# Executive Summary of

# Software Patch Management

# Web Application

**Sponsored by: Coupa Software**

*Team members:*

Amruta Gadgil

Fei Hao

Pramothini Dhandapany

*Faculty advisor:*

Patrick Tague ([tague@cmu.edu](mailto:tague@cmu.edu))

December 11, 2015

Patching is a critical task in IT infrastructure maintenance. New security vulnerabilities get released almost every day. In practice, most of the time software patches are applied manually and when there are hundreds of patches available for system assets, it becomes crucial to know the priority of patches. Many times, the priority decision is not obvious to make. CVSS (Common Vulnerability Scoring System) score, provided by CVE (Common Vulnerabilities and Exposures), is accepted as a standard mechanism to assess the vulnerabilities by a wide range of organizations and companies.

However, the CVSS score is not sufficient to prioritize patches because it does not take into account the environmental impact, which is the importance of host machine (asset) on which the patch will be applied. Environmental impact may include factors like the location of host machine in the network, the services running on the host machines and backup availability.

In this project, besides CVSS score, we introduced the environmental impact, in the form of asset importance rating, into determining the business risk of software vulnerabilities. Based on this design, we implemented a web application that facilitates users to view, update and manage software vulnerabilities. It will take in a vulnerability report file, pull a complete system inventory, and show the vulnerabilities and assets dynamically on a responsive web interface. Moreover, the web application allows system administrators to assign and change asset importance rating according to the environmental impact of assets, which will update the priority of vulnerabilities.

Since the business risk of vulnerabilities is a combination of both severity and environmental impact, our web application will generate a more accurate vulnerabilities priority report for users to refer to when applying patches. In the web application, the vulnerabilities could be exported and downloaded as standard, processable format, such as csv or json.

Beside these functionalities, we also designed the web interface to be more user friendly by including a dashboard, distribution modification slider, sorting, searching, and so on.

For the technology stack, we used Django-Python framework for web application development, PostgreSQL for backend database, REST API for database connector, Bootstrap for UI, and JavaScript plugins like jQuery for dynamic contents.

The code could be found on Github at https://github.com/Pramothini/webapp. The web application is hosted on heroku, and it can be accessed by http://software-patch-management.herokuapp.com.