Jason Stuemke & Kathryn Lovett
Instructor Steve VanDevender
CIS 399 - Unix
12 August 2016

Final Project Writeup

For our final project, we successfully set up a VPN service on our Server Instance Maeve for our users' benefit. In this document, we will discuss how we have met the goals we outlined in our proposal for adding a VPN to our services, as well as how we have tested our completion of those goals. We will discuss how a VPN service would affect the user population, and any potential issues that we may have to face by enabling the VPN service. We will cover any maintenance required for this service and, lastly, we will review the documentation that would be required by our VPN service.

We successfully met our goal of setting up a VPN that allows our users to connect to our VPN connected machine (Maeve) as if they are on the same network. We have been unable to test our ability as of yet, as we do not have a network storage location set up, but are planning to do so in the near future. We have, however, been able to successfully connect to our Server Instance Maeve over a VPN connection using the application Tunnelblick. In order to make such a connection possible, we started out by installing the openvpn package on Maeve. To do so, we ran these commands:

```
sudo yum install -y openvpn
sudo modprobe iptable_nat
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
sudo iptables -t nat -A POSTROUTING -s 10.4.0.1/2 -o eth0 -j MASQUERADE
```

We then navigated to our openvpn folder on Maeve, and created a key called ovpn.key:

```
cd /etc/openvpn
sudo openvpn --genkey --secret ovpn.key
```

Next, we created a server configuration file for our VPN called openvpn.conf. Our configuration file contains the following information:

```
port 1194
proto tcp-server
dev tun1
ifconfig 10.4.0.1 10.4.0.2
status server-tcp.log
verb 3
secret  ovpn.key
```

Next, we started our VPN service on Maeve. We moved our key to the client computer, and proceeded to install the OpenVPN GUI (in our case, Tunnelblick). Once installed, we created a configuration file called myconfig.ovpn for the client side containing the following information:

```
proto tcp-client
remote <your EC2 IP here>
port 1194
dev tun
secret "C:\\Program Files\\OpenVPN\\config\\ovpn.key"
redirect-gateway def1
ifconfig 10.4.0.2 10.4.0.1
```

After associating our client-side configuration file with Tunnelblick by dragging the file to the application GUI, we were able to set up a VPN connection to Maeve.

Our user population will have greater access to their data through our VPN service. They will be able to connect to our Server Instance and thereby access data stored there that they otherwise would not be able to acquire. This increases their mobility, as well as their ability to work with confidential data in a secure manner. User support will change by requiring that we maintain users' access to the VPN

functionality of our Server Instances. We will have to deal with any issues connecting or outages of our VPN service.

Security issues that we face with our VPN service mirror those involved in making regular user accounts. We would want to ensure that all the user accounts created are authorized, and have appropriate login credentials associated with the accounts. The software we are using, OpenVPN, typically utilizes two forms of authentication: static key and TLS. Our VPN service is currently set up with a static key, but we would most likely switch to TLS once we implement access at a user level, where the user's username and password as well as status determines their access to our VPN service. We have also created our VPN so that all network traffic goes through it, preventing any potential security risks involved in implementing a split-tunnel VPN.

We have completed this project by implementing the VPN service in our Server Instance Maeve and ensuring that it is working correctly by connecting to it. In order to maintain the project in the future, we will need to make sure our Server Instance does not go down, verify that deactivated user accounts no longer have access to VPN and that active users maintain access to our VPN service. In an effort to prevent outages of our service, we will implement OpenVPN software on our Puppet Master, and use a Load Balancer so that if, for example, Maeve went down, users would maintain access to our server over VPN through Clarence (our other Server Instance). Lastly, in order to extend our service to our users, we will require in-depth documentation on how users can install the necessary software and access our Server Instance over VPN.