James Kang & Owen McEvoy

CIS 399

Steve VanDevender

Summer 2019

Final Report

For the final project, our group (Team Rocket) decided that we would be installing Emacs on our instances and allow each of our users to be able to use Emacs on our system. To test whether or not our attempts were successful, we coordinated with our class mates and had them log into our instances to use Emacs. We believe that our project was successful because each user we have contacted was able to use Emacs and install a package of their choice.

Our project was inspired by the fact that not everyone wants to use vi or vim and this allows the users to use a more advanced and personable editor should they choose to do so. The effect on the user populations should either be no change (for the users who would rather use vi or vim), or it will result in positive change (for users who would rather use Emacs). The user support issues we might face is that some of the users will be unaware of how to use Emacs. In this case, they could either use either vi or vim, or they could read the user documentation which we have provided which gives a basic tutorial. One nice thing about Emacs is that it comes with a thorough tutorial that the users should reference once they begin growing comfortable with it.

Since Emacs has so many customizable features, there are many security risks that could arise. The solution we found for this issue would be only allowing packages from gnu, the creators of emacs, to be installed. Another issue that we may run into would occur if a user wished to install every possible package Emacs offers, thus taking up a lot of our system’s memory. To address this potential issue, we could restrict both the number of packages a user can install or the maximum size of a package a user can install.

We were able to install Emacs on our system easily using a few lines of puppet code. The real challenge lied in administering a default configuration file for all of the users, installing helpful packages using apt for our users, and installing packages within emacs that our users will likely need. One example of a package we installed using apt, was python-mode which helps make editing, debugging, and running python programs much easier. An example of a package we installed within Emacs, was context-coloring which makes highlights the code in your current scope. If we were to continue working on this project, the next two goals would be restricting user packages based off of their source and their size.

For an example of the user documentation we wrote to give our users a brief introduction to Emacs, please refer to our GitHub. It provides a quick explanation for how to open a file, close/save Emacs, and install new packages. The image shown below shows the puppet code we used to ensure that Emacs is installed on our instances.

