to upload a file from the VPN server and the third button is used to download the shared files from the VPN server.

5. IMPLEMENTATION MODEL

Implementing the proposed system focuses on collaborating different modules that works individually to function in a multi-tier architecture that ensure higher level of authentication for a VPN deployed on a Raspberry Pi. The modules incorporated in the system are:

PAM- Pluggable Authentication Module: PAM is widely used for authenticating users against a system that has accounts created on it. Each PAM authentication is done by comparing whether the entered credentials belong to a user account in the system OS. If so, access is granted. The PAM module also, by default, indicates that each user will have their own desktop on the server system. It is useful for a simple user credential authentication.

Client Specific Secret Key Authentication: This authentication layer lies above the PAM module. The client must first clear the traditional authentication after which he encounters the PAM authentication and finally after that, he will be reaching this last layer of authentication which is encoded to the client file and a secret key is attached. The key is generated at the time of ‘.ovpn’ file creation on the server side.

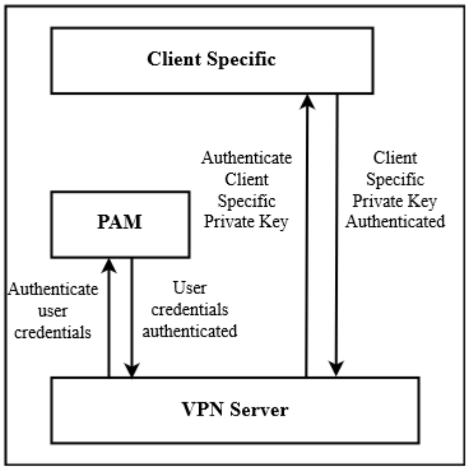

Client Specific
Authenticate Client Specific Private Key
Client Specific Private Key Authenticated
PAM
Authenticate user credentials
User credentials authenticated
VPN Server

Fig. 2. Multi-tier Authentication Module.

Shared Folder: There exists a shared folder between the VPN server and the client that is going to connect to it. This folder is hidden behind the different layers of authentication and can only be accessed once the VPN connection is established. For better understanding, consider a scenario:

Scenario: A client wishes to upload/download a file from the VPN server. For this, the client will have to follow the following sequence of actions:

1. Request to server for VPN access.

2. Pass through multiple layer of authentication

3. Establish a secure connection between the server and itself.

4. Then access the shared folder to perform desired file transfer.

Hence, even to share a file, the user must pass through the multi-tier architecture and connect to the VPN first.

Hotspot: Extending the VPN connection was our final step in the implementation module. The client has established its own secure connection with the VPN server and now wishes to connection their handheld devices to the same network. For this, we have implemented a hotspot module that will allow the user to extend the connection to nearby devices using SSID-Password method.

Algorithm:

Here:

RPI: Raspberry Pi

Client: VPN Client

$\text{VPN}_{S:}$ VPN Server

PAM: Pluggable Auth. Module

CSM: Client Specific Auth. Module

VPN_H: VPN Hotspot

Client_F: Client File

CA: Certificate Authority

Cert: Client Certificate

SK: Secret Key

PK: Private Key

$\text{H}_{\text{device:}}$ Handheld Devices

$\text{H}_{\text{credentials}}$: Hotspot Credentials

Raspberry Pi consists of three authentication modules:

RPI [VPN_s, PAM, CSM]

Step 1: Client requesting VPN_s for Access by Sending its Client_F

$$\text{Client} \longrightarrow \text{VPN}_S\text{: Client}_F\text{ [CA, Cert, \{SK\}]}$$

Step 2: VPN authenticates the File received as an access request by the Client.

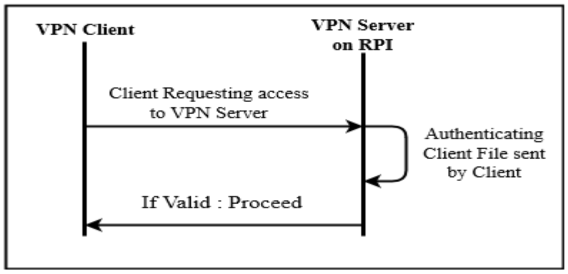

VPN Client
VPN Server on RPI
Client Requesting access to VPN Server
Authenticating Client File sent by Client
If Valid : Proceed

Fig. 3. Phase I.

Step 3: If Client = Valid; VPN_s requests client to provide the PAM credentials.

Step 4: Client sends the PAM credentials that consist of a username and a password to PAM module.

$$\text{Client} \longrightarrow \text{PAM}_{:}\text{ Credentials [username, password]}$$

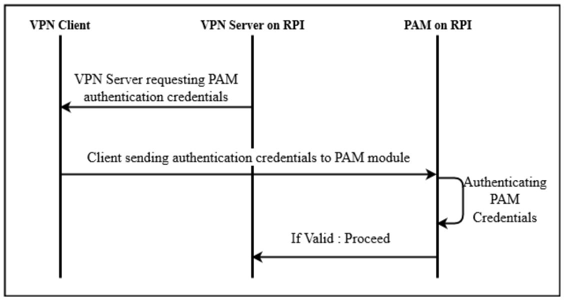

VPN Client
VPN Server on RPI
PAM on RPI
VPN Server requesting PAM authentication credentials
Client sending authentication credentials to PAM module
Authenticating PAM Credentials
If Valid : Proceed

Fig. 4. Phase II.

Step 5: If Credentials = Valid; VPN$_s$ requests client to provide the Client Specific Private Key.

Step 6: Client sends the Client Specific Private Key to CSM.

$$\text{Client} \longrightarrow \text{CSM}: [\{\text{PK}\}]$$

Step 7: If {PK} = Valid; the CSM permits VPN$_s$ to grant access to client.

Step 8: The VPN grants access to the Client and a Virtual Tunnel is established through ISP.

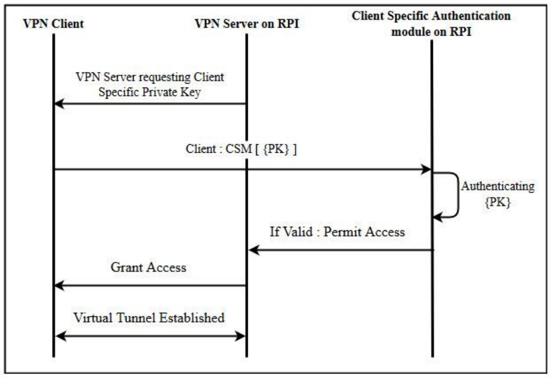

VPN Client
VPN Server on RPI
Client Specific Authentication module on RPI
VPN Server requesting Client Specific Private Key
Client : CSM [{PK}]
Authenticating {PK}
If Valid : Permit Access
Grant Access
Virtual Tunnel Established

Fig. 5. Phase III.

Step 9: Once the VPN connection is established, the Client turns itself into VPN$_H$, in order to extend the VPN connection to H$_{Device}$.

$$\text{Client}: = \text{VPN}_H$$

Step 10: To connect to the VPN$_H$, H$_{Device}$ provides it's H$_{credentials}$ that consist of SSID and password to the VPN$_H$.

$$\text{H}_{\text{Device}} \longrightarrow \text{VPN}_\text{H}: \text{H}_{\text{credentials}}[\text{SSID,password}]$$

Step 11: If Hcredentials = Valid; the VPN connection is extended to H$_{Device}$.

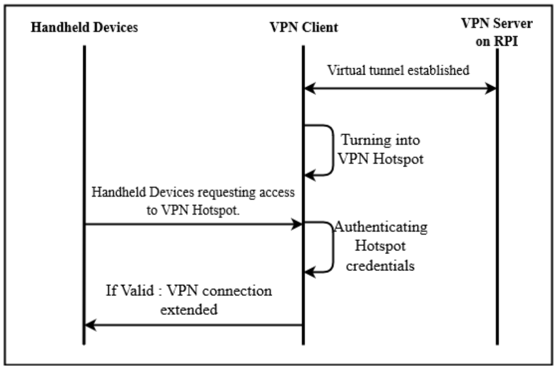
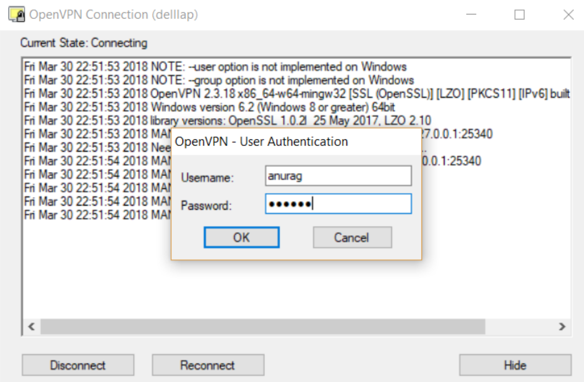

Handheld Devices
VPN Client
VPN Server on RPI
Virtual tunnel established
Turning into VPN Hotspot
Handheld Devices requesting access to VPN Hotspot.
Authenticating Hotspot credentials
If Valid : VPN connection extended

Fig. 6. Phase IV.

The result of implementation of the individual modules was a sophisticated and secure 3-tier authentication system that enables user to connect securely to the VPN server.

Output Screenshots:


OpenVPN Connection (delllap)

Current State: Connecting

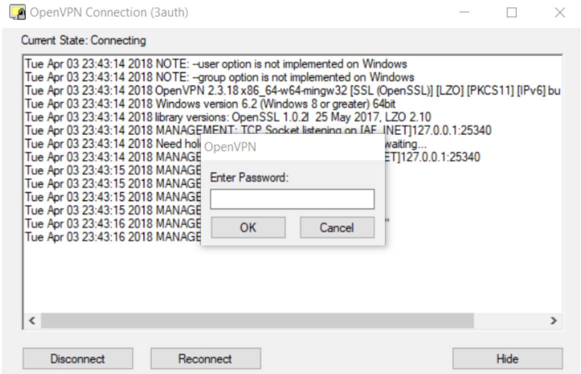
```
Fri Mar 30 22:51:53 2018 NOTE: --user option is not implemented on Windows
Fri Mar 30 22:51:53 2018 NOTE: --group option is not implemented on Windows
Fri Mar 30 22:51:53 2018 OpenVPN 2.3.18 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [PKCS11] [IPv6] built
Fri Mar 30 22:51:53 2018 Windows version 6.2 (Windows 8 or greater) 64bit
Fri Mar 30 22:51:53 2018 library versions: OpenSSL 1.0.2l  25 May 2017, LZO 2.10
Fri Mar 30 22:51:53 2018 MAN                                   27.0.0.1:25340
Fri Mar 30 22:51:53 2018 Nee                                   ..
Fri Mar 30 22:51:54 2018 MAN                                   0.0.1:25340
Fri Mar 30 22:51:54 2018 MAN
Fri Mar 30 22:51:54 2018 MAN
Fri Mar 30 22:51:54 2018 MAN
Fri Mar 30 22:51:54 2018 MAN
```

OpenVPN - User Authentication

Username: anurag

Password: ••••••

Fig. 7. Post file verification, PAM authentication prompt.


OpenVPN Connection (3auth)

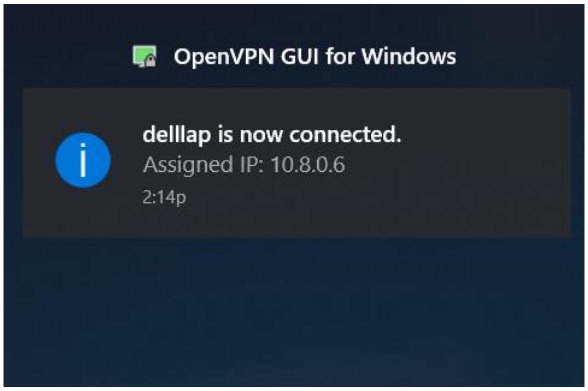
Current State: Connecting

```
Tue Apr 03 23:43:14 2018 NOTE: --user option is not implemented on Windows
Tue Apr 03 23:43:14 2018 NOTE: --group option is not implemented on Windows
Tue Apr 03 23:43:14 2018 OpenVPN 2.3.18 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [PKCS11] [IPv6] bu
Tue Apr 03 23:43:14 2018 Windows version 6.2 (Windows 8 or greater) 64bit
Tue Apr 03 23:43:14 2018 library versions: OpenSSL 1.0.2l  25 May 2017, LZO 2.10
Tue Apr 03 23:43:14 2018 MANAGEMENT: TCP Socket listening on [AF_INET]127.0.0.1:25340
Tue Apr 03 23:43:14 2018 Need hol                              waiting...
Tue Apr 03 23:43:14 2018 MANAGE                                ET]127.0.0.1:25340
Tue Apr 03 23:43:15 2018 MANAGE
Tue Apr 03 23:43:15 2018 MANAGE
Tue Apr 03 23:43:15 2018 MANAGE
Tue Apr 03 23:43:15 2018 MANAGE
Tue Apr 03 23:43:16 2018 MANAGE                                "
Tue Apr 03 23:43:16 2018 MANAGE
```

OpenVPN

Enter Password:

Fig. 8. Post PAM, client specific secret key prompt.


OpenVPN GUI for Windows

delllap is now connected.

Assigned IP: 10.8.0.6

2:14p

Fig. 9. Post 3-tier auth. VPN connection established.

6. APPLICATIONS

- When you want to share file remotely and securely: The security of the files while sharing it on an open network is always at risk. Also, a question arises of access the files when you are away from home. The proposed system resolves these issues, since the VPN enables the user to access their files remotely and through a secured virtual tunnel.

- When You Want Privacy and Advocacy: When you are away from home and you wish to connect to a secure network, VPN is the way to go about it. It not only strengthens your connection but also increase privacy and advocacy by encrypting all the data transferred whether at home or abroad.

- When you want a secure Wi-Fi connect: Once connected to the system, the client can extend its secure VPN connection to the nearby devices by converting itself into a Wi-Fi hotspot. This enables the non-client devices to use the secure network of the VPN.

7. CONCLUSION

The VPN server deployed on Raspberry Pi enables user to access their virtual connection to home network at any time and on a low power consumption. The multi-tier authentication assures the security of connection establishment between the server and the client, while the hotspot module extends the VPN connection to the handheld devices.

REFERENCES

[1] Constadinos Lales Aparicio Carranza. Using the raspberry pi to establish a virtual private network (vpn) connection to a home network. International Conference on Portable devices, 2014.

[2] Thomas Berger. Analysis of current vpn technology. First International Conference on Availability, Reliability and Security, 2012.

[3] M.A. Rizvi Anupriya Shrivastava. External authentication approach for virtual private network using ldap. First International Conference on Networks and Soft Computing, 2014.

[4] M.Huerta L.Caldas-Calle, J.Jara Member and P.Gallegos. Qos evaluation of vpn in a raspberry pi devices over wireless network. International Caribbean Conference on Devices, Circuits and Systems, 2017.

[5] Use PAM to Configure Authentication. https://www.digitalocean.com/community/tutorials/how-to-use-pam-to configure-authentication-on-an-ubuntu-12-04-vps. October 3, 2013.