Abstract—Public communication path, including the Internet has security risk that is vulnerable to disruption. Whereas the local area network (LAN) has limited coverage area to communicate. In this case, Universitas Muhammadiyah Riau (UMRI) has 2 (two) separate buildings which makes it difficult to build a local network. The use of wired communication will require considerable expense. In this work we apply Virtual Private Network (VPN) as the local network by using public medium. The network server does not use public IP but enables it to run anywhere as a computer network with public IP on the internet. We applied for a Linux the OpenVPN Server and OpenVPN Client for Linux and Windows environments. There are 2 main servers in the central building and 59 clients on both buildings that are implemented. This scheme not only improves accessibility on the local network, but also protecting the security of local data on a public medium.

Keywords—Linux, local, network, OpenVPN, public, private, virtual, VPN.

I. INTRODUCTION

Virtual Private Network (VPN) are providing a secure and private connection in order to ensure communication privately and securely over an insecure medium public network. VPN are capable of establishing a secure virtual link between different buildings [1][2][3]. VPN connections establish secure connections between a remote user and a main network by encrypting packets sent though the Internet rather than building a true private network [4][5][6]. Application of VPN allows users to use local network as if the Internet network. The use of VPN will make users more comfortable in terms of work. VPN makes it easy for users to do the job even if they are out of the office[7]. VPN is categorized as a form of wide area network. Government agencies, large corporates, and educational institutions use VPN technology to enable remote users to securely connect to a private network [5][7].

II. PRESENT SITUATION ANALYSIS

At present, Universitas Muhammadiyah Riau (UMRI) uses an academic information system as a whole. The application of information systems are running in the main campus as internal network, which need to ensure security while opening up the main campus network and local sub campus network connections. Lectures in sub campus cannot access to the main campus information system efficiently and safely, as well as enhance the internal information sharing efficiency of colleges and universities to ensure that all kinds of information within the university office system for all users are available.

Client (lecturers and administrators) in the sub campus need to access the academic information, also need to access the internal network on main buildings. That requires network can provide remote access, convenient and low cost in the office for lecturers, making all kinds of information system can be visits whenever and wherever.

This system runs on a computer network. There is a server side as a database. The server is accessible to all users in the network. Currently UMRI has two different campus locations although it is still in a city. Distance between main campus with sub campus about 3 km. This distance is far enough if it should be connected with the cable. The main campus has three faculties, namely Faculty of Computer Science, Faculty of Economics and Faculty of Communication, while on campus 2 there are two faculties, namely Faculty of Engineering and Faculty of Mathematics for Natural Science & Health. UMRI uses the Academic Information System which is named Sisfo or SIA. Sisfo is used by Lecturers and other Administration to finance including students. For security reasons this system is accessed locally.

Universities need to share internal resources with each other buildings. But the sharing of internal resources is different from the relationship between the main campus and the sub campus, which use a completely open sharing mode, the sub campus belong to the main campus use the same internal network in logic. The way of sharing university and other building resources are limited, the sub campus may not open all of the internal information on main campus. While this system is only used for one building, there is occur the problem that the building has been increased. The problems arise that can only be solved with the first two things to build a system based on the internet means raised to the public level. It just that security needs to be a concern. Both need to be built fiber optic-based computer network, and fiber optic is quite expensive so this solution quite hard to be used.

Management policies install internet on both campuses. The Internet installed to make sure communications becomes comfortable on both buildings but Sisfo remain in the local position so that sub campus cannot access and use Sisfo as
well. We offer to solve term to this condition is to build a VPN as an alternative solution to this problem.

III. BACKGROUND AND RELATED WORK

OpenVPN is an open source software application that implements VPN techniques. It is a full-featured Secure Socket Layer (SSL) VPN which implements OSI layer 2 or 3 secure network extension using the industry standard SSL protocol, supports flexible client authentication methods, encryptions and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface, the data pass through of the VPN and tunnel technology [5][8][9][10]. OpenVPN is an open-source application but also available for other platform such as MS Windows, Linux, Mac OS X and so on to developed a secure communications between machines in separate locations over a potentially insecure link [1][11][12]. OpenVPN supports both TCP and UDP protocols even if, as other VPNs, it works better over UDP than TCP, especially if the link between the sites have packet drops. TCP introduces a slight overhead since packets are prefixed by their size (16 bit, unsigned). OpenVPN traffic is encapsulated inside UDP or TCP tunnel [6][8][11].

The OpenVPN consists of a set of installation and configuration tools which allow for simple and rapid deployment of VPN remote access solutions using the OpenVPN open source. This configuration is most commonly seen when the Access Server resides in an internal network, providing VPN access to users outside the main network. In this configuration the Access Server has one network interface connected to the private network. This scenario is illustrated in Figure 1.

![OpenVPN Access Server Topology](image)

**Figure 1 OpenVPN Access Server Topology [13]**

This configuration is most commonly seen when the Access Server resides in an internal corporate network but it also has its own public IP address.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenVPN Access Server</td>
<td>The OpenVPN server daemon along with the Access Server’s configuration and maintenance software running on a server computer</td>
</tr>
<tr>
<td>User</td>
<td>An individual attempting remote access to private network resources via the public Internet</td>
</tr>
<tr>
<td>Client</td>
<td>A computer (operated by a user) running OpenVPN client software in order to gain access to private network services via the OpenVPN Access Server</td>
</tr>
<tr>
<td>User Credentials</td>
<td>A username and password used to authenticate a user</td>
</tr>
<tr>
<td>Client Configuration File</td>
<td>A file which contains all of the information required for an OpenVPN client to securely connect to the OpenVPN server. User credentials are not included in the client configuration</td>
</tr>
<tr>
<td>Connect Client</td>
<td>A client running on the Access Server which delivers client configuration files and/or pre-configured Windows client installer files to authenticated users. The Connect Client also allows for a user to login and connect through the browser</td>
</tr>
<tr>
<td>Admin Web UI</td>
<td>A Web server running on the Access Server which is used by the administrator to configure the settings of the Access Server</td>
</tr>
</tbody>
</table>

An OpenVPN Access Server deployment consists of one server, many clients and many users. Each client machine in this topology uses the public IP network (the Internet) to communicate with the OpenVPN Access Server and thereby gains VPN-protected access to the private IP Network connected.

IV. METHODOLOGY

Tools and materials used are computer hardware and network including Internet and software. The computer device used consists of two computer server devices with the operating system Fedora Linux. Both server computers are located on the UMRI student campus server room. Seen in figure 2 there is a proxy computer with public IP (27.8.67.38) tun 10.8.0.1 (ip vpn) and campus system computer with local
Another computer needed is a client computer that is self from a computer with ubuntu linux operating system and windows.

Figure 2. Main Network Architecture of UMRI

The installation configuration file is shown below:

```
# yum install openvpn
# cd /usr/share/doc/openvpn/examples/easy-rsa/2.0
# vim vars
# easy-rsa parameter settings

# NOTE: If you installed from an RPM,
# don't edit this file in place in
# /usr/share/openvpn/easy-rsa --
# instead, you should copy the whole
# easy-rsa directory to another location
# (such as /etc/openvpn) so that your
# edits will not be wiped out by a future
# OpenVPN package upgrade.
# This variable should point to
```
The configuration file of the server is shown below:

```bash
# the top level of the easy-rsa
# tree.
export EASY_RSA="pwd"
export OPENSSL="openssl"
export PKCS11TOOL="pkcs11-tool"
export GREP="grep"
# with easy-rsa.
export

KEY_CONFIG=$EASY_RSA/whichopensslcnf
EASY_RSA

# it correctly!
export KEY_DIR="$EASY_RSA/keys"

# Issue rm -rf warning
echo NOTE: If you run ./clean-all, I will be doing a rm -rf on $KEY_DIR

export OPENSSL="openssl"
export PKCS11TOOL="pkcs11-tool"
export GREP="grep"

# with easy-rsa.
export

KEY_CONFIG=$EASY_RSA/whichopensslcnf
EASY_RSA

# the requested executables
#
export OPENSSL="openssl"
export PKCS11TOOL="pkcs11-tool"
export GREP="grep"
# This variable should point to
# the openssl.cnf file included
# with easy-rsa.
export

KEY_CONFIG=$EASY_RSA/whichopensslcnf
EASY_RSA

# Edit this variable to point to
# your soon-to-be-created key
# directory.
#
# WARNING: clean-all will do
# a rm -rf on this directory
# so make sure you define
# it correctly!
export KEY_DIR="EASY_RSA/keys"

# Issue rm -rf warning
echo NOTE: If you run ./clean-all, I will be doing a rm -rf on $KEY_DIR

# PKCS11 fixes
export PKCS11_MODULE_PATH="dummy"
export PKCS11_PIN="dummy"

# Increase this to 2048 if you
# are paranoid. This will slow
# down TLS negotiation performance
```
# as well as the one-time DH parms
# generation process.
export KEY_SIZE=1024

# In how many days should the root CA key expire?
export CA_EXPIRE=3650

# In how many days should certificates expire?
export KEY_EXPIRE=3650

export KEY_COUNTRY="INA"
export KEY_PROVINCE="RIAU"
export KEY_CITY="Pekanbaru"
export KEY_ORG="UMRI"
export KEY_EMAIL="masharum@yahoo.com"

# source ./vars
akan tampil perintah selanjutnya sebagai berikut:
NOTE: If you run ./clean-all, I will be doing a rm -rf on
/usr/share/doc/openvpn/examples/easy-rsa/2.0/keys
#
# ./clean-all
# ./build-ca ca

Figure 3. Key Server Configuration

This command serves to delete all the keys that have been made, it is recommended to be careful when using the following command because it will delete all the keys that have been made before. Next live create a key server with the command as follows:

Figure 4. File dh1020.pem

The command “ls” is to view directory, should directory keys have been formed, then enter into the keys directory with the command cd keys, then check directory if there are 4 create the file ca.crt ca.key index.txt serial, then the creation of key ca is finished

Making the key dh with the following command:

Figure 5. Key Server Build

The basic command to ensure that key server

V. ANALYSIS EVALUATION

This research results shows that configuration running well and prove that connection between main building and sub campus can build well.

A. Server Configuration

Application server configuration on Fedora Linux under the basic command to ensure that key server:

# ./build-ca ca

The basic command to ensure that key server
B. Client Configuration

The key server is formed, the next step is just make the key for the client. The key for the client should be differentiated by the name of each user. UMRI has two campuses namely the main campus and sub campus. The main campus is separated from sub campus but the information system used in the main campus should be accessible to the sub campus. The main campus using the local server and sub campus has 20 computers consisting of two faculty, lecturers are on sub campus using personal PC as much as 10 key, main campus also have 5 (five) key message, admin also have counted 5 people for admin, and lecturers main campus beside the department also have 10 key message, thus 56 (fifty six) key client must be created.

Given the key to be made enough required the name of each according to his needs, in this research the naming of key based on faculty in sub campus that is Natural Science and Engineering and also lecturers.

C. Network Authentication and Configuration

The result keys that have been formed and stored in the folder keys are then moved to the folder /etc/openvpn, with the command as follows:

```
# cp -r keys / etc / openvpn
```

![Figure 7. Key Client Build](image)

Figure 7. Key Client Build

If there is no warning means is finished, and VPN can be used next only to live other client configuration in accordance with the operating system used. The next client is Ubuntu 10 and Linux mint, these two operating systems are derived from

![Figure 8. Key Client Stored](image)

Fig. 8 Key Client Stored

The final analysis of making a VPN server is to configure server.conf, configurations as follows:

```
# cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn
# gunzip server.conf.gz
# gedit server.conf
local 100.10.0.10 #ip dari isp
port 1194
;proto tcp
proto udp
;dev tap
dev tun
cia ca.crt
cert server.crt
key server.key # This file should be kept secret
dh dh1024.pem
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
client-to-client
keepalive 10 120
comp-lzo
persist-key
persist-tun
status openvpn-status.log
verb 3
# /etc/init.d/openvpn restart
```
the Debian installation command is also the same with Ms Windows command. At last a VPN created client on Windows operating system for Windows users. The first step is to download OpenVPN on the official OpenVPN site. The configuration in accordance with the required key the same as the steps on the previous operating system. Then copy the required key file to c:\Program Files\openvpn\config, locate the client .conf file copied also in the config folder. Run vpn by right click on sismbol openvpn GUI, then click connect, Next just use like normal internet connection.

VI. CONCLUSIONS

This research result and analysis shows that the remote access on VPN allow an access through local computers at anytime and anywhere. Based on VPN, private connection over public IP is more secure than internet. improve the safety of campus network interconnection and enhance the accessing experience of users outside the campus to access the resource in remote. But VPN requires a large scale bandwidth to leverage local systems on the public network. The configuration on the client is quite hard and unhurried for the administrator. Users will also have to an experience one, so special training is required for user.

Based on local area network interconnection and remote access requirements, UMRI VPN security interconnection scheme, has a certain practical significance. Firewall technology on VPN improve the safety of campus network interconnection and enhance the accessing experience of users outside the campus to access the resource in remote. This research provide a reference for interconnection network to other institution at the same case with UMRI

ACKNOWLEDGMENT

This research is an implementation of local network at UMRI. We thank UMRI that this paper can be reported and supported for publication.

REFERENCES