RAC System Security Plan

[Company name] | [Company address]

Systems III

Philip Dumaresq

Fall 2017

Contents

[Introduction 1](#_Toc499713523)

[User Authentication 2](#_Toc499713524)

[User Authorization 2](#_Toc499713525)

[Encryption 2](#_Toc499713526)

[Confidentiality of data 2](#_Toc499713527)

[Use of SSL 2](#_Toc499713528)

[Backup of data 3](#_Toc499713529)

[Privacy Considerations 3](#_Toc499713530)

[Database security 3](#_Toc499713531)

[Directory path Manipulation 3](#_Toc499713532)

[Injection attacks 3](#_Toc499713533)

[Input validation 3](#_Toc499713534)

[Testing techniques 3](#_Toc499713535)

[Conclusion 3](#_Toc499713536)

# Introduction

This document will discuss the security testing that bust be done on the RAC system for the Development Project to ensure that the system is secure. The report will cover user authentication, user authorization, encryption, confidentiality of data, use of SSL on the server, privacy, database security, directory manipulation, injection attacks, and testing techniques. Establishing a security test plan for the RAC system will make the system more secure for it’s users.

# User Authentication

In the RAC system, we have users who must create accounts and log into the system. These users must be authenticated in two different ways. Some types of users in our system will be logging into the system through our own database of users, and these are users that we must handle the authentication for. However, other types of users in the system such as the RAC advisor must still be authenticated through Active Directory, so we must make sure that our system is able to do Active directory authentication. Otherwise, we could have anyone logging into our system, which would allows things like bots to be making RAC requests, which would ruin the integrity of our system and our data.

# User Authorization

There are multiple different levels of users in this system, each of which has access to a unique set of pages. The candidate, RAC advisor and Content Specialist have very little overlap in what they’re able to access, so we don’t want one user being able to modify the path in the URL and suddenly being able to access the RAC advisor’s home page and modify their data so that they’ve suddenly passed the system. If a candidate had access to the RAC advisor’s data, then they could also look at documents that other Candidates have submitted, and included in the possible documents is ministry transcripts. This is protected B information, so there needs to be no way any unauthorized user can access this data. The same goes for the RAC advisor, if someone managed to gain access to their account, we wouldn’t want them being able to access the Candidate’s self evaluations and manipulating their data, user authorization in this system is very important.

# Encryption

There needs to be encryption in the system between the user and the server. Everything should be done over HTTPS because the candidate will be sending personal information to the server. This includes uploading transcripts as documents, which as mentioned is considered Protected B information, so no one listening over a network should be able to grab that file as it’s going through. They also need to be able to protect the personal information of a user, such as their address or how you can contact them. This information could be used to hook candidates into phishing scams. We also need to be sending passwords encrypted to the server, so they won’t be sent in raw text, they need to be encrypted so that the passwords are safe, and people can’t access user accounts.

# Confidentiality and privacy

There is lots of data that will be stored in our databases that is confidential information. Things such as high school or college transcripts is considered Protected information according to the government, so we need to be able to assure our users their information will be 100% confidential. There will need to be confidentiality of data so that user’s files won’t be accessible and only the RAC advisor will ever see the files they’ve uploaded to the system. User privacy is also a big concern since we don’t their usage necessarily getting spread out.

# Use of SSL

SSL should absolutely be used to pass data to our sever. We’re going to be sending government protected information to our server, so HTTPS will be the protocol we should be using for this. We also don’t want to be sending things like emails, phone numbers and passwords unencrypted over our network, so SSL should absolutely be something that we enable on our server to be used. Without the use of SSL, documents and data would be sent to the server entirely unencrypted, which means anyone listening over the network could access this information.

# Backup of data

A lot of the data that we are storing in our system is going to be something that needs to be preserved over long periods of time because candidates need to use this a source for their education. Because of this, if any data got lost, then some users could wind up in a lot of trouble having no record of going through this system. Because of this, there should be backups of data on different physical machines that are taken every now and then to ensure that users don’t lose their progress through the system, or lose the fact that they ever existed in the system.

# Database security

Database security is a huge concern of ours. A lot of possible attacks will be right against the database itself, without going through the front end of our system, so we need to ensure that the database is very secured. The database will contain lots of sensitive information and personal information, so we need to ensure that the database is protected and ensure that should someone gain access to it, they won’t have the authorization to be able to do anything to the database.

# Directory path Manipulation

Path truncation is something that won’t be of a concern in this system. With MVC it doesn’t show the file name as a path of the path in the URL, so removing that to access directories won’t be an issue. The .NET framework also handles requests trying to traverse through directories, so reverse directory transversal isn’t something that we will have to handle ourselves because .NET will make sure that things can’t be accessed.

# Injection attacks

Cross-site scripting and SQL injection will be something that we’ll need to make sure to protect against. The .NET framework already handles sanitizing input before we ever do anything, so it will take care of some of the work for us, however we will still need to make sure that raw SQL will never get sent to our database.

# Conclusion

In conclusion, we need to make sure to implement most different types of security. We need authentication and authorization in the logging into the system and to ensure that users can’t access another user’s information. We need to have encryption when passing data to or from the server because the system will contain lots of confidential and private information about the candidates for the RAC system. We also need to make sure to have a secured and backed up database since an attack on our database would be detrimental to our system’s integrity and to the candidates who rely on their existence in our system.