Email Confidentiality Scanner
Brandon Dixon, Nicholas Schillaci, Ryan Hudson
Department of Computer Science, Rowan University
Faculty Advisor – Jack Myers

Introduction
Organizations of all types require email security in order to keep users/employees from transmitting confidential business information.

Tigerscan acts as an email security scanner which scans outgoing email content for confidential information and scores the email appropriately. Terms and scores can be managed by a registered administrator of the system.

Description
Tigerscan indexes and searches the body text of an email for confidential terms defined by the company. A confidentiality score is calculated based off the number of sensitive terms or patterns that were detected, if the score is above a set threshold, then the email will be rejected as too confidential to send. The score is determined by a heavily modified version of the Bayes Naïve Spam Filtering Algorithm while full-text indexing and searching is handled by Apache Lucene.

Data Flow
● One or more text files are added to the GUI
● When hitting the scan button, Lucene stores the terms in the text body to a Lucene directory.
● All terms are stemmed down to their root forms (i.e. running becomes run) and any punctuation or white space is removed.
● Next it pulls the confidential terms from a database, stems the terms to their root form, and queries the indexed terms.
● If one or more matches is found, the algorithm determines the probability that this email is confidential.
● A confidentiality threshold is to be set by an administrator to dictate what an email that should or should not be sent.

The program can also be run in headless mode from command line, bypassing the management GUI and streamlining the process to calculate the confidentiality of a given email. This can be run by a simple plugin written for an email client, which can then receive the email and flag it.

Data Flow Diagram (Package)

Probability of Confidentiality
The system scores emails based on the probability that they are confidential. The algorithm determines the score based on individual word hits when compared to the database, then combines these scores to give an overall score. This is a modified version of Bayes Naïve Spam Filtering Algorithm while full-text indexing and searching is handled by Apache Lucene.

Acknowledgements
Matthew Paule, Zackary Flake, and Samantha Brech for their contributions to the development.

Introduction
Organizations of all types require email security in order to keep users/employees from transmitting confidential business information.

Tigerscan acts as an email security scanner which scans outgoing email content for confidential information and scores the email appropriately. Terms and scores can be managed by a registered administrator of the system.

Description
Tigerscan indexes and searches the body text of an email for confidential terms defined by the company. A confidentiality score is calculated based off the number of sensitive terms or patterns that were detected, if the score is above a set threshold, then the email will be rejected as too confidential to send. The score is determined by a heavily modified version of the Bayes Naïve Spam Filtering Algorithm while full-text indexing and searching is handled by Apache Lucene.

Data Flow
● One or more text files are added to the GUI
● When hitting the scan button, Lucene stores the terms in the text body to a Lucene directory.
● All terms are stemmed down to their root forms (i.e. running becomes run) and any punctuation or white space is removed.
● Next it pulls the confidential terms from a database, stems the terms to their root form, and queries the indexed terms.
● If one or more matches is found, the algorithm determines the probability that this email is confidential.
● A confidentiality threshold is to be set by an administrator to dictate what an email that should or should not be sent.

The program can also be run in headless mode from command line, bypassing the management GUI and streamlining the process to calculate the confidentiality of a given email. This can be run by a simple plugin written for an email client, which can then receive the email and flag it.

Probability of Confidentiality
The system scores emails based on the probability that they are confidential. The algorithm determines the score based on individual word hits when compared to the database, then combines these scores to give an overall score. This is a modified version of Bayes Naïve Spam Filtering Algorithm, where certain variables have been created or modified to make the algorithm applicable in this context. The program saves data from each scan to improve the accuracy of future scans. The formulae shown below describe the following probabilities:
● probability that a given email is confidential
● probability an email is confidential given a word hit
● probability any given word is in a confidential email
● probability any email is confidential.

\[
P(C) = \frac{\prod_{i=0}^{n} P(C|W_i)}{P(W) + \prod_{i=0}^{n} (1 - P(C|W_i))}
\]

\[
P(C|W) = \frac{P(W|C) \cdot P(C)}{P(W|C) \cdot P(C) + P(W|\overline{C}) \cdot P(\overline{C})}
\]

\[
P(W|C) = \frac{nE}{nE + nE_{Not} + nE - 1 + nE_{Not}}
\]

\[
P(\text{Any is } C) = \frac{\sum_{i=0}^{n} P(E_i|C)}{n}
\]