Statistical Profiling and Visualization for Detection of Malicious Insider Attacks on Computer Networks

Jeffrey B. Colombe
The MITRE Corporation
Emerging Technologies Office
McLean, VA
(703) 883-5307
jcolombe@mitre.org

Gregory Stephens
The MITRE Corporation
Dept. of Secure Distributed Computing
McLean, VA
(703) 883-3242
gstephens@mitre.org

ABSTRACT
The massive volume of intrusion detection system (IDS) alarms generated on large networks, and the resulting need for labor-intensive security analysis of the text-based IDS alarm logs, has recently brought into question the cost-effectiveness of IDSs. In particular, when host-based IDSs are used to monitor an organization's internal networks, the majority of the resulting alarms represent legitimate, automated system administration. Because of the absence of ground truth about known attacks, we propose an unsupervised, anomaly-based method for automatically distinguishing alarms that are potentially generated by malicious insider attacks, from the repetitive and temporally structured legitimate system-administration alarms. The majority of previous work in this area has used heuristic and statistical filtering techniques to discard a relatively large proportion of alarms in the final presentation to the security analyst, which is a potentially dangerous practice. Instead, we demonstrate the use of a typicality measure to visualize the apparent risk associated with alarms, while retaining information about the temporal context of the entire alarm stream for the analyst to view. The relevance of the statistical method is examined by comparing the results to a set of analyst-curated alarms from an operational environment.

Categories and Subject Descriptors
CR Categories: C.2.0 [**Computer-Communication Networks**] General--Security and Protection; G.3.2 [**Probability and Statistics**] Statistical Computing; H.1.2 [**Models and Principles**] User/Machine--Human Factors; H.2.8 [**Database Management**] Database Applications--Data Mining; H.4.2 [**Information Systems Applications**] Types of Systems--Decision Support; I.3.3 [**Computer Graphics**] Picture/Image Generation--Display Algorithms; I.3.6 [**Computer Graphics**] Methodology and Techniques--Interactive Techniques; K.6.5 [**Management of Computing and Information Systems**] Security and Protection.

VizSEC/DMSEC'04, October 29, 2004, Washington, DC, USA.

General Terms
Algorithms, Security, Human Factors.

Keywords
information visualization, cognitive load, human-computer interaction, anomaly detection

1. INTRODUCTION
Intrusion detection systems (IDSs) have become an important part of the organizational infrastructure used to monitor computer and network security. IDSs can be network-based or host-based and can be architected to detect external attackers or malicious activity within an organization's networks. In most IDS implementations, a knowledge engineering approach is used to codify intrusion or misuse signatures based on the experience of network security analysts. Network or host activity is compared to intrusion or misuse signatures, and alarms are generated if the activity matches any of the signatures. This type of IDS is called *signature*-based (as opposed to *anomaly*-based). Individual IDS sensors generate alarms, each of which has parameters identifying the specific alarm and detailing the conditions under which it occurred. The alarms from all IDS sensors are typically aggregated into a single alarm stream which is then centrally stored within a database where it is analyzed by security analysts.

One of the greatest obstacles limiting the effectiveness of today's IDSs is the enormous volume of alarms that they generate. Large-scale IDS implementations can generate millions of alarms per day, far beyond the ability of a security analyst to analyze and interpret. An overwhelmingly large number of these alarms are false positives, requiring the security analyst to hunt for the relatively infrequent true positives in a mountain of false alarms. The high false alarm rates of IDS alarms has been identified as a major drain on human labor resources that brings the cost-effectiveness of IDS software under question [1, 8]. We have observed that the alarm overload problem becomes particularly severe when IDSs are used to monitor an organization's internal networks for insider misuse. In the case of host-based IDSs on internal networks, we have found that system administrators performing authorized, automated maintenance across large numbers of internal hosts cause the large majority of alarms, which are false. As a result, the security analyst must either manually investigate all of the alarms or filter out all alarms associated with system administrators. Both of these options

carry enormous risks. The sheer volume of host-based alarms makes manual investigation impractical and risks missing a true positive alarm. On the other hand, filtering out alarms associated with system administrators essentially blinds the analyst to system administrator misuse.

This paper explores the use of inference and visualization techniques to effectively filter out the host-based false positives caused by authorized, automated system administrator activity, which may include, for example, the installation of new software applications or the changing of configuration settings across an organization's computer base. Because of the rarity of known insider attacks to serve as training data for supervised learning [2], we employ an unsupervised learning technique to measure how statistically anomalous each alarm appears to be given the alarms that surround it in the alarm stream.

Prior work directed at reducing cognitive load for the security analyst has used a variety of approaches for conducting unsupervised anomaly detection. Some of these approaches seek to group or cluster alarms based on the similarity of their descriptive content or their temporal ordering, resulting in a *symbol dictionary* representation of alarm classes that simplifies the information presented to the security analyst [e.g., 4,15,16,18]. Several other techniques, discussed below, begin their analysis with a symbol dictionary formed directly from the set of unique alarm descriptions, and use this representation to analyze the statistical behavior of alarm streams. The simplest approach to anomaly detection is to use first-order alarm statistics, or an alarm histogram. Alarms with a higher support, or relative frequency, in the training data can be regarded as less likely to be hostile, given the observation that true attacks are very infrequent [2]. Other methods use the temporal association of alarms to determine normally patterned activity, and to distinguish anomalous activity by contrast. These include association rule mining (ARM) [12], frequent episode detection [3,13], and hidden Markov modeling (HMM) [11]. All of these methods for anomalous alarm detection use a hard threshold between normal and anomalous events to give a security analyst a reduced set of alarms to consider, by selecting a subset for display. In such approaches, all alarms still exist in a repository, but the typical alarms that are discarded from a user display would only be scrutinized directly at a later time, as part of a forensic investigation resulting from suspicious evidence of a true attack. The danger in such an all-or-none approach is that important and sometimes subtle information might be lost if the majority of alarms are excluded from the security analyst's display entirely [9, 10].

The use of visualization techniques has been identified as a possible solution to the inefficiencies of current IDS methods, which rely on an overwhelming amount of text data that consist largely of false alarms, and that are difficult to assess for risk, particularly on the short timescales required for real-time analysis of intrusions [6]. The urge behind using visualization to analyze large and complex datasets makes use of the intuition that a picture may be worth a thousand words, but may be much easier and less time-consuming for human brains to interpret [17]. Most visualizations of computer networks to date have been concerned with traffic flow analysis, bandwidth issues, or the relative efficiency or performance of communication channels. Visualizations for intrusion analysis in computer networks have been developed more recently and represent a relatively small subset of the visualization methods available for analyzing the behavior of networks. The majority of these have used a similar approach to visualizing intrusion alarm data, employing node-link representations with glyphs, color codes, and spatial layouts [5,6]. Histograms of activity patterns, and animations of traffic history, have proven useful for both real-time and forensic traffic analysis [6]. The need to reduce analytic clutter provided by false alarms, and to use some sense of typical network behavior to highlight unusual or suspicious activity, have been identified, and initial approaches have been developed [e.g., 5,19].

2. METHODS

2.1 Data Representation and Statistical Analysis of Alarms

We converted the comma-delimited text descriptors of the RealSecure alarm format:

```
EventID:146321,
EventDate:2001-07-27 03:56:13.000,
Event_Name:Audit_policy_change,
EventPriority:2,
SourceIPAddress:010.010.012.012,
SourcePort:65535,
DestIPAddress:010.010.012.012,
DestPort:65535,
Account Management Failure:-,
Account Management Success:+,
AlertFormatVersion:85,
AttackOrigin:Local,
etc.
```

into a binary representation indicating the presence or absence of each comma-delimited text descriptor, for example:

$$(1, 0, 0, 1, 1, 0, 0, 1, ..., 0)$$

Each element in an alarm vector corresponds to a specific descriptor token. A '1' indicates the presence of a token in an alarm, and each alarm vector is as long as the lexicon of tokens that have been seen so far in the data set. This representation, called a *multivariate Bernoulli event representation*, encodes all of the information in an alarm description, but renders all of the descriptions numeric and generates vectors of equal length from alarm to alarm, which facilitates machine learning approaches and alarm visualization. An alternate representation of the alarm stream was generated in which each unique alarm description was given a symbol, represented as a single binary entry in a symbol dictionary. In some cases, the original attribute descriptions are quantitative and may not be suitable for tokenizing, for example the `EventDate`. If quantities are highly variable, the diversity in the data set will also make the formation of a lexicon and a symbol dictionary prohibitively expensive due to an explosion of unique tokens and unique Bernoulli alarm descriptions. A minority of existing approaches [e.g., 7] preserve quantitative and/or qualitative attribute descriptions of events as real-valued vectors $\Re^d$, and apply methods such as clustering or kernel-based comparisons to discover anomalous alarms, or patterns of alarms.

Two IDS data sets were available for analysis, one a notional data set created under partially controlled laboratory conditions, the other a data set from an operational environment. The notional data set consisted of 221,635 alarms gathered from an installation of the RealSecure host-based IDS over 110 days in MITRE's

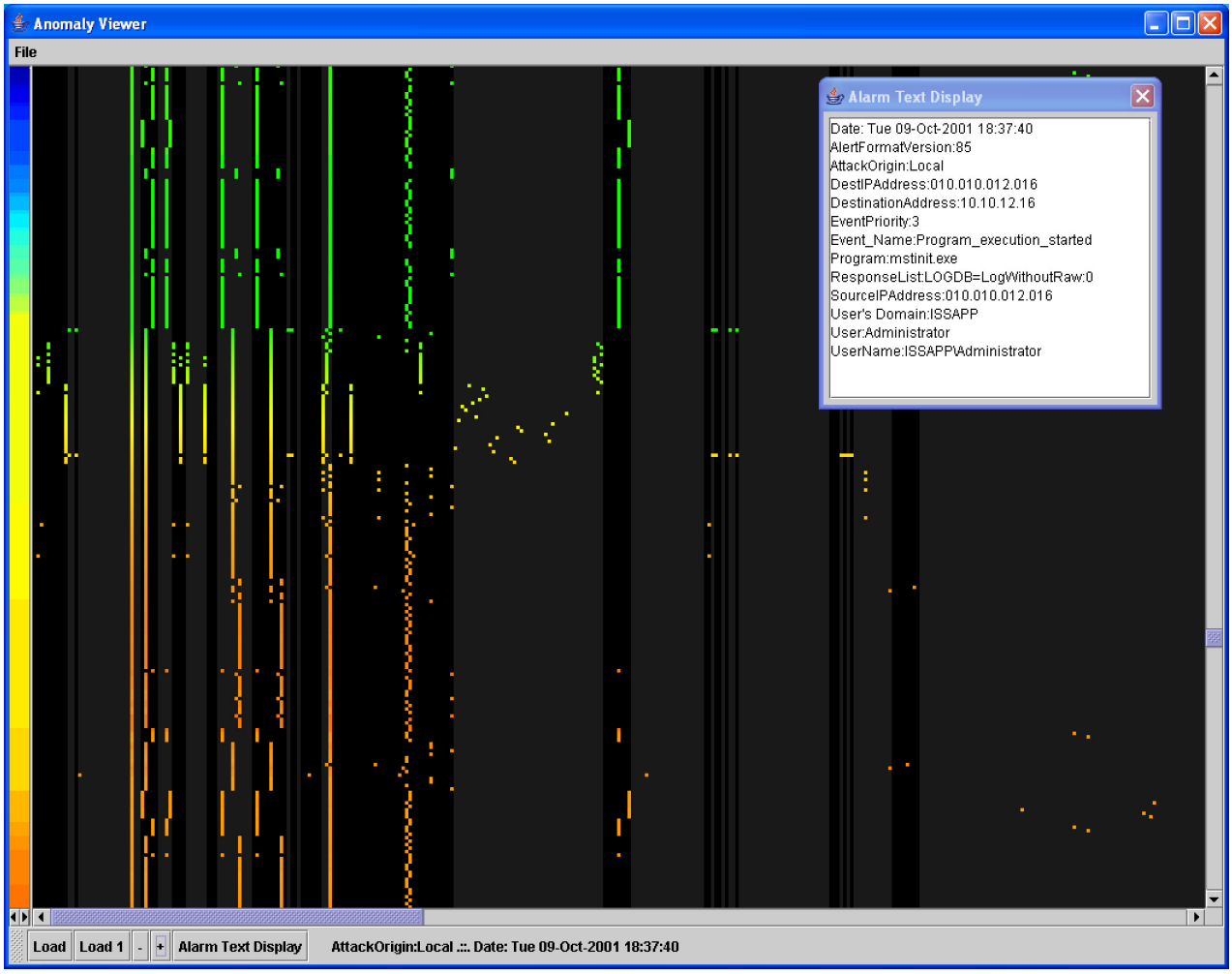

Figure 1. Alarm visualization software showing a tokenized Bernoulli vector representation of notional alarms. This visualization consists of *alarms* (rows) in chronological order, from top to bottom. Each column indicates the presence or absence of a specific descriptive *token* (key:value descriptor). Pixels that are illuminated in a column indicate the presence of that row's token in the alarm, and their color indicates the typicality score of the alarm. Black/grey pixels represent the absence of tokens. The color code on the left represents the time modulo one hour, where deep blue is the top of the hour and deep red is the bottom of the hour. The text window at the upper right shows the text descriptors of a mouse-clicked alarm in the main window.

Information Systems Security (INFOSEC) Laboratory, McLean, VA. The notional alarm data contained a mixture of routine non-automated user activity and more rare automated activity caused by the running of a vulnerability scanner. The operational data set consisted of 502,125 RealSecure host-based IDS alarms collected over 24 hours on a large computer network at a MITRE customer site. This data set contained a large volume of automated system administrator activity mixed in with the non-automated activity of both regular users and system administrators. We examined the first-order statistics (frequencies of descriptors and symbols; results not shown) and second-order statistics (frequencies of pairs of symbols within a time window) of host-based IDS alarm streams generated in both notional and operational security environments.

In order to perform anomaly detection, a statistical *typicality* score was calculated for each alarm. Space does not permit a detailed reporting of the method, however, the authors may be contacted for a longer version of this communication. To measure the typicality of an alarm, a user-defined time window was used to define a set of nearby alarms and their symbol representations. The typicality of alarm i was the sum of the number of times its symbol representation has appeared within a time window alongside the other symbols in the entire available history of activity on the network. If an alarm symbol has appeared often in proximity to the same symbols it appears alongside now, its typicality will be high. If an alarm symbol appears in an unusual temporal context, its typicality will be low and it will be regarded as anomalous. The time window used in the present study was 15 minutes for notional data, and 15 seconds for operational data based on manual inspections of the alarm rate in each domain. A

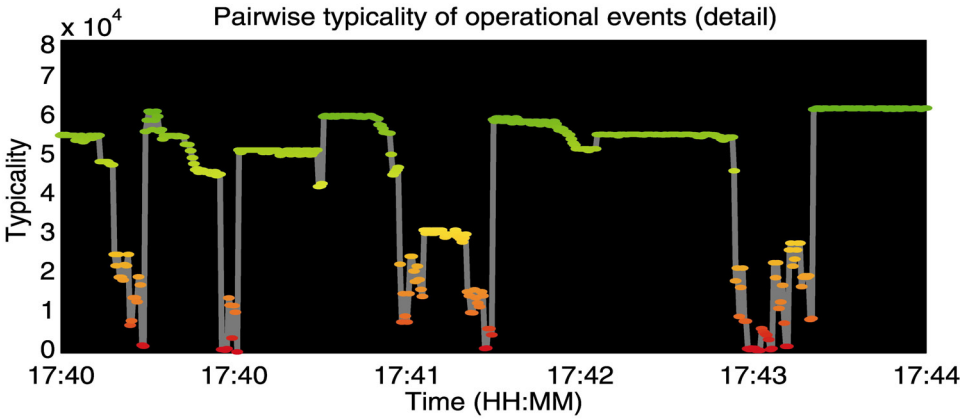

Figure 2. Detailed timeline of the typicality scores of operational alarms. A baseline of high-typicality alarms is punctuated by bursts of less typical alarms. Cool tones (greens) indicate highly typical alarms, warmer tones indicate anomalous alarms.

set of 998 alarms from the operational data set was chosen at random, and checked to ensure a relatively homogeneous coverage of the timeline of the data set so that all epochs of activity were represented (not shown). A security analyst familiar with the operational site curated alarms by comparing their individual content to the context of alarms surrounding them in time, and classifying each alarm as either legitimate automated use of administrative accounts, or events that were judged not to be legitimate automated account use and thus candidates for further analysis as potential malicious insider activity. The correlation coefficient between the typicality score and the frequency of curation as nonautomated status, and receiver operating characteristics (ROC, not shown), were calculated.

2.2 Visualization

Rather than use a threshold for excluding typical alarms, we have explored the use of visualization techniques to simultaneously display a typicality score based on low-order *n*-gram statistics [14], in this case pairwise association frequencies, along with several other important parameters of each alarm. Our goal was to transform what begins as a rapidly accumulating sea of text strings into a patterm of alarm dots plotted in their relative time sequence so that adjacent temporal structure and statistical typicality across a large number of adjacent alarms can be seen at a glance. The idea of the display is to tap into the user's visuospatial pattern recognition skills. Even if a set of alarms have a moderate or high typicality score, but the pattern seems unusual nonetheless, an analyst is able to 'drill down' in the GUI by clicking on individual events to study the text alarms and further assess the threat. For visualization purposes, alarms were displayed so that the absolute sequence of alarms (encoded as horizontal position on the screen), the *in modulo* time of alarms (for example, modulo an hour, encoded as rainbow palette coloration), and absolute time of alarms (in graphic and text form) were all readily available. The typicality score T_r of events was displayed as a color code with a palette that ranged continuously from red (for low typicality alarms) through yellow (for intermediate values of T_r) to green (for highly typical alarms). Color-coded numerical data was normalized so that the extremes of the color palettes fit the extremes of the encoded variables. Zoom functions were implemented to magnify displayed regions of an alarm stream, and the text representation of an alarm was displayed by clicking on the illuminated pixel of the symbol representation of the alarm. Visualization software was developed in MATLAB and in Java to facilitate manual examination of both the Bernoulli representation and symbol representation of alarm streams, displaying the exact ordering of alarms, the exact timing of alarms, and the typicality scores of alarms, as well as the text-based descriptions of alarm properties and times.

3. RESULTS AND DISCUSSION

The notional alarm stream was characterized by a low level of highly regular and repetitive activity, with infrequent bursts of unusual activity, much of which was apparently automated (e.g., vulnerability scans). The background level of automated activity was nonstationary, varying substantially in epochs lasting on the order of weeks (not shown). Visualization of the notional data using the Bernoulli representation (Figure 1) shows activity that is repetitive, with both sequence and precise temporal structure that is ordered, but not strictly periodic. There were 1090 unique event descriptions, resulting in a symbol dictionary with 1090 distinct entries. The distribution of pairwise association frequencies between symbols in inside the user-defined time windows (15 minutes, maximum 500 events) was highly peaked at zero, with a range of mostly very low values and a small number of very high values (maximum 793,031 joint occurrences). The typicality of alarms varied in correspondence with the epochs of varying alarm frequency. The background of repetitive automated activity had locally high typicality scoring that was punctuated by bursts of unusual activity with a relatively low typicality score. In Figure 1, alarms are color-coded by their typicality, where typicality is mapped from its lowest to highest historical value on a smoothly varying red-yellow-green color map. Figure 1 shows a volley of relatively fast, atypical automated activity (yellow to red) interrupting a slower, more typical sequence of automated alarms (in green).

The operational data, gathered from a much larger network than the notional data, showed much higher absolute alarm rates in total (not shown). When converted to the Bernoulli and symbol representations, the vast majority of unique alarms were revealed to be very infrequent, and a small number of unique alarm descriptions dominated most of the alarm stream. A very large number of very stereotyped alarms, with very stereotyped key fields, dominated the alarm stream. 17,693 distinct alarm signatures (symbols) were seen in the operational data. Detailed examination of segments of the alarm stream reveals a somewhat similar pattern of typicality scores as was seen in the notional data, where plateaus of relatively typical activity are punctuated by volleys of substantially less typical alarms (Figure 2). To provide a cursory check on the effectiveness of the typicality measure as a warning of potentially malicious insider activity, a simple classification technique was applied to the analyst-curated alarm data, using a classification threshold that varied as a function of normalized typicality for distinguishing nonautomated from automated activity. Space does not permit graphic presentation of these results; authors may be contacted for a longer version of this communication. However, the good correlation (R = -0.789) between typicality scores and the frequency of nonautomated alarms, and the relatively poor performance of typicality as a classification threshold (area under ROC = 0.605), both emphasize the importance of typicality-based *visualization,* rather than classification and discarding, of alarms.

A potential weakness of this method, as with all anomaly-based approaches, is a degree of vulnerability to *replay attacks*, in which normal system activity is recorded and replayed with minor hostile variations. Normal behavior may also be *modeled*, for example probabilistically, using techniques that will generate typical sequences of alarms to disguise hostile actions. Such statistical modeling would, however, represent a sophisticated, labor-intensive effort requiring a thorough knowledge of the network of interest, and is thus considered a more unlikely threat whose detection is beyond the scope of this effort.

4. ACKNOWLEDGEMENTS

This work was funded through the National Security Agency's R23 group. The operational data set was provided by an anonymous MITRE sponsor. Helpful comments were provided by Dave DeBarr, Dale Johnson, Amy Herzog, Eric Bloedorn, Kate Arndt, and Tom Goldring. Demonstration software was developed by Chris Johnson and Diane Christoforo.

5. REFERENCES

[1] Allan A (2003) Intrusion detection systems: Perspective. *Gartner Research Report*. April 24, 2003.

[2] Axelsson S (2000) The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security* 3:186-205.

[3] Clifton C and Gengo G (2000) Developing custom intrusion detection filters using data mining. In: *Military Communications International Symposium (MILCOM2000)*.

[4] Dain O and Cunningham RK (2002) Fusing heterogeneous alert streams into scenarios. In: *Applications of Data Mining and Computer Security*. Kluwer Academic Publishers, Boston.

[5] Erbacher RF and Frincke D (2001) Visual behavior characterization for intrusion and misuse detection. *Proceedings of the SPIE 2001 Conference on Visual Data Exploration and Analysis VIII*, San Jose, CA, January 2001, pp 210-218.

[6] Erbacher RF and Sobylak K (2002) Improving intrusion analysis effectiveness. *2002 Workshop on Computer Forensics,* Moscow, ID, September, 2002.

[7] Eskin E, Arnold A, Prerau M, Portnoy L and Stolfo L (2002) A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In: *Applications of Data Mining in Computer Security (Advances in Information Security, 6)*, Jajodia S and Barbara D, Eds. Kluwer Academic Publishers.

[8] Hulme GV (2003) Gartner: Intrusion detection on the way out. *Information Week*, June 13, 2003.

[9] Julisch K (2003) Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security* 6.

[10] Julisch K and Dacier M (2002) Mining intrusion detection alarms for actionable knowledge. *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining*, Edmonton.

[11] Lane T (1999) Hidden Markov models for human/computer interface modeling. In: *Proceedings of the IJCAI-99 Workshop on Learning about Users*, pp 35-44.

[12] Manganaris S, Christensen M, Zerkle D and Hermiz K (2000) A data mining analysis of RTID alarms. *Computer Networks* 34:571-577.

[13] Mannila H, Toivonen H and Verkamo AI (1997) Discovery of frequent episodes in event sequences. *University of Helsinki Department of Computer Science, Series of Publications C, Report C-1997-15*.

[14] Mel BW and Fiser J (2000) Minimizing binding errors using learned conjunctive features. *Neural Computation* 12:247-278.

[15] Ning P, Cui Y and Reeves DS (2002) Constructing attack scenarios through correlation of intrusion alerts. *9th ACM Conference on Computer and Communications Security*.

[16] Ning P and Xu D (2003) Learning attack strategies from intrusion alerts. *10th ACM Conference on Computer and Communications Security*.

[17] Tufte ER (1983) *The Visual Display of Quantitative Information*. Graphics Press.

[18] Valdes A and Skinner K (2001) Probabilistic alert correlation. *4th Workshop on Recent Advances in Intrusion Detection (RAID). LNCS*. Springer Verlag, Berlin, pp 54-68.

[19] Vert G, McConnell J and Frincke D (1998) Towards a mathematical model for intrusion. *21st National Information Systems Security Conference*, October 1998, pp 329-337.