Phishing Email Detection

The way we determined whether or not an email is a phishing email was by comparing the contents of the email address with a list of addresses which are known to be or likely are spam addresses. We did a similar thing with the body of the email. We kept a list of common words/phrases found in phishing emails and compared them with the contents of the given email. The likelihood that an email is a phishing email is based on whether or not the email was in the spam email list and how many matches there were between the text body and the list.

This method can be improved by allowing the user to tell the system whether or not the email actually was a phishing email. If the user says that it is, some of the words/phrases from said email, along with the senders email address, would be added to their respective lists. This way there will be more data to check against for the following email submitted. Eventually the program should become fairly competent at correctly identifying phishing emails. It could also be useful to keep a list of good words/phrases that help you determine that a specific email is not a phishing attempt and is actually a legitimate email.

Malicious Query Measure

The way we determined if the first query was malicious was by implementing a check to see if the query contained anything that would not be considered an item (i.e checked to see if common SQL injection attacks are used, such as ‘1=1’ or ‘DROP’) if the query contained anything of the like we reported that a high probability of it being a SQL attack. For the second query we did something similar. A username and password should not contain common SQL injection patterns so we would raise a high probability of it being an attack if the query contained those common patterns.

This approach could be improved by adding to the list of possible SQL injection techniques. Currently, the scanners will only check for approximately 15 different attack techniques. In addition, the current implementation would reject users that have passwords that contain common SQL injection characters. We could improve this by creating more specific checks.