IOWA STATE UNIVERSITY
Digital Repository

Retrospective Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

2007

Detection of encrypted streams for egress monitoring

Paras Malhotra
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

 Part of the Computer Sciences Commons

Recommended Citation
Malhotra, Paras, "Detection of encrypted streams for egress monitoring" (2007). *Retrospective Theses and Dissertations*. 14632.
https://lib.dr.iastate.edu/rtd/14632

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

www.manaraa.com

Detection of encrypted streams for egress monitoring

by

Paras Malhotra

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-majors: Information Assurance; Computer Engineering

Program of Study Committee:
Thomas E. Daniels, Major Professor
Douglas W. Jacobson
Douglas D. Gemmill

Iowa State University

Ames, Iowa

2007

www.manaraa.com

UMI Number: 1447482

UMI

UMI Microform 1447482

Copyright 2008 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

www.manaraa.com

TABLE OF CONTENTS

www.manaraa.com

www.manaraa.com

LIST OF FIGURES

www.manaraa.com

ABSTRACT

Leakage of confidential information from an organization's networks has become a big threat to its information security. Egress monitoring and filtering have thus become popular for detecting such security breaches. Egress monitoring tools scan outgoing packets for keywords or their combinations present in the confidential documents. These content filtering techniques however fail when the data is encrypted.

The solution proposed in this thesis is simple yet an effective approach to prevent information leakage when the data is encrypted. We assume that a policy is in place which disallows encrypted content from specific hosts, ports and applications and wish to detect any violations to this policy. This work aims at analyzing encrypted and unencrypted traffic flows across a gateway and detecting unauthorized encrypted traffic flows. The work discusses a low level approach to detect encryption, based on entropy calculation and packet analysis. The technique is based on the fact that encrypted data consists of a random distribution of symbols whose entropy is expected to be quite high as compared to an unencrypted file. Techniques to differentiate between encrypted and high entropy compressed traffic are also discussed. This thesis implements and compares statistical methods for a fast online detection of encrypted traffic from all the other unencrypted traffic flowing across a network.

www.manaraa.com

CHAPTER 1. INTRODUCTION

A company's employees are a major threat to a company's confidential data which may consists of design documents, financial information, code for a new software etc. A survey conducted in 2006 by Computer Security Institute/ FBI computer Crime and Security puts leakage of private information and internal security as the two biggest concerns of companies [1].

Most egress filtering techniques aim to filter out IP addresses and ports or are content based, i.e. they parse the text in the packets leaving an organization's network and block them if they violate a rule. These techniques fail when the data leaving the network is encrypted. In this dissertation this problem is addressed, by developing methods which carry out a series of statistical tests on the outgoing data to detect encrypted traffic. We also differentiate encrypted traffic streams from compressed traffic streams which appears quite random. The suspected encrypted traffic streams could then either be quarantined or completely blocked from leaving the company's gateway by dropping them and logging an alert.

The proposed approach offers a unique solution which does not depend on specific applications or contents of various files. Encrypted stream is detected by calculating the entropy and randomness of the outgoing stream, and determining the variation from the expected values. The entropy or randomness of an encrypted stream is quite high and we make use of the low entropy present in an unencrypted stream by a series of statistical tests, which can be implemented online on a company's networks. The proposed technique achieves a low false positive rate and a negligible false negative rate when detecting encryption.

www.manaraa.com

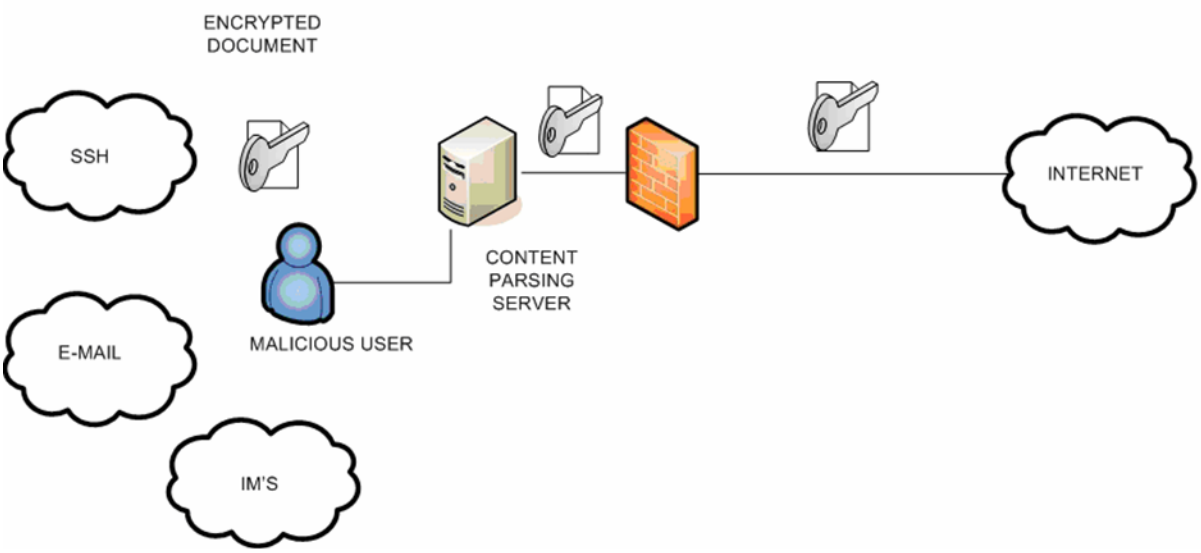

Figure 1: Threat Model

1.1 The Threat Model

The assumptions we make in this work is based on a threat model which is both simple and realistic. The threat model we consider consists of an experienced user familiar with the concepts of encryption and having the knowledge of content parsing egress filters which may or may not be configured on an organization's gateway. He is capable of employing any or all of the below methods to evade detection of leakage of confidential information. Some of the methods are:

1) The malicious employee removes the file headers of files such as PDF, JPEG etc to render the content scanning egress tools useless which parse outgoing files for file extensions.
2) The malicious employee tries to encrypt the document using a good encryption scheme such as AES, so as to evade the egress filters.
3) The malicious employee may encrypt and/or remove the headers of the encrypted file to

www.manaraa.com

evade detection.

1.2 Thesis Structure

The rest of the thesis is structured as follows. The thesis first discusses ‘related work’ in Chapter 2. Chapter 3 describes general concepts of entropy and randomness in compresses and encrypted file streams. Chapter 4 provides an overview of various statistical tests that were implemented. The basic flow diagram underlining our approach to differentiate encrypted and unencrypted streams is discussed in Chapter 5. This chapter also discusses the experimental design and the implemented tests and the experimental results we obtained to support our approach. Chapter 6 discusses future work and possible extension of this research and we draw conclusions in Chapter 7.

www.manaraa.com

CHAPTER 2. RELATED WORK

Egress filtering techniques have been around for quite some time and they deal with filtering outbound spam, malicious traffic and confidential information. Filtering outbound traffic is often a common solution to mitigate Denial of Service (DOS) attacks [2]. Almost all of the present techniques try to filter traffic by matching signatures already present in the database.

According to [3], almost all corporate information is present in an electronic form, and is easily accessible by any employee. Thus there is a high risk involved if the sensitive data leaks from the company due to an employee mistake or incase of a deliberate disclosure. The solution proposed, creates a fingerprint of sensitive documents by taking in to account words, sentences or data fields in a document and analyzing the data fields using a combination of rules and categories for an exact or a partial matching. The technique however requires maintaining a large fingerprint database which can become extremely difficult to manage for large organizations. It also doesn't offer a real time solution as content parsing and pattern matching is time consuming.

The paper by R. Clayton [2], deals with extrusion detection and monitoring the outbound emails from a company's networks. The paper deals with automatic processing of email logs at a server of an ISP that is used to send emails through SMTP. Their technique tries to detect bulk emails which are returned since they were sent to invalid or invented addresses. The mechanism uses protocol details such as the 'HELO' command and the 'Mail From' fields to identify the sending host and the email id of the sender. It also considers the IP addresses of the sender in the header. Some of the heuristics deal with spotting malformed

www.manaraa.com

'HELO' strings and detecting email loops. A related concept of outbound filtering is discussed in this work, whereas our aim is to detect outbound encrypted emails containing confidential information. This is achieved by calculating statistics of an outbound packet instead of parsing it and filtering it for sensitive data. The technique can also be used in extrusion detection in a possible scenario where a trojan/backdoor installed by a malicious outsider or a competitor on a rival's machine/s tries to encrypt confidential data and send back the information to itself. Such encrypted content would be blocked before it can leave the company's gateways.

A lot of work has been done in traffic analysis and analyzing anonymity systems [4, 5, 6]. Liberatore, et al. [4] looks at encrypted SSH tunnels and anonymous communication systems which aim to hide the true destination and source of HTTP requests. Their technique involves analyzing encrypted traffic by comparing it with traffic whose profiles are known in advance. Sun, et al. [5] discuss a similar approach for statistical identification of encrypted web browsing traffic where instead of just looking at the packet lengths they consider HTTP object counts and sizes. A signature of encrypted web pages is maintained consisting of the number of objects requested by a web browser in downloading a web page and the object lengths. Both the techniques use classification algorithms such as Bayesian classification algorithm, for matching and identifying the test corpus libraries of known encrypted traffic. The above techniques do well in inferring encrypted web traffic since they have nearly static sizes, but it is infeasible to use them to determine encrypted files which may be in different formats or of different sizes and could be encrypted using any key. Since files can be encrypted using any key, maintaining a profile of all the encrypted confidential files is nearly impossible. The technique involves the comparison of packet lengths, which can be altered

www.manaraa.com

by padding, thus rendering the identification difficult. Dang, et al. [6] analyze network traffic for detecting self-decrypting exploit code. They propose a hybrid approach that consists of static analysis and emulated instruction execution techniques to detect the decryption routine typical to most encrypted exploit code. The proposed method tries to locate the position where the decryption routine starts from. They then use backward data flow analysis to find the decryption instructions and then attempts to identify the self modifying decryption routines by an emulated execution of already recovered decryption instructions. Our aim however is to statistically determine the existence of encrypted traffic when no decryption routine is present.

A white paper by McAfee [7] describes one of their Network IPS, which detects SSL encrypted attacks. The IPS tries to mitigate such attacks by inspecting the encrypted traffic by storing the private key of the SSL server on sensors deployed in the network. The sensors decrypt the flowing traffic using the session keys derived from the private key. Once decrypted the packet is processed and forwarded to signature matching engines, to detect a possible attack.

Encrypted data is more uniformly distributed than unencrypted data, and this is used to locate cryptographic keys hidden in large amounts of data [8] such as on a hard disk or tape. Shamir et al. [8] discusses algebraic attacks in locating secret RSA keys in long streams of data, and uses statistical methods to identify cryptography keys embedded in large data streams. They propose a method to divide the entire dataset in to small blocks and measuring the entropy of each block by counting the unique byte values of data in order to determine high entropy regions which would indicate locations where a cryptographic key may be present. This method of frequency calculation to determine entropy works well when the

www.manaraa.com

surrounding data has low entropy such as in ‘text’ but fails when the encrypted data is interspersed with other high entropy data such as compressed text, or file formats such as PDF, and JPEG. More specific statistical tests are thus required to distinguish the high entropy data from encrypted data.

www.manaraa.com

CHAPTER 3. CONCEPTS OF COMPRESSION AND ENCRYPTION

This chapter gives a brief overview of egress filtering; it's importance and describes the concepts of randomness and entropy in relation with compression and encryption algorithms. It also gives an insight on how a compressed and encrypted data stream is supposed to look like when viewed as a stream of bytes.

3.1 Egress Filtering

Egress filtering means filtering outbound traffic as it leaves a computer or a network for a destination address that is outside a company's networks. Screening egress traffic is essential as it could contain confidential information, malicious traffic trying to attack computers on the outside or even traffic from a bot spamming the internet. It may even be a malicious user or a company's employee trying to send some confidential information to the company's rivals either through emails or by file sharing using P2P applications or through instant messengers. The above mentioned scenarios are a big cause of concern for organizations and leads to losses and embarrassment when confidential data gets leaked. The outbound data should therefore be given the same importance as the inbound data and should be closely monitored for any classified information and malicious activity. This work discusses differentiating encrypted outbound traffic stream from all the other traffic streams which may contain confidential information of an organization such as financial documents, software designs, code etc.

www.manaraa.com

3.2 Entropy and Randomness

A related concept to randomness in an information/data stream is the concept of entropy or uncertainty in that stream. According to Information Theory [9], entropy is described as the measurement of the unpredictability of data in the information source, in other words it means the measurement of randomness in data. The entropy of the data increases with greater randomness in the data in the information source. It can also be interpreted as the shortest message in bits that needs to be transmitted in order to express the correct meaning of the message. A compressed message therefore has a high degree of entropy and appears random.

Shannon [9] gives the following equation to calculate the entropy of an information source involving the discrete variable 'X' as:

$$H(x) = \sum_{i=0}^{n} p(x_i) \log_2 (1/\, p(x_i)) \qquad (1)$$

$$= -\sum_{i=0}^{n} p(x_i) \log_2\, p(x_i) \qquad (2)$$

x: A random variable

p(x): The probability density function of x.

The maximum entropy from the above equation can only be obtained when the probability of occurrence of each event is equally likely. This means that if any one of the event is more likely to occur than the other, the overall entropy of the system decreases. Thus for a system to have a high entropy, the occurrence of each of its events should be as random as possible, in other words the occurrence of an event should be unpredictable.

www.manaraa.com

3.3 Encryption and randomness

Randomness plays an important role in cryptography. Encrypted data should appear to be a random string of characters such that no meaningful information can be inferred by looking at the data or by carrying out any statistical tests on it. This means that the encrypted data should not show any resemblance to the plain text. An ideal encryption algorithm thus would create an encrypted stream which would appear as a random noise to the observer. This unpredictability of an encrypted stream of data is desirable as it is difficult to establish a predictable pattern in the stream. The characters in the encrypted stream should therefore be independent of any of the previous characters present in the stream. Therefore if there are two random variables X and Y then the condition that X does not reveal any information about Y is only possible when X and Y are statistically independent. In other words [9], the mutual information I(X, Y) which can be defined as the dependence of one variable on the other should be much low. Therefore as discussed above an ideal encrypted stream should have an equal probability distribution of the characters present in the stream. The entropy of encrypted data is considered to be high as there is a lot of uncertainty present in the information stream as to what the original message is. In other words the encrypted message has higher entropy than the original message.

3.4 Compression Algorithms and Randomness

Most of the present compression formats such as zip, gzip, bzip and rar make use of compression algorithms such as LZ77 [12] , Deflate, and encoding techniques which include run-length coding and Huffman coding [12]. The basic aim of any compression algorithm is

www.manaraa.com

to reduce the size of the data by decreasing the redundancy present in the data. The compressed data is thus devoid of any duplicate patterns present in the original data. A compressed data can be considered a fairly random stream of bytes. According to the Kolmogorov-Chaitin complexity [10-11] a string is considered to be random if there exists no other shorter description of the string. Since each byte in the compressed data contains more information than the original uncompressed data, the entropy of this compressed data is greater than its uncompressed version. Despite the low redundancy and high entropy of a compressed stream, there exists a small difference in the entropy levels of an encrypted stream and a compressed stream. Compression algorithms such as Gzip [12] work by replacing the repeated sequences of characters in the data by a pointer to the previous repeated sequence in the form of a distance pair and a length field which gives the length of the repeated sequence. The maximum distance can be up to 32K bytes and the length up to 258 bytes. The match lengths and match distances are further compressed using Huffman trees [12], which are stored in the compressed form at the start of each block. Other algorithms such as Bzip [13] use a variant of run-length encoding for compression by replacing the repeated sequences by a single repeated character and a count. The implementation of the compression algorithms gives an idea that the compressed stream shares certain characteristics with the original uncompressed stream. Therefore a compressed stream cannot be considered a truly random stream even though the difference between their entropy and the encrypted stream is not significant. The resemblance that a compressed data shows to its uncompressed version can be used to differentiate between the compressed and encrypted data.

www.manaraa.com

CHAPTER 4. OVERVIEW OF STATISTICAL TESTS

This chapter describes some of the statistical tests used to calculate randomness and measure entropy in data streams. All of these tests have been used in the course of this research and at times two or more of these tests have been used together to draw conclusions. Some of the tests described below were instrumental in this research in differentiating encrypted streams from the other unencrypted streams. The rest of the tests although did not give the desired results, they gave a direction to keep trying different tests to eventually achieve the research goals. A brief overview of some of the statistical tests which were used to check randomness and calculate entropy in streams is as follows:

4.1 Discrete Runs Test

This test is a type of an empirical test which helps determine whether a given sequence of numbers has statistical properties similar to a sequence whose elements have a uniform probability distribution [14]. In the test consecutive elements in the stream of data are parsed till they are monotonically increasing. The length of this run is calculated once the sequence stops increasing.

The input to the test [14] is a stream of integers $X = X_1, X_2, X_3...$, where $X_i \varepsilon Z^d$ for all X_i. The stream of integers is supposed to contain at least one pair (X_i, X_{i+1}) so that $X_{i+1} < X_i$. The run length L_x of X is a positive integer n such that

$$X_1 <= X_2 <= X_3.....< X_n > X_{n+1}$$

www.manaraa.com

When more than one run is to be calculated the next sequence starts from X_{n+2} to ensure that the calculations are independent of each other. These observed run lengths are compared with the expected run length probabilities for a stream of uniform distributed data given by the ‘runs test probabilities table’ [14]. A Chi Square test can then be used to calculate the deviation of the observed runs from the expected runs. A high deviation of the observed runs from the expected runs would mean a non uniform distribution in the given sequence of data and hence non randomness.

4.2 Phase Space Analysis

This technique differs from the above techniques in testing a data stream for randomness as this test deals with looking at the data stream in a 'n' dimensional space. As described by [15], phase space can be defined as a 'n' dimensional space that can be used to depict the state of an ‘n’ variable system. The shape thus generated by plotting the data in a phase space shows dependencies between successive elements in the sequence of data.

A three dimensional plot of a one dimensional input can be generated by a technique known as delayed coordinates [15]. For a stream of data with the content 'C' present in the stream, the points (x, y, z) in the phase space can be calculated using the following equations:

$$X[i] = C[n-2] - C[n-3] \quad (3)$$

$$Y[i] = C[n-1] - C[n-2] \quad (4)$$

$$Z[i] = C[n] - C[n-1] \quad (5)$$

www.manaraa.com

A three dimensional phase plot of a high entropy random data would reveal no structures or patterns. It would appear as a uniform distribution of points in the phase space forming no distinguishable patterns. A low entropy data stream on the other hand would show a distinctive correlation pattern since the successive values in the data are not independent of each other.

4.3 Auto Correlation Test

This is the cross correlation test of the data stream with itself which can be used to reveal a cyclic pattern in the stream. This test can be used to detect the periodic nature of the examined data. The randomness of the data is checked by calculating autocorrelation for the values of the data stream at different time lags [16]. The autocorrelations should be much close to zero if the data stream is truly random, otherwise one or more correlations would show high variations from 0. The autocorrelation is calculated as follows [16]:

$$\boldsymbol{R_h = C_h / C_o} \qquad (6)$$

$\boldsymbol{C_h}$: Auto covariance function

$\boldsymbol{C_o}$: Variance function

R_h: lies between -1 and +1

C_h can be calculated by the below equation:

$$C_h = \frac{1}{N}\sum_{t=1}^{N-h}(Y_t - \bar{Y})(Y_{t+h} - \bar{Y}) \qquad (7)$$

www.manaraa.com

N: Total length of the data stream

h: Correlation lag

$\overline{Y}$: Mean of N values

and C_0 is the variance function calculated by the equation:

$$C_0 = \frac{\sum_{t=1}^{N} (Y_t - \overline{Y})^2}{N} \quad (8)$$

4.4 Kolmogorov-Smirnov test (KS-test)

This test is an alternative to the chi-squared goodness of fit test [17]. Similar to chi-square, it can be used to test the hypothesis that a given sample follows a hypothesized distribution. It is more sensitive to small variations between the observed and the hypothesized distributions. The test is computed as follows [17]:

a) S(x), the empirical cumulative distribution function is computed from a sample of N observations.

b) G(x), the theoretical cumulative distribution function is computed by considering that the null hypothesis is true.

c) $C=\max_i |G(x_i) - S(x_i)|$ is calculated for each of the N sample points.

d) A confidence value (α) is chosen, and the hypothesis is rejected if the tabulated critical value at that level of significance is greater than the value of C.

www.manaraa.com

4.5 Arithmetic Mean

This test calculates the average mean of the data present in the traffic stream. The traffic stream is made up of symbols whose values range from 0 to 255. The arithmetic mean of a stream is calculated as:

$$A = (C1+C2+C3+.....+Cn)/n \quad (9)$$

C: ASCII value of the characters in the stream.

N: Total no. of characters in the stream.

Arithmetic mean of an encrypted stream should be much close to the value '127.5' since it's the average value and there's an equal probability of the occurrence of each character in the stream. On the other hand for an unencrypted stream, the arithmetic mean should show a large deviation from the value '127.5'. This test can be implemented online easily without many resources and can quickly screen the low entropy data streams from the high entropy streams.

4.6 Information Entropy

Information entropy is the measure of the density [9] of the information present in a data stream. In other words it's an indicator of the amount of information present in the stream. A high amount of information present indicates that the stream cannot be compressed too much without loosing any information present. Entropy is also the amount of uncertainty present in a stream; a completely random stream would thus have high entropy. Shannon [9] gives the equation for calculating the entropy of the data present in a file/stream as:

www.manaraa.com

$$H(x) = \sum_{i=0}^{n} p(x_i) \log_2 (1/\, p(x_i)) \qquad (10)$$

$$= -\sum_{i=0}^{n} p(x_i) \log_2\, p(x_i) \qquad (11)$$

X: Discrete random variable

p(x): Probability density function of X.

For a stream with a highly random distribution of characters the entropy should be close to 8bits per symbol. This means that even if the previous symbols are known, it's still impossible to predict the next symbol in the stream. This means that the stream has a high information density and is essentially random.

4.7 Chi-Square Test

This is one of the commonly used tests to check for randomness in a stream of data. This test calculates the degree to which 2 given samples of data differ from each other. In other words it is used to test whether a set of data follows a specific distribution. The Chi-Square test tells the degree of confidence one can have in rejecting or accepting a hypothesis. The following hypothesis is considered in this work:

P1= The data stream follows a particular distribution/The data stream is highly random.

P2= The data stream does not follow a particular distribution/The data stream is not random.

www.manaraa.com

The entire data stream is divided among k bins and the chi-square is calculated using the equation:

$$X^2 = \sum_{i=1}^{k} (O_i - E_i)^2 / E_i \qquad (12)$$

k: No. of bins

O_i: Observed frequency for the i^{th} bin.

E_i: Expected frequency for the i^{th} bin.

The chi square values are expressed as an [18] absolute number and a percentage which gives a measure of the frequency with which a truly random sequence exceeds this value by chance. As defined by [18] if the percentage 'p' is:

a) 1%>p or p>99% the stream is not random and the hypothesis P1 is rejected.

b) 1%< p <5% or 95%<p<99%, the sequence is “suspect” [18].

c) 90%< p <95% and 5%< p <10%, the sequence is “almost suspect” [18].

4.8 Index of Coincidence Test

The index of coincidence for a text [19] is the measure of the probability that two letters selected from the text are identical. Thus it's a statistical measure of the redundancy in a text. For a truly random, high entropy stream the index of coincidence is much close to 1. If the value shows a large deviation from 1, it indicates that the stream is highly structured and has low entropy. The Index of Coincidence can be calculated [19] as:

www.manaraa.com

$$\mathrm{IC} = \frac{\sum_{i=1}^{c} n_i (n_i - 1)}{N (N - 1) / C} \qquad (13)$$

N: The length of the text,

n_i: The frequency of the character i in the text,

C: The number of letters in the alphabet.

www.manaraa.com

CHAPTER 5. IMPLEMENTATION

This chapter first gives an overview of the basic approach taken to differentiate encrypted data streams from unencrypted streams. It then discusses the various statistical tests which were tried in order to obtain satisfactory results. Some of the tests did not perform as well as others but they gave us confidence that a similar approach would work if it was tweaked or more refined. A series of tests were performed on the test corpus consisting of varying file formats and lengths. The tests were evaluated and compared on the amount of data they required, the false positive and negative rates obtained and the ease with which they could be implemented. Finally we recommend some tests which gave suitable results and tried to minimize the false positive and false negative rates to make them feasible for use on a network as an online filter.

5.1 The basic flow diagram

The figure (2) below is a flow diagram which gives the fundamental steps of distinguishing encrypted data streams from other non encrypted streams. This process of differentiating encrypted streams can be broken down in to 4 steps as described below.

5.1.1 Checking for encryption ports and addresses

The outgoing data stream is intercepted by the filtering tool at a company's gateway and the source IP addresses and ports are checked. If the IP address and port matches the list of ports and addresses on which encryption is not allowed, the data stream is subjected to filtering tests, as described in the next steps.

www.manaraa.com

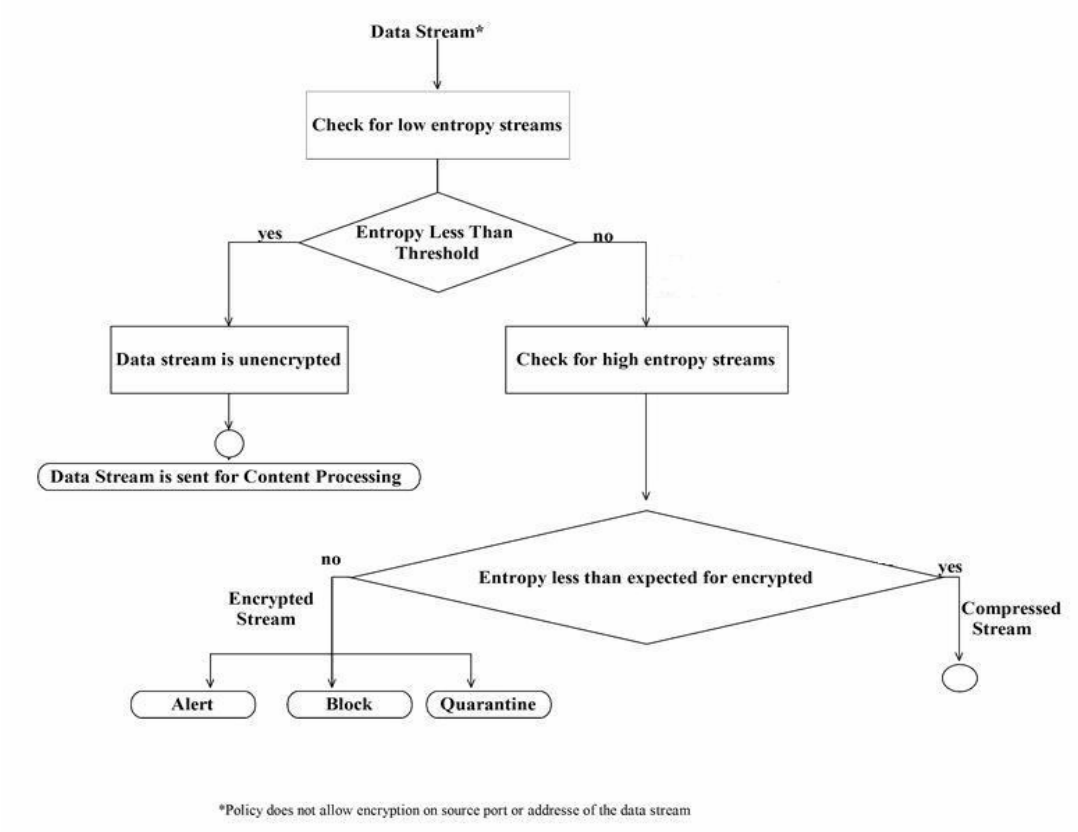

Figure 2: Basic Flow Diagram

5.1.2 Checking for low entropy streams

If the data stream is to be checked for encrypted traffic as determined above, the entropy of the incoming data stream is calculated. The entropy gives a measure of randomness present in the data stream. The first step therefore is to check for low entropy in the outgoing data stream. In this step we check whether the outgoing stream is a plain text or some data having high redundancy and less randomness. If the entropy of the stream is above a specified threshold, the stream has a high chance that it's encrypted and is subjected to further tests. The uncertainty in determining encryption is because the data stream may be compressed or it may be a high entropy file format which would also result in characteristics

www.manaraa.com

similar to an encrypted stream; therefore other tests are required to detect encryption with a high precision. If however the entropy of the stream is less than the threshold, the stream is considered to be unencrypted and is forwarded to the content filters to check for confidential information.

5.1.3 Checking for high entropy streams

If the data stream cannot be categorized as a low entropy stream above, then in order to categorize it as encrypted we need to make sure that the stream is not one of the other highly random data streams such as PDF, Jpeg etc or a compressed stream. In order to differentiate encrypted streams from other high entropy streams, further statistical tests are carried out. This differentiation however is not easy as most of the modern compression algorithms are capable of compressing data very well. The compressed stream therefore has less redundancy and some tests fail to differentiate encrypted streams from compressed streams. However as some of our initial results showed, a compressed stream does have certain characteristics which can be used to make the differentiation from highly random encrypted data. Some of the tests that were carried out proved to be quite useful and gave results which showed that an encrypted stream can be differentiated from high entropy streams without a high false positive and false negative rate. Later in the chapter we go in to the details of these tests, the observations made and the results obtained.

5.1.4 Blocking or Passing the stream

If the data stream shows a high deviation from the expected as determined in the previous step, then the null hypothesis that the stream is encrypted is rejected and the stream

www.manaraa.com

is forwarded for content processing to determine if it contains any sensitive information. Otherwise if the stream shows quite small deviation from the expected, the null hypothesis cannot be rejected and the data stream is then either blocked from leaving the gateway, sent for quarantine or is forwarded for content processing and an alert is logged. Figure 3 below shows a high level security model to counter the threat model shown in chapter 1. The model now has a statistical analysis server which intercepts all traffic going out of the company's networks. Statistical tests are carried out on the outbound traffic stream and if it matches the encrypted profile, they are blocked from leaving the networks and the incident is logged and an alert generated.

5.2 Test Corpus

The test corpus considered for testing; consisted of files in various formats such as DOC, JPEG, TXT, and PDF. Most of the sensitive/confidential documents exist in one of these formats; hence the four file types were used in the tests. As many as 100 files of various sizes ranging from 5KB to 5 MB were considered for carrying out the tests. The size range considered would encompass most of the company documents. Differentiating encrypted data from unencrypted is easier for big file sizes and a 5MB upper limit is sufficient to differentiate encryption from all the other data with low false positive and negative rates. The files were compressed using Gzip, Zip, Rar, and Bzip compression formats, which are some of the common compression formats being used on Windows and Linux based systems. The algorithm used to encrypt files was AES-256 bits, in Cipher Block Chaining mode which has been adopted as the encryption standard by the US government and is one of the default options for many encryption tools. We obtained similar results for

www.manaraa.com

files encrypted by DES in Cipher Block Chaining mode.

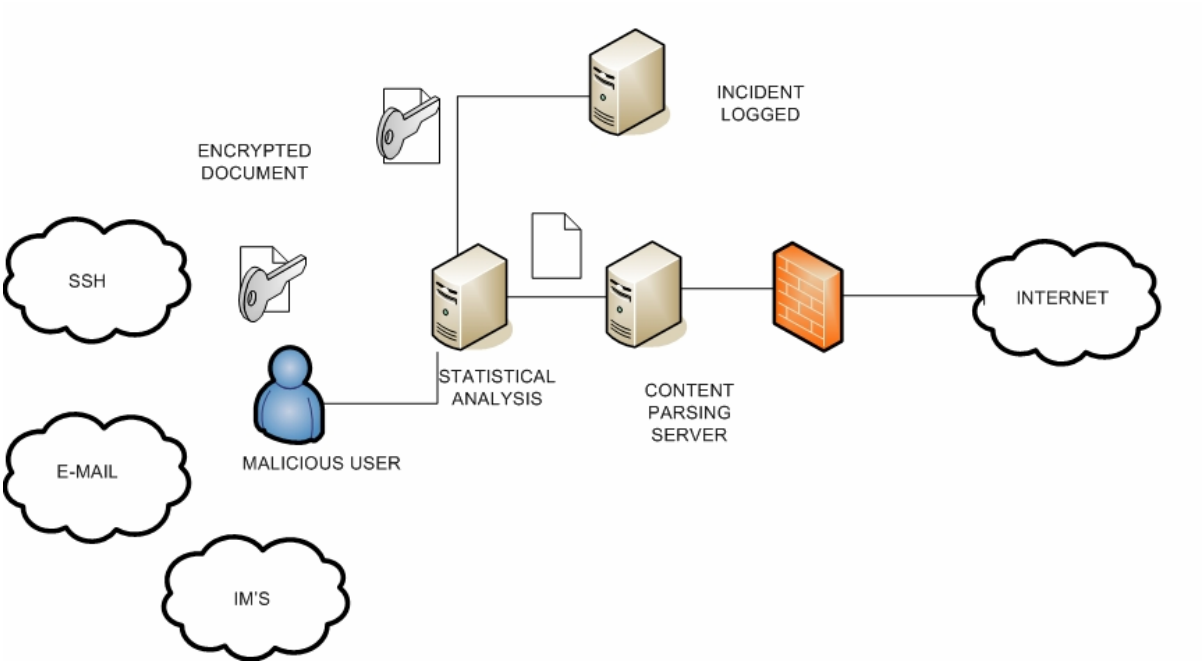

Figure 3: The Security Model

5.3 Assumptions

The initial step is to determine whether the outgoing traffic stream needs to be filtered for encrypted confidential content. As mentioned before in this chapter, an assumption is made that the organization has a rule or a policy in place to disallow encrypted content from specific ports or IP addresses. The outgoing packets are checked whether their source IP addresses or ports are those which need to be filtered for encrypted content. If the packet needs to be filtered, its data payload needs to be extracted from the Ethernet packet. An Ethernet frame consists of an Ethernet header followed by an IP header and a TCP header and finally the TCP data which is being transmitted. The data payload starts approximately

www.manaraa.com

after the initial 54bytes of the Ethernet frame, thus the initial headers are stripped off to obtain the data payload. Once the payload is extracted, it is stored in a buffer until we achieve minimum number of bytes required for the tests. The buffer sizes range from 1KB to 8 KB and since the maximum size of a data payload is around 1500bytes around six to seven packets are required at a minimum to make up a buffer. The outgoing packets can be stored in an array of buffers to be used for real time statistical tests. The next major steps required to differentiate between the encrypted and unencrypted traffic stream is to first filter out the low entropy stream such as payload in the form of Doc, Txt and other text format which have a relatively low entropy as compared to the encrypted traffic stream. If the traffic fails to be categorized as a low entropy stream, it needs to be checked for high entropy content such as compressed streams having high randomness comparable to that of encrypted streams.

5.4 The Initial Tests and Observations

Figure 2 shows the basic approach taken to differentiate between encrypted and other traffic leaving a company's networks. This section describes some of the initial tests that were carried out to implement the steps described above.

5.4.1 The Frequency Analysis

Frequency analysis of a data stream shows the distribution of symbols in the stream. Figures 4 - 6 below shows the frequency distribution of three data streams, a DOC file, the same file encrypted and compressed. The encrypted stream shows a much uniform distribution of symbols, looking at Figure 6 the encrypted stream looks like a uniform noise having no distinguishing pattern. Figure 4 on the other hand shows a frequency distribution

www.manaraa.com

of a DOC file. The figure clearly shows that the distribution is not uniform and this is not surprising as the distribution depicts low entropy and non random distribution characteristics of the English language. Some of the characters in the English language are more common than others and hence the irregular distribution. Figure 5 is a frequency distribution of a compressed DOC.

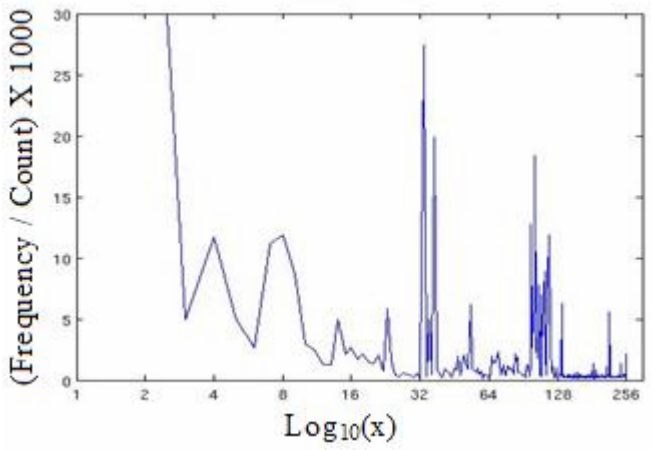

Figure 4: Frequency Distribution of Doc Stream.

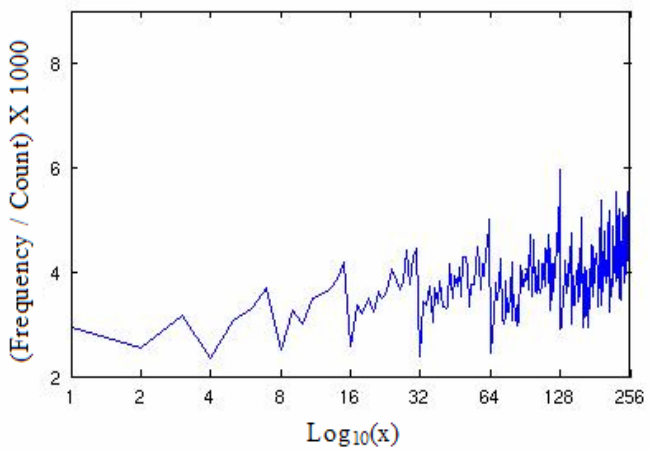

Figure 5: Frequency Distribution of Compressed Doc Stream.

www.manaraa.com

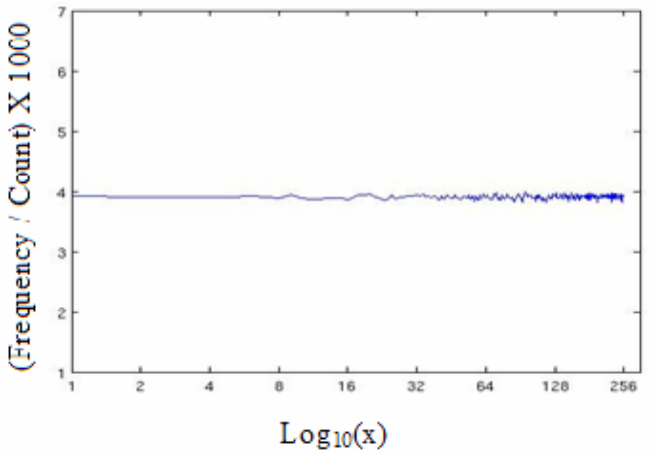

Figure 6: Frequency Distribution of Encrypted Doc Stream.

The frequency distribution of a compressed file appears to be fairly uniform but on looking at it more closely a striking characteristic can be noticed. The figure shows that the frequency distribution at values of powers of '2' is much low. The drop in the frequency becomes quite apparent at higher powers of '2' and as the file size increases. The compressed stream even though has low redundancy and therefore high entropy, it still has some low entropy content which could be due to the implementation of the compression algorithm. According to [9] a perfect compression is quite difficult to achieve and one can expect some redundancy present in a compressed data stream. This non random behavior can be used in differentiating high entropy content from encrypted content.

5.4.2 Phase Space Analysis

This test [15] as described in the previous chapter deals with looking at a data stream in a 'n' dimensional space. This technique [15] of plotting the data in a phase space shows

www.manaraa.com

dependencies between successive elements in the data stream. This test was used to plot the data stream in a 3 dimensional space using the technique known as delayed coordinates.

The equations (3) (4) & (5) given in the previous chapter were used to plot the content 'C' present in the data stream. Three types of data streams of varying entropy such as low entropy plain text streams, high entropy compressed streams and encrypted streams were plotted in order to get an idea of the dependency that was present in the three streams. The Figures 7, 8, 9 below, shows a 3 dimensional plot of a plain, compressed and an encrypted stream respectively. As can be seen by looking at the three figures, the phase space plot of the plain text stream is highly structured. In other words there are areas in the plot where the data content is denser than the rest of the areas. Such distribution in the plot shows that dependencies exist between successive elements in the data stream. Figures 8 and 9 on the other hand show a uniform distribution of the content in the data stream. This tells us that the data stream has quite few dependencies and therefore has high entropy. The results show that both encrypted and compressed streams have much less redundancy and therefore uniform distribution. Phase space analysis therefore did not prove to be a good differentiating test for encrypted and other high entropy streams; it gave us a good idea about the dependencies present in different types of data streams.

www.manaraa.com

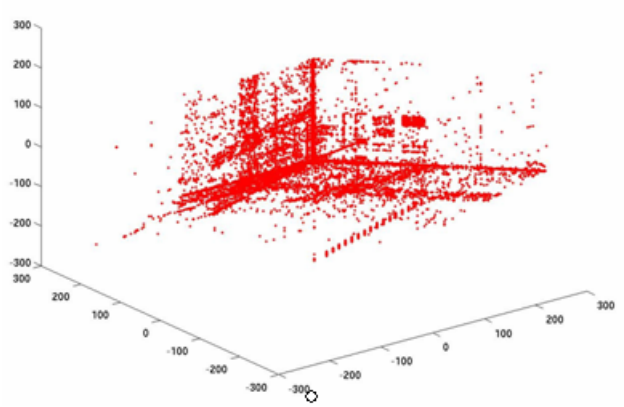

Figure 7: PSA plot of a Doc File

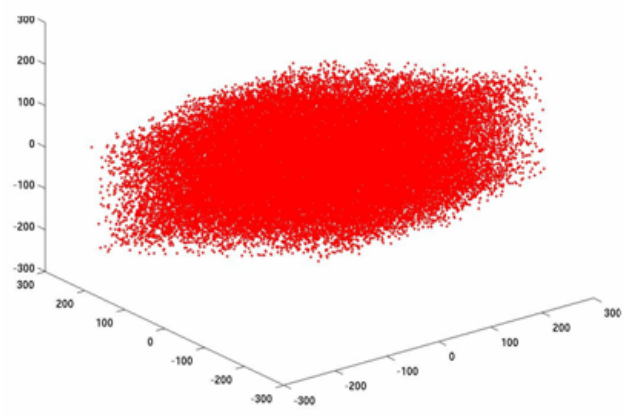

Figure 8: PSA plot of a Compressed Doc File

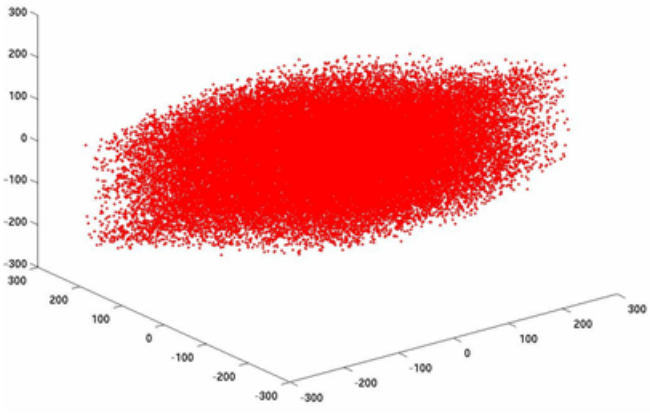

Figure 9: PSA plot of an Encrypted Doc File

www.manaraa.com

5.4.3 KS Test

The Kolmogorov-Smirnov test as described in the previous chapter is a test which can be used as an alternative to the chi-square test. This test was used to check whether a given data stream follows the hypothesized distribution which is highly random as in an encrypted stream. The test was carried out on the three types of data streams ie. low entropy, high entropy compressed and encrypted. The implementation considers 8 bins having 32 values each which means the range from 0-1 is divided in to 8 intervals. The theoretical cumulative distribution is calculated as (1/256)*32*N_{th} interval where N ranges from one to eight. This is because the expected distribution is the distribution of the encrypted data stream which is considered to be highly random and thus each of the symbols has an equal probability of occurrence.

The three tables below show the results of applying the KS test on the three data streams. Table (a) gives the observations of the low entropy data. There is a large deviation between the empirical cumulative distribution S(x) and the theoretical cumulative distribution G(x). This shows that for a low entropy stream this test does well in differentiating from the hypothesized distribution. On the other hand table (b) shows a compressed stream close to the hypothesized distribution and thus this test fails in differentiating high entropy streams from encrypted streams. The below tables give values for 3 file streams having different entropy. The entire file stream was considered as a one continuous data stream and the KS test was applied on it. The packets flowing across a network have a maximum size of 1500 bytes, which is much smaller in size than the file streams considered above.

www.manaraa.com

S(x)	G(x)
0.153	0.125
0.262	0.250
0.363	0.375
0.495	0.500
0.590	0.625
0.702	0.750
0.810	0.875
1.000	1.000

Table (a)

S(x)	G(x)
0.124	0.125
0.248	0.250
0.375	0.375
0.500	0.500
0.627	0.625
0.752	0.750
0.878	0.875
1.000	1.000

Table (b)

S(x)	G(x)
0.125	0.125
0.250	0.250
0.375	0.375
0.500	0.500
0.625	0.625
0.750	0.750
0.875	0.875
1.000	1.000

Table (c)

To simulate the real scenario approximately 60 files from each of the file streams considered above were taken as blocks of 1024 bytes and were then subjected to the KS test. As described in the previous chapter, difference $C=\max_{i} | G(x_i) - S(x_i)|$ was calculated for each of the block in each file stream.

The value 'C' computed is the maximum absolute difference between the cumulative observed distribution and the cumulative expected distribution when considering the hypothesized distribution. This value 'C' can then be compared to a critical value in the Kolmogorov-Smirnov distribution table for a sample size equal to 256. The critical value as given by the table can be calculated as 1.22/SQRT(N) at .10 confidence level where N is the sample size which is equal to 256 in this case. If the computed value 'C' is less than the critical value obtained from the table, the null hypothesis is rejected. In this case the critical value at .10 confidence level came out to be '0.0762'.

The average of the values of 'C' was calculated for all the blocks across all the 60 files considered in each stream and the results are summarized in the table (d) below:

Streams	Low Entropy	High Entropy	Encrypted
μ(C)	0.2026	0.0305	0.0226

Table (d)

www.manaraa.com

The mean of value ‘C’ for low entropy streams is more than the critical value calculated from the KS tables and therefore the null hypothesis can be rejected. The value ‘C’ for high entropy streams is less than the critical value and thus the hypothesis that the distribution is truly random cannot be rejected for high entropy streams. This test therefore also cannot be used to differentiate between encrypted and other high entropy streams even though it does well in differentiating low entropy streams from encrypted steams.

5.4.4 Auto Correlation Test

This test is used to check for randomness in a given stream of data by calculating dependencies/correlations for the data values at various time lags. The auto correlation [16] can be calculated by the equations (6)-(8) given in the previous chapter. For a highly random data stream the auto correlation of the data at any time lag is expected to be quite close to 0. This means that the data stream has quite few or no dependencies present, in other words if any part of the data stream is known it is almost impossible to determine the rest of the stream just by looking at the previous values. Auto correlation values for a data stream having much low randomness would be appreciably higher than 0.

The auto correlation test was implemented to differentiate between encrypted and unencrypted streams by checking for randomness in the data. The null hypothesis was that ‘an encrypted ASCII stream is highly random and the probability of occurrence of each data value is approximately equal to 1/256’. The equations (6)-(8) were used to calculate the auto correlation values for low, high and encrypted data streams. The time lags used to test the hypothesis are shown in the table (e) below. The time lags considered are in powers of 2, this is because the frequency distribution of compressed streams at powers of 2 was observed to

www.manaraa.com

be quite low and we hoped that by checking for correlations in the data streams at these lags could show a characteristic pattern which may help in differentiating high entropy and encrypted streams.

The observations in the table (e) below are the average auto-correlation values across 60 file streams from each of the three entropy categories, each broken in to 32Kbyte data blocks. The observations show that the auto-correlation values for encrypted streams are quite close to 0, thus satisfying the randomness criteria. The low entropy streams on the other hand have a large deviation in their auto-correlation values from 0, thus showing their non-random nature. The high entropy streams also satisfy the randomness criteria to an extent as the deviation from 0 is not large enough to reject the hypothesis. This test therefore also fails in differentiating between high entropy and encrypted streams.

Average Auto-correlation values across 60 streams

Time Lags	Low Entropy	High Entropy	Encrypted
1	0.3250	0.01200	0.00459
2	0.3117	0.01268	0.00462
4	0.3505	0.01034	0.00450
8	0.3263	0.00848	0.00467
16	0.2957	0.00720	0.00460
32	0.2744	0.00645	0.004627
64	0.2536	0.00589	0.00458
128	0.2397	0.00540	0.00464

Table (e)

www.manaraa.com

5.5 Initial Observations and Conclusions

The tests described above do well in differentiating between low entropy streams and encrypted streams. As can be seen from the observations above, both the 'KS' test and the 'Auto-correlation' test offers solutions to clearly differentiate between encrypted and low entropy streams, but fail to differentiate encrypted streams from high entropy streams. The high entropy streams considered are compressed streams having much low redundancy and high entropy. The frequency analysis of compressed streams shows a distinctive behavior at powers of 2. This may be because of the implementation of the compression algorithms. None of the tests described above were able to make use of the characteristic nature of high entropy compressed data streams. The next section describes tests which when implemented made differentiating high entropy streams from encrypted streams easier and gave low false negative and positive rates. The section also describes statistical tests to quickly differentiate low entropy streams from encrypted streams for an online implementation and with quite less overhead.

5.6 The Final Tests

Differentiating between encrypted and unencrypted data was done by a set of statistical tests which tested the data corpus for high entropy and randomness. Whenever the tested data showed signs of low randomness, or dependency on the previous data stream, the hypothesis that '*the test data is encrypted*' was rejected. The test data consisted of files of various common formats like DOC, JPEG, TXT, and PDF. A description of the test corpus has been given the previous section. The tests were applied to the test corpus and depending on the false positive and negative rates, were either tweaked and improved upon or provided

www.manaraa.com

some information previously not known, to be used for later tests. An effort was made to implement the tests so that they could work online with real time network traffic and still achieving acceptable false positive and negative rates.

5.6.1 The Filtering Process

The first step in detecting encrypted streams is to determine whether or not the data stream is a low entropy plain text. To determine this, the 'Information Entropy' of the data was calculated. The Information entropy of the outgoing traffic stream is calculated using the Shannon's entropy equation (2) as given below:

$$H(x) = \sum_{i=0}^{n} p(x_i) \log_2 (1/\, p(x_i)) \qquad (14)$$

$$= -\sum_{i=0}^{n} p(x_i) \log_2\, p(x_i) \qquad (15)$$

P(x) is the probability of occurrence of the i^{th} character in the data stream.

The frequency of occurrence of each ASCII character in the stream is calculated, and this value is divided by the total length of the stream to calculate the probability of each of the 256 possible ASCII characters present in the stream. The logarithm can be calculated in real time quickly using a lookup table having all the possible log values. An entropy value closer to 8bits/character indicates a high entropy stream or a highly random stream of data which is a characteristic of encrypted data whereas values appreciably less than 8bits/character indicates low entropy content.

Another simple method to differentiate high entropy streams from low entropy streams is the calculation of arithmetic mean (μ). The mean is calculated by simply adding

www.manaraa.com

the byte values of each character in the stream as it is encountered and then dividing the sum by total number of characters encountered in the stream. A uniformly random data stream would have a mean value of “127.5” which means that there’s a uniform distribution of characters in the stream and thus every character has an equal probability of occurrence. A low entropy data stream would have a mean value which would show a high deviation from the expected mean value of “127.5”.

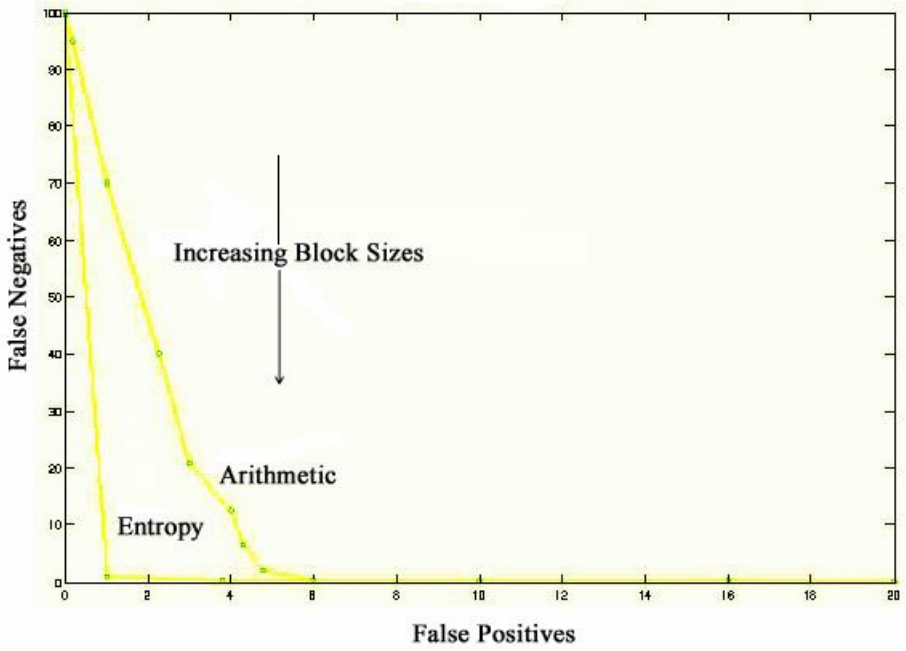

Figure 10: ROC plot of Arithmetic Test versus Entropy Test

Shown above in Figure 10 is a ROC curve, which plots False Positives against False Negatives for the two methods, described above to differentiate low entropy streams from high entropy streams. We consider encrypted and unencrypted files in various formats such as Docs, Jpegs, Pdf, and Txt. The points on the plot indicate different block sizes ranging from 256bytes to 8192bytes. A total of 100 encrypted and unencrypted files of different sizes were used to generate the plot above.

www.manaraa.com

A deviation of 0.5 or less from the expected mean of 127.5 was considered as acceptable for an encrypted data stream. If $\mu<127$ or $\mu>128$ for an encrypted stream, it was considered to be a false negative and if $127 < \mu < 128$ for an unencrypted stream it's a false positive. For the entropy test, the threshold was kept at 7.95, therefore any encrypted data stream with entropy value less than 7.95 was considered a false negative and a false positive was when a compressed stream had entropy more than 7.95. The ROC curve indicates that there's a high false negative rate for both arithmetic and entropy tests at smaller block sizes ranging from 256 bytes to 8192 bytes. For a 4KB block the False Negative rate decreases drastically to .98% for entropy test. On the other hand the false positive rate increases as the block sizes increase for both the methods but the increase is more at higher block sizes for the entropy test as compared to the arithmetic test. The best results with fewest false positives and negatives occur at block sizes 4KB and 5KB for the entropy test.

5.6.2 Index of Coincidence

If the data stream is unable to be categorized as a low entropy stream, there's a high possibility that the stream may be compressed or is a highly random file stream such as JPEG, PDF which have a compressed format. We now require methods to differentiate such data streams from encrypted streams.

The next step in determining the presence of encrypted streams is calculating the Index of Coincidence (IOC) [19] of the data stream which exhibits a high entropy behavior in the above step. The Index of Coincidence is a statistical measure of redundancy in a stream and even unencrypted high entropy streams such as compressed and other compressed file formats show a distinctive behavior. The IOC value [19] of a uniform distribution is very

www.manaraa.com

close to 1, which means that the data stream is independent and has very little or no redundancy present.

The IOC values were computed for high entropy files (encrypted and compressed) for a block size of 1024bytes. The IOC values were calculated using the IOC equation (16) below.

$$\mathrm{IC} = \frac{\sum_{i=1}^{c} n_i (n_i - 1)}{N(N-1)/C} \qquad (16)$$

N: The length of the text,

n_i: The frequency of the character i in the text,

C: The number of letters in the alphabet.

The values were plotted for an encrypted document (DOC) file having approximately 1800 blocks. The figure (11) below shows an IOC plot of an encrypted and a compressed file, the Y axis represents the IOC values. The plot shows a randomly distributed stream as the IOC values are much close to 1. The plot in figure (11) also shows the same file compressed and the IOC values show a large deviation from 1, thus showing that the compressed stream is not independent and has some redundancy present. The peaks in the compressed plot in figure (11) indicate areas of low entropy and high redundancy. The peaks occur frequently throughout the stream which shows that the compressed stream has certain dependencies which occur after some amount of data has been processed; we suspect this is due to the nature of the compression algorithm.

www.manaraa.com

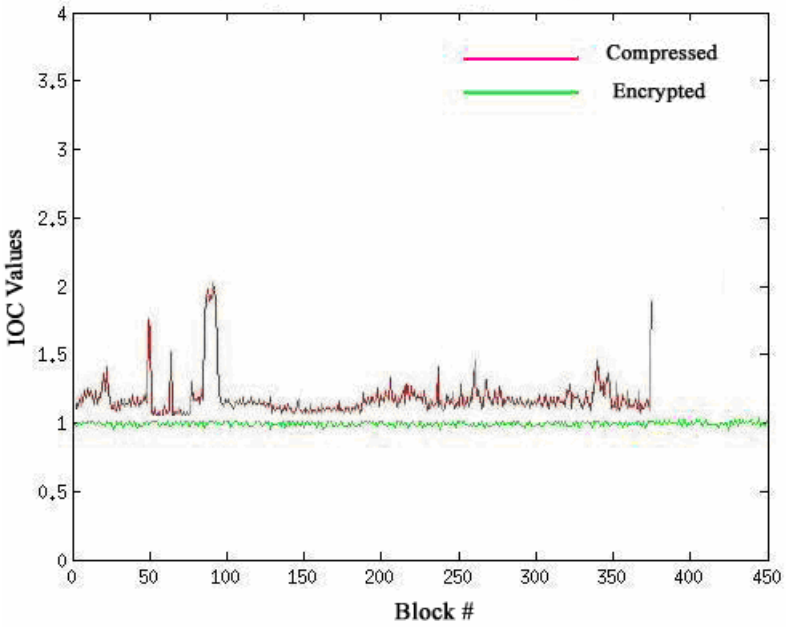

Figure 11: IOC plots of Compressed and Encrypted Streams

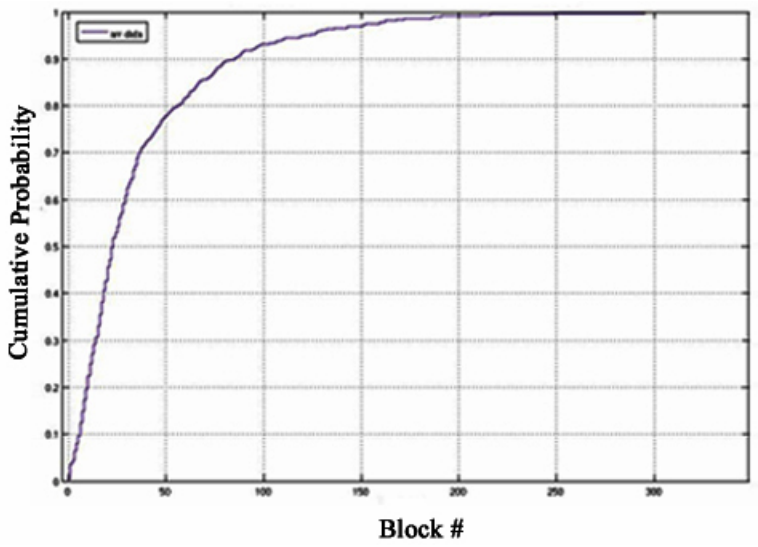

Figure 12: Cumulative Probability Plot of Distance between IOC peaks

The distance between the peaks thus provides an estimation of the minimum number of data blocks required to differentiate between high entropy encrypted and compressed streams.

www.manaraa.com

The figure (12) above shows a cumulative probability distribution of the distance between the IOC peaks across all the compressed files. The threshold for a peak was calculated by the below equation.

$$T=\mu+2\sigma \qquad (17)$$

Here μ is the mean IOC value of the high entropy data stream which appears quite random and σ is the standard deviation of the IOC values from the mean. The threshold "T" was calculated to be approximately 1.062. The peaks are determined by finding the IOC values which are greater than the threshold "T" and have a higher value than all the other surrounding peaks within a window of size 'S'. The figure (12) above shows that around 90% of the peaks occur within a distance of 75-80 blocks of data or less. In other words for a 90% confidence level to detect compressed stream one needs to collect around 75KB of the stream.

5.6.3 Differentiating high entropy content

After calculating the IOC values and determining the approximate block distance between the peaks from the IOC plots, we estimate the number of data blocks needed to differentiate between high entropy content and encrypted stream with a high confidence level.

The final step in differentiating encrypted streams from compressed streams is to implement a method which would look at the deviation of the high entropy compressed data stream from the expected encrypted data stream. We implement the Incremental Chi-Square test to differentiate the high entropy stream from encrypted stream, wherein we buffer 70-80

www.manaraa.com

KB of data to carry out the Chi-Square test. Chi-Square test is one of the most commonly used tests to check for randomness in a data stream. The test calculates the degree to which the two given samples of data differ from each other therefore it's used to test whether a set of data follows a particular distribution.

The Chi-Square test also tells the degree of confidence in rejecting or accepting a hypothesis. The null hypothesis that we make is that '*the data stream is encrypted and thus highly random and has a uniform distribution of characters*'.

We expect the probability of occurrence of each ASCII character in an encrypted data stream to be 1/256 as each character is equally probable to occur. The equation used to calculate the Chi-Square is as follows:

$$X^2 = \sum_{i=1}^{k} (O_i - E_i)^2 / E_i \qquad (18)$$

k: The no. of bins

O_i : The observed frequency for the bin i

E_i: The expected frequency for the bin i.

To classify the data stream 32 bins were considered i.e. 31 degrees of freedom, with each bin having the frequency counts of 8 characters. The expected frequency E_i is equal to 1/256 * 8 * (The total length of the block). This is the frequency one would expect if the data stream was encrypted as the probability of occurrence of each character is equal. The frequency of each character in a block is determined and the above Chi-Square equation is used to calculate the chi-square value of the entire block. This value is stored and the next block of the data stream is appended to the current block and the chi-square value is

www.manaraa.com

calculated again as above. This procedure continues until sufficient no. of blocks are appended which is determined by the distance between the IOC peaks in the previous step.

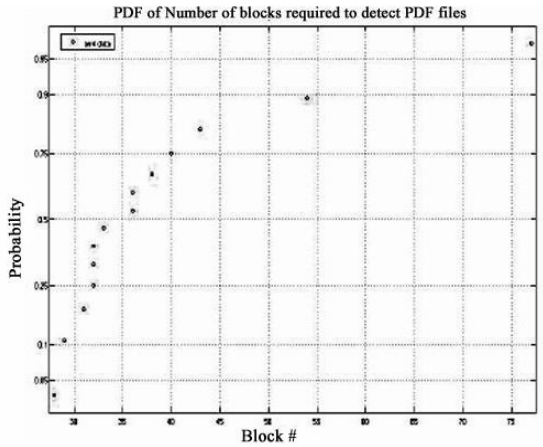

Figure 13

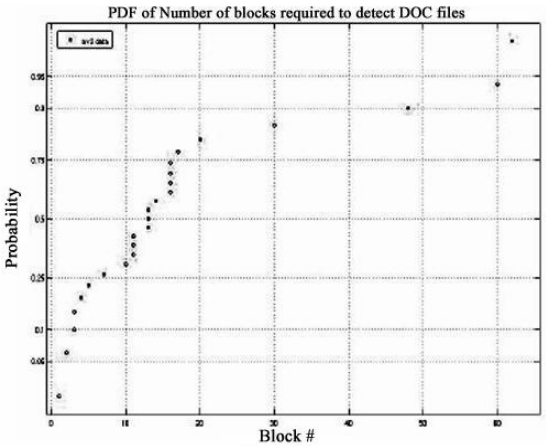

Figure 14

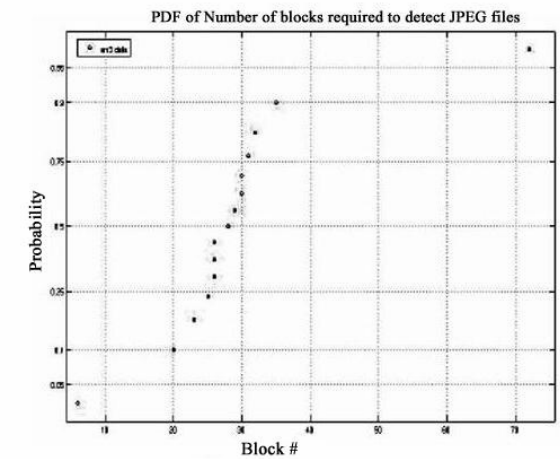

Figure 15

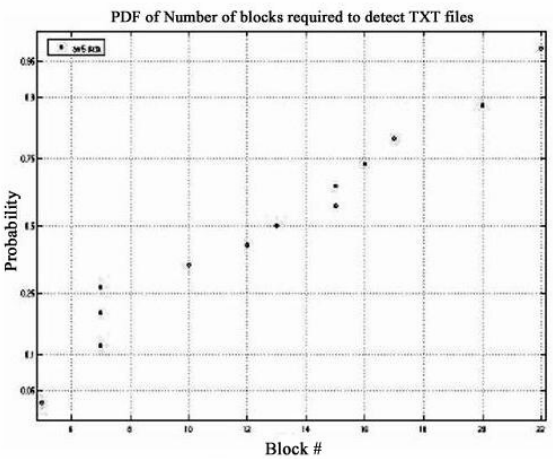

Figure 16

The chi-square values obtained for the data blocks are then compared with the value in the chi-square probability table for a confidence level. The confidence level we choose is '0.90' which means that the probability that the observed chi-square value is higher than the tabulated value is 10% or less. Thus if more than 1 Chi-Square value observed in 10 values is

www.manaraa.com

more than the value in the chi-square distribution table, there is a high probability that the data stream is not encrypted and the stream is considered to be compressed.

The average no. of blocks of data required to differentiate high entropy compressed stream from encrypted streams for different file formats is shown in the above Figures 13 - 16. The no. of blocks in the above figures is calculated by taking the mean of the distances between the IOC peaks. The mean distance gives an idea about the no. of blocks that would be required on an average to detect the different file streams.

The above Figure 13 shows that around 30 blocks or 30KB of data is required to make a distinction between encrypted files and compressed PDF files. The confidence level at 30KB is 90%, which means that 1 out of 10 files could be classified incorrectly as encrypted and therefore blocked from leaving the company's gateway. We don't think this to be too intrusive as we consider a strong security policy which disallows any confidential data to leave a company unauthorized, and therefore consider the false positives acceptable. Similarly the Figures 14 - 16 above show that it takes around 30, 35 and 20 blocks to detect DOC, JPEG and TXT respectively 90% of the time. This is because the PDF and JPEG formats are highly compressed and thus the data stream has high entropy. The more entropy, the more the data is required to differentiate it from the encrypted stream. The above plots show that as the number of blocks increases, it gets easier to differentiate between compressed and encrypted streams and at relatively higher block sizes it’s possible to achieve a confidence level close to 100%. This is because if the dataset is large, there is more probability of dependencies within the data and a possibility of higher redundancy as some of the data may not be compressed as efficiently as it would when the dataset is small. The data

www.manaraa.com

stream therefore shows characteristics rather similar to a low entropy stream, and these patterns become more apparent when more data blocks are considered.

5.7 Final Observations

For low entropy streams in our corpus 98.7% of the streams will be filtered out in the first stage itself. Only 1.3% low entropy streams go through the next step. The Figures 13 – 16 indicate that 90% of the compressed blocks of any file format can be differentiated within 20-30 blocks. This is a worst case scenario when we consider high entropy compressed streams but for low entropy streams only a few blocks would be required to reject the hypothesis. So for low entropy streams the false positive rate is negligible and approximately equal to zero.

High entropy and encrypted streams satisfy both the arithmetic and entropy tests all the times. Encrypted streams do not show any distinctive behavior for the IOC test and shows much less deviation from the expected, the false negative rate is negligible for these streams. High entropy compressed streams on the other hand show a distinctive profile which can be used to distinguish them. The IOC test gives a maximum number of blocks required to detect high entropy data streams. The more the data blocks considered, higher is the probability of differentiating high entropy streams from encrypted streams. We consider 90% confidence level as an acceptable level for differentiation since this does not lead to a high false positive and at the same time keeps the number of data streams low. For any practical online implementation of these methods for differentiating high entropy streams a lower than 5% false positive rate is quite difficult to achieve but a 10% rate is optimum for a fast implementation.

www.manaraa.com

CHAPTER 6. CONCLUSION

Egress filtering has become an essential component of any organization's information security. Most of the egress filtering tools only look at the contents of the data packets leaving a network but fail to perform if the data is encrypted. We have presented statistical techniques to differentiate encrypted data from all the other data streams flowing across a gateway. We show that encrypted data satisfies all of the statistical tests quickly whereas other data formats except compressed show a large deviation from the expected random distribution and hence are rejected at varying data lengths. Compressed streams mostly satisfy randomness tests but have a distinctive IOC profile which is used to determine the amount of data stream required to reject the null hypothesis that the stream is encrypted. We use Shannon's information entropy equation to differentiate between the encrypted and other low entropy streams. To differentiate between high entropy streams such as encrypted and compressed we employ an incremental chi-square approach where the no. of blocks are determined by the average distance between the peaks in the IOC profile.

www.manaraa.com

CHAPTER 7. FUTURE WORK

This research is an attempt to propose methods and techniques to detect encrypted documents leaving an organization's networks. We believe that differentiating encrypted traffic flows from compressed can be done by considering lesser number of data blocks and still achieving the same or lower false positive and negative rates. We think that it's also possible to determine the compressed and encrypted file formats by measuring the entropy of the data stream. Similarly the encryption and compression algorithms may be determined as well.

www.manaraa.com

BIBLIOGRAPHY

[1] L.A.Gordon, M.P.Loeb, W.Lucyshyn and R.Richardson. *CSI/FBI Computer Crime and Security Survey*. Computer Security Institute, *2006.*

[2] J. Pierce. *Egress Filtering For a Better Internet.* GSEC Practical Version 1.4c, SANS Institute, September 7, 2004.

[3] B.E. Burke. *Information Protection and Control: Websense Targets the Insider Threat.* A white paper sponsored by Websense, March 2007.

[4] M. Liberatore and B. N. Levine. *Inferring the Source of Encrypted HTTP Connections*. Department of Computer Science, University of Massachusetts, Amherst, MA 01003-9264. 2006.

[5] Q. Sun, D. R. Simon, Y. Wang, W. Russell, V. N. Padmanabhan, L. Qiu. *Statistical Identification of Encrypted Web Browsing Traffic*, Technical Report MSR-TR-2002-23, Microsoft Research, Microsoft Corporation, March 8, 2002.

[6] Q. Zhang, D. S. Reeves, P. Ning, S. P. Iyer. *Analyzing Network Traffic To Self-Decrypting Exploit Code*. Cyber Defense Laboratory, Computer Science Department, North Carolina State University, Raleigh. 2007.

[7] McAfee Network Protection Solutions, White Paper. *Encrypted Threat Protection Network IPS for SSL Encrypted Traffic*. October 27, 2007. < www.mcafee.com/us/local_content/white_papers/wp_encr_th_prot.pdf>

[8] A. Shamir and N.V. Someren. *Playing hide and seek with stored keys*. Proceedings of the third international conference on financial cryptography, Pages: 118 – 124. September 22, 1998.

[9] C.E. Shannon, *Communication Theory of Secrecy Systems*. The Bell system technical journal, Vol 27, pp. 379-423, 623-656, July, October, 1948.

[10] A.N. Kolmogorov. *Three approaches to the quantitative definition of information.* Problems of Information, Transmission, vol.1, pp. 1–7, 1965.

[11] G.J. Chaitin. On the length of programs for computing finite binary sequences. *J. ACM*, vol.13, no.4, pp. 547–569, 1965.

[12] J.L Gailly, M Adler. *Inflate & Deflate algorithms.* October 12, 2007. <*http://www.gzip.org/algorithm.txt*>.

www.manaraa.com

[13] J. Seward**.** bzip2 and libbzip2, version 1.0.3, A program and library for data compression. October 20, 2007 <http://www.bzip.org/1.0.3/bzip2-manual-1.0.3.html>.

[14] A. Shapira. *The Discrete Runs Test and the Discrete Maximum of t Test.* Technical Report CS 96-15, ECSE Department Rensselaer Polytechnic Institute Troy, NY 12180. June 10, 1996.

[15] Symantec, Support and services, White Paper. *Strange attractors and Tcp/Ip sequence number analysis*. October 12, 2007. *<http://www.bindview.com/Services/Razor/Papers/2001/tcpseq.cfm>*.

[16] Engineering Statistics Handbook. Autocorrelation Plot. October 27, 2007 *<http://www.itl.nist.gov/div898/handbook/eda/section3/autocopl.htm>*.

[17] U.W Pooch, J.A. Wall. *Discrete Event Simulation: A Practical Approach.*

[18] Engineering Statistics Handbook. Chi-Square Goodness-of-Fit Test. October 27, 2007. *<http://www.itl.nist.gov/div898/handbook/eda/section3/eda35f.htm>*.

[19] D.E Knuth. Art of Computer Programming, Volume 2: Seminumerical Algorithms

[20] W.F Friedman. *The Index of Coincidence and Its Applications in Cryptography*. Laguna Hills, CA: Aegean Park Press, 1986.

www.manaraa.com