NoSQL databases have been gaining popularity over the last few years. Such big companies as Expedia, Shutterfly, MetLife, and Forbes use NoSQL databases to manage data on different projects. These databases can contain a variety of information ranging from nonproprietary data to personally identifiable information like social security numbers and run the risk of cyber intrusion.

The goal of this thesis is to thoroughly explain a method of Incidence Response with MongoDB, a NoSQL database provider. The first part of this study was to research MongoDB vulnerabilities and produce an audit log that could be analyzed. The next part of the method involves a new self-built software tool that analyzes and parses MongoDB audit log's and displays the data on an html application. Within the application there are different pages that were created to show possible intrusions and activities that took place on the instance of MongoDB. The created tool also allows the user to conduct multilevel search queries on the audit log data. The results of this method proved to speed up the time it would've taken to manually analyze an audit log and to draw conclusions of what had taken place on a MongoDB instance.

While building the tool for this thesis, a major focus was on the advanced search page and optimization of processing the user's queries. The search algorithm for this page was made to transform a user's query into postfix format (ex. cond1 cond2 op1) to allow for linear processing. The algorithm would then scan through an audit log's data once and evaluate the query against each record of the log. This method was later improved upon by introducing short circuiting and Javascript loop optimization. To capture the queries execution times, complexity length, and log file size, a PHP mechanism was included in the code. This allowed for the performance results to be adequately analyzed and plotted on charts. This proved to be very useful in drawing conclusions about the search algorithm and in finding ways to optimize it. The results of the analysis were that as the size of the audit log increases, so will the execution times. Also, by adding short circuiting, the search processing was sped up by more than 50% in most queries.