Matt Ping

Brian Wilhelm

CS 575

Final Project

**Malware Information Management Solution**

Malware has been increasing in scope and complexity in recent years. With this increase comes numerous variants of existing malware. In order to better keep track of the malware that exists, our team seeks to implement a solution for capturing information related to malware. Rather than simply providing an interface to track this information, our goal is to provide a service layer to handle the management of data and apply appropriate rule checking. This will give users the ability to incorporate the data collected into another solution or create their own user interface if they choose.

In order to achieve the above, we selected the Play framework because of numerous advantages that it has in this context. First, in order to establish a lightweight service layer, Play provides an easy to use Scala framework for implementing RESTful services. Second, it utilizes a container-less deployment model, whereby the machine it is deployed on simply requires a JVM and the Play framework will manage HTTP requests that are received. Additionally, this also helps with horizontal scaling of the application if larger deployments are required. Finally, Play also provided a number of “templates” that provide much of the structure required for our use case. For instance, the “modern-web-template” provides HTML5, AngularJS, a CoffeeScript implementation, a REST Service layer and a data access layer through MongoDB. Our team removed the CoffeeScript implementation and used a more common MongoDB driver to further simplify the environment. To make the deployment easier for the project, our team utilized MongoLab, which provides a free hosted development environment for MongoDB.

Below is a diagram showing the various components of the architecture described above. As is shown, a user can access the solution through an HTML page. This provides simplistic search, create and update capabilities. These HTML pages then communicate with a REST service layer written in Scala. While the deployed pages communicate with the REST service layer, any other user who wished to write their own application could communicate with these same services. From this service layer, the database operations are handled via a MongoDB driver. MongoDB provides a noSQL option, which enables the data structure to more easily grow to accommodate new or changed malware scanners, methods for identification or the inclusion of additional attributes that are relevant without having to change the interfaces to the database or REST services.