An Empirical Analysis of NATE - Network Analysis of Anomalous Traffic Events*

Carol Taylor
Computer Science Department
University of Idaho, Moscow, Idaho 83844
208-885-4077
ctaylor@cs.uidaho.edu

Jim Alves-Foss
Computer Science Department
University of Idaho, Moscow, Idaho 83844
208-885-5676
jimaf@cs.uidaho.edu

ABSTRACT

This paper presents results of an empirical analysis of NATE (Network Analysis of Anomalous Traffic Events), a lightweight, anomaly based intrusion detection tool. Previous work was based on the simulated Lincoln Labs data set. Here, we show that NATE can operate under the constraints of real data inconsistencies. In addition, new TCP sampling and distance methods are presented. Differences between real and simulated data are discussed in the course of the analysis.

Keywords

Intrusion Detection, Statistics, Traffic Analysis

1. INTRODUCTION

The accelerating trend of computer security incidents appears unaffected by the increasing concern and attention to computer security from government, research and corporate groups. Reported computer security incidents have jumped from below 5000 in 1998 to 35,000 in 2001 [4]. These statistics suggest that securing computer systems against threats from intruders is a difficult problem not solvable in the near future. One answer to the computer security dilemma is intrusion detection. Intrusion Detection Systems (IDS) provide detection capability for intruders that penetrate other system security defenses. All IDS's utilize some type of monitoring to assess the system's state and determine the occurrence of an intrusion. One of the primary distinctions between IDS's is their detection scope. Commonly, ID systems focus on one host machine or multiple machines connected to a network. Network IDS's typically monitor network traffic as their data source while host IDS's utilize system information such as system or application logs.

New Security Paradigms Workshop '02, September 23-26, 2002, Virginia Beach, Virginia.

A 2000 CMU report [1] reviewed the state-of-the-art in ID and identified research needs for current and future ID development. An area in need of further research was the development of IDS's capable of monitoring high speed network connections. Today's network IDS's can't keep up with current traffic and as network speeds increase, the IDS performance gap will widen. Consequently, some IDS researchers have questioned the feasibility of future network monitoring. Another concern is the limited amount of administrator time available for monitoring networks. Smaller organizations typically lack dedicated computer security personnel and system administrators must assume the role of computer security officer. Consequently, an IDS should not require a large investment in time for configuration and management. Current network ID research emphasizes large, comprehensive solutions, but fails to account for the time required for IDS operation.

In NSPW 2001, we presented NATE, a low cost approach to network ID that addressed both high speed traffic monitoring and administrator time constraints. As presented, NATE was a minimalist approach to network ID. The idea was to create a light-weight monitoring tool that purposely ignores attacks buried in the packet payload and recognizes only those attacks that are detectable from packet header information. From the header information, a small number of attributes are selected to further streamline the system and reduce processing speed. NATE utilizes an anomaly based detection method eliminating the need for constant update of rules or signatures. Results from our initial investigation showed that NATE could identify TCP attacks that are detectable from header information such as most probes, scans and Denial of Service (DOS) types of attacks. Further results with both TCP and UDP protocols are presented in [28].

The data used to develop and test NATE came from the MIT Lincoln Labs data set [16] created for IDS evaluation. Several problems with this data set were noted [18] and researchers have begun to question the validity of results based on this data set. Since NATE was developed and tested with this simulated

* This work was funded in part by a grant #60NANB1D0116 from the National Institute of Standards and Technology (NIST)

data set, there was some concern that our previous success was due to the data and that results would not be repeatable under more realistic conditions. In order to address these concerns, we tested NATE with a real data set from an operational network (outside an academic setting). This paper presents our analysis with actual network data. We also report results from additional statistical distance and sampling methods.

We begin in Section 2 with a review of relevant research with an emphasis on recent developments in the area of statistical ID. In Section 3 NATE's features are discussed followed by a description of the data collection, analysis and attack identification methods in Section 4. The empirical analysis and discussion are presented in Section 5 and 6. The paper concludes with a summary and future research direction, Section 7.

2. CURRENT RESEARCH

Recent reviews of IDS's divide these systems into a much finer grained taxonomy than host verses network which is the more typical approach [3,7]. ID techniques span a wide range of methods including expert system, pattern matching, state transition analysis, neural network and statistics [7]. A broad classification of these detection techniques places them into either anomaly based or signature based methods. Anomaly based detection seeks to identify a normal system state and detects deviations from that state as signs of anomalous activity. Signature based detection identifies intrusions by comparing a current signature against known patterns, rules or states to recognize an intrusion.

Previously, we reviewed network and network/host systems that were then compared to NATE [27]. Here, we want to highlight statistical intrusion detection in order to place NATE's detection method in the context of other statistical approaches.

Traditionally, anomaly based detection was accomplished with statistics. Wisdom and Sense [29] and Haystack [31] used statistics to monitor changes in user behavior. NSM [10] used statistics along with rules to monitor LAN traffic. The Emerald IDS [20] statistical component was inherited from a previous SRI IDS, the Nides system [12]. The Nides statistical component set the standard for statistical based ID for a number of years. This method computes a historical distribution of continuous and categorical attributes which are updated over time. Deviations from historical norms are based on a chi-square like statistic. Ji-nao, a router based IDS also uses the Nides statistical component [31]. Most of the preceding methods update the measures continuously with new information.

A classification tree approach was developed in [5] to formulate a statistically derived rule set for classifying intrusive activity. The technique also uses network traffic header information but it is not clear how efficient the method would be under actual operation. Another recent technique utilizes conditional probability to determine the likelihood of anomalous behavior [8]. The method works by computing the likelihood of the nth call given n-1 previous calls. Yet, another recent statistical method analyzes system calls in privileged processes with discriminant analysis, a multivariate grouping technique [2]. This method appears to be quite efficient utilizing only 11 system calls to distinguish between normal and intrusive behavior.

Cluster analysis has previously been applied to intrusion detection [17,23]. Portnoy used cluster analysis to group network traffic based on a large number of traffic characteristics. This approach was similar to NATE but differs in several important characteristics [23]. While NATE creates clusters of normal behavior for anomaly detection, Portnoy forms clusters of both anomaly and normal behavior in order to match anomalous sessions. Their method also bases decisions on the frequency of anomalous vs. normal traffic and makes the assumption that normal traffic is more frequent than abnormal traffic. This assumption is not always valid depending on network conditions (i.e. during a flood event). In [17], cluster analysis is used to group machines from a large network based on similar traffic characteristics.

3. NATE'S FEATURES

NATE embodies a unique set of characteristics not previously encountered in ID solutions. Anomaly based detection was deliberately implemented in an effort to streamline system maintenance. An anomaly based approach should translate to fewer updates of rules or signatures as new attacks are discovered. The idea is that new attacks should be detected automatically. Another benefit of anomaly detection is that detection is not limited to known attacks but extends to previously unknown intrusions [7]. In contrast to most statistical anomaly based methods, the system administrator will not need to know normal system parameters [25, 20, 31] to configure the system. Self-configuration is possible as a result of automating the data collection and construction of the cluster database.

NATE seeks to minimize the amount of data needed for attack detection by measuring a small number of attributes. The attributes that appeared to best distinguish between normal and anomalous TCP sessions included the frequency of TCP P (Push) and Ack (Acknowledge) flags, the average and total number of bytes transferred, and the percentage of session control flags. In [27], we describe these attributes in more detail. This reduces the processing time which increases the efficiency for potential high-speed network monitoring. Another feature related to fast operation is a focused detection coverage for attacks that can be discovered from header information. Currently, this includes all probes, scans and many DOS types of attacks. Restricting the focus to just packet headers yields another benefit in that NATE can handle both regular and encrypted data.

4. METHODS

Previously, we discussed the statistical and sampling methods used in NATE's initial development. A brief summary of these methods is presented. New techniques including a different sampling strategy and an alternate distance metric are treated with more detail.

4.1 Cluster Analysis

The purpose of cluster analysis is to group data so that objects in a given cluster are similar to each other and dissimilar from

other clusters [13]. Our purpose in applying cluster analysis to network packet data is to form clusters of normal traffic in an effort to capture the normal network state. It is not known how many actual groups there are in TCP data since new TCP-based applications are constantly being introduced.

4.2 Sampling Strategy

Many studies involve populations that are too large to analyze. Sampling is the standard approach in statistical analysis for obtaining a subset that is representative of a larger population. Note that prior to sampling the population, extreme, high-valued TCP sessions will be eliminated as outliers. Outlier removal is standard with most statistical analysis and for cluster analysis is essential for obtaining a good cluster solution.

In sampling network data for anomaly based detection, it is important to collect the full range of network behavior. Otherwise, normal behavior could be mistaken for abnormal resulting in many false positives. The next two sections discuss two types of sampling strategies for collecting network data.

4.2.1 TCP Type Sampling

Sampling by TCP type[1] requires that each major traffic type be represented in the sample. A major TCP type is defined here to mean relative high frequency compared to less common types. Originally, a standard sample size calculation was computed for each major TCP type to insure that common types are adequately represented [27]. However, results from the standard sample size equation tended to yield samples that under-represented the less variable types and over represented the more variable types. Another sampling strategy is to include equal numbers of each TCP type to insure sufficient representation of all traffic found on the network. Thus, each major type will be randomly sampled at an equal rate.

4.2.2 Attribute Distribution Sampling

Our previously reported cluster analysis results suggest that TCP traffic doesn't cluster into distinct groups by type [27]. We thus adopted a sampling strategy that previously yielded good results with UDP data [28]. The entire population was first divided into groups based on 2 times[2] the standard deviation of the most variable attribute[3]. For example, if the most variable attribute is Total Packets, and the standard deviation of Total Packets is 30, then groups will be formed by dividing the records based on Total Packets values from 1 to 60, 61 to 120, 121 to 180 etc. Each group will be randomly sampled to include an equal number of records. The expectation is that systematic inclusion of records based on distribution of the most variable attribute will provide a more complete sample of network traffic.

4.3 Distance Measures

Distance measures quantify the dissimilarity between individual normal clusters and a potentially anomalous TCP session. The measures transform a vector of TCP session attributes into a single valued distance which can be assessed for significance.

The Mahalanobis distance is a measure based on the covariance matrix of the attributes and incorporates the relationships between attributes. Mahalanobis distance values can be compared to a chi-square distribution so that significance of the results can be assessed. This was discussed in [27].

While the Mahalanobis distance produced good results, there were some problems with the method since mapping to a chi-square distribution requires multivariate normality which is not always achievable. Another way to determine distance is with Euclidean distance. While Euclidean distance does not map to a known statistical distribution, significance of the values can be determined empirically by computing a natural bound on the values. The equation for the Euclidean distance [13] is,

$$dist(x,y) = \left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2}$$

where x and y are two attribute vectors representing TCP sessions and n is the number of attributes measured. For our use, x represents a new TCP session and y is the cluster mean from an individual cluster of TCP sessions.

Determining the natural bound of Euclidean distance values involves the use of Chebyshev's inequality [24]. The equation for this limit is,

$$Pr\{|x - \mu| \geq k\sigma\} \leq 1/k^2$$

where x is a random variable with mean, μ, and standard deviation, σ. Here, x is a single Euclidean distance, μ is the mean Euclidean distance values for a normal cluster and k is a multiplier that determines the significance level. Chebyshev's inequality sets the natural bound on the variability of the points within a given cluster and is computed separately for each cluster. The value produced is a probability that the value, x, comes from a population with mean, μ, and standard deviation, σ [24]. The Euclidean distance computed between a TCP session and a cluster can be compared to this bound to see if the distance is significant, i.e. outside the Chebyshev inequality bound. K can be set to approximate the typical significance of a known distribution. For example, if we set k = 4.47, then the probability is set to .05, $1/4.47^2$, which is the usual cut-off point for significance using an F test and the normal distribution [24].

5. EMPIRICAL ANALYSIS

Initial results from NATE were promising based on the simulated Lincoln Labs data. In order to confirm these results and show that the system will work on an actual network, we conducted a second empirical analysis using data captured from an operational network. Results from this analysis are presented in this section.

[1] TCP type is defined as the application that generated the TCP traffic, i.e. ftp generates ftp control traffic, port 20 and ftp data traffic, port 21.

[2] 2x the standard deviation was chosen since this is the error bound of the sample size equation.

[3] In taking multivariate samples, it is common to sample based on the most variable attribute.

5.1 Network Environment

The data was collected from a small functional network where the hosts perform web and e-mail server functions (Figure 1). A network firewall was active, mostly performing NAT[4] for several servers. Additionally, each server host had its own firewall, ipfirewall, which comes with the FreeBSD operating system. Host firewalls were also configured to allow only a specific set of traffic. Allowed traffic consisted mostly of https-secure http, ssh- secure shell and smtp-simple mail transport. Some http, unsecured web traffic was also allowed. A more complete discussion of the traffic is given in Section 5.3, Attack Screening. The operating system for all the hosts consisted of FreeBSD, a unix variant.

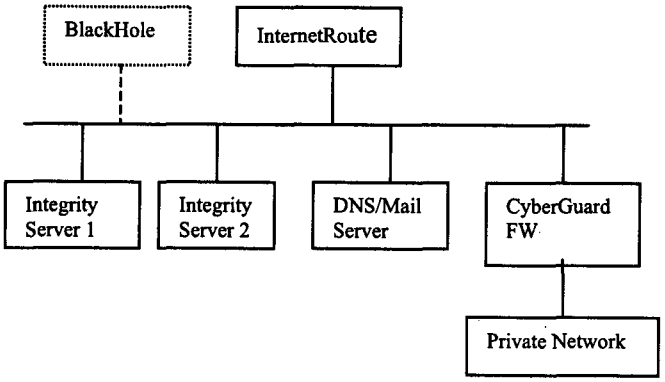

Figure 1. Network configuration

5.2 Data Collection

Data was collected over a consecutive 10 day period by BlackHole [19], a custom network sniffer built on top of BPF and libpcap. Because the output was in pcap format, tools such as tcpdump [11] could be used to read and filter the data. The sniffer collected all network data and stored it in 100 Mbyte files. A script was written to screen the files for TCP traffic which was aggregated into TCP sessions. A session is typically defined as all the packets between two unique source and destination port/IP addresses.

5.3 Attack Screening

One of the stated problems with anomaly based methods is the risk of incorporating anomalous behavior in with the normal data when establishing the normal state [7]. In utilizing simulated data, this was not a problem since embedded attacks were identified as part of the training data. However, real data potentially has anomalous data mixed in with normal data and must be screened prior to analysis of the normal data. Two types of anomalous sessions were of concern. The first type consists of the probes, scans and DOS types of data characterized by missing TCP session attributes and few packets. The second class of anomalous data are individual sessions that attempt to flood by sending large numbers of packets or bytes in order to overwhelm a given service.

[4] NAT stands for Network Address Translation and is done for internal IP addresses which get mapped to alternate addresses so internal machines are protected from outsiders.

Searching the data set for probes and scans turned out to be fairly easy by creating a script based on missing TCP attributes such as bytes, Push and Ack flags. Attacks that flood by sending an overload of bytes or packets within a single session will be filtered out in the normal outlier removal process[5]. This network had two active web servers configured to allow ssh, https, auth, and icmp traffic. Additionally, these machines could make DNS queries via UDP but not TCP. A separate machine served as the DNS/mail server and was allowed ssh, smtp, DNS via UDP, pop3 auth, and local network traffic. Network traffic outside this limited range was suspect for the web and mail servers. One machine functioned as a general purpose machine and had no well-defined security policy.

The individual firewalls were configured to disallow ftp and telnet connectivity so these connection attempts only show the first few packets since the rest of the session is cut off.

The results of traffic screening for this network were interesting yielding a large number of anomalous sessions. The frequency of these sessions is presented in Table 1.

By far, http traffic dominates contributing about 85% of the anomalous traffic. The remaining anomalous types included https at 5%, domain traffic at 2% and the remaining types at 1% or less. Examining the large valued sessions eliminated as outliers showed no obvious anomalous activity.

Table 1. Frequencies of anomalous network traffic

Traffic Type	Frequency %	Description
http	85	Web traffic
https	5	Secure web traffic
domain	2	Name server traffic
sunrpc	1	Remote proc. call
telnet	< 1	Remote connection
ftp	< 1	File transfer

Examination of the anomalous looking sessions showed that multiple types of scans were present in this data set. The most common type of scan was a Syn scan of all existing machines for a particular service. All existing machines were sent syn packets for http, https, dns, sunrpc, ftp and telnet among others. Variations of this scanning activity were seen including sending ack packets instead of syn packets and sending small amounts of data. The ack scan is an attempt to bypass firewall filtering and is a feature of the common nmap [20] scanning tool. Sending small amounts of data appears to be similarly motivated since sessions with zero bytes are easily filtered. Port scans were less common where multiple ports on one machine were queried. Stealth scans were common with time delays of seconds to hours.

Upon examination of this data, some of the anomalous appearing traffic was actually normal. A number of auth sessions turned out to be legitimate. These sessions originated from the internal mail server but matched exactly auth traffic

[5] The assumption is that an unusually high count of either bytes or packets will exceed most normal sessions and be eliminated as an outlier.

from external machines. It seems that the other internal hosts on the network were rejecting the auth traffic from the mail server so the auth sessions were never established. Solutions for anomalous appearing legitimate traffic will be discussed in Section 6.

5.4 Sampling Results

Sampling was accomplished two ways as outlined in Section 4.2. For this network, four TCP types dominated with frequencies presented in Table 2. As can be seen from the frequency distribution, https is overwhelmingly dominant followed by smtp, ssh and http.

Table 2. Frequencies of normal network traffic

Traffic Type	Frequency %	Description
https	94	Secure web traffic
smtp	4	Mail traffic
ssh	2	Secure shell
http	< 1	Web traffic

Each type was sampled at 60 data points for a total sample of 240 points. This data set is referred to as TypeSample. Dividing the population into groups based on the distribution of Total Packets, resulted in five groups. Each of these groups was also sampled at 60 creating a data set with 300 points called GroupSample.

5.5 Cluster Results

Results from the cluster analysis for the two data sets were similar each containing cluster solutions of 5 clusters. Cluster composition was mixed with clusters consisting of multiple TCP types. Differences between clusters appeared to be based solely on magnitudes of the session attributes and not on TCP type differences. This was true for both data sets. A possible reason for the lack of clustering by TCP type is presented in Section 6. Cluster distribution for each data set is presented in Figures 2 and 3.

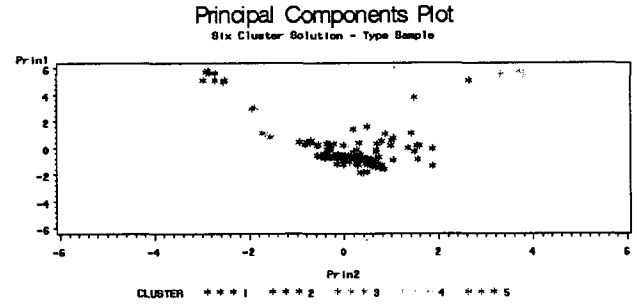

Figure 2. Cluster data distribution for TypeSample

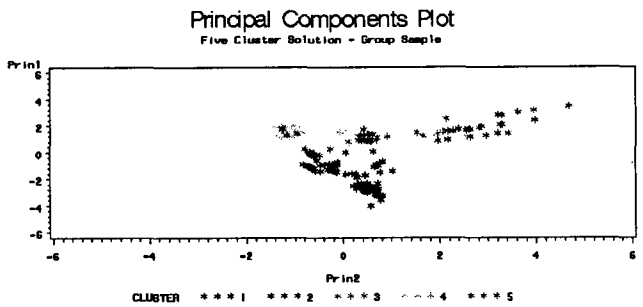

Figure 3. Cluster data distribution for GroupSample

While the plots are similar, the GroupSample plot, Figure 3, shows more separation of the data points than the TypeSample data set. When the data already forms natural groups clustering the data is more straightforward. Thus, the GroupSample data set produced a more even distribution of data among the clusters compared to the TypeSample data set which had most of its points assigned to Cluster 1 with few points distributed among the remaining four clusters.

5.6 Attack Recognition

After creating the normal cluster database for each data set, attack sessions were selected to test against these databases. From the attacks identified in Section 5.2, several were chosen for testing including a Portscan, Ipscan, Ack Portscan and Data[6] Portscan. Since no obvious flood attempts were noticed during the collection period, two DOS attacks from the Lincoln Labs data set, Neptune and Mailbomb, were included in order to test against this type of activity. Neptune, is an attack that attempts to flood the machines capacity to accept new TCP connections via half-open connection attempts [6, 14]. Mailbomb tries to flood the mail server by sending a lot of script-generated mail messages [14].

Both Euclidean and Mahalanobis distances were computed for each data set, GroupSample and TypeSample in order to compare results from both sampling methods and the distance measures. Tables 3 and 4 present results from the Mahalanobis distance tests for each data set. The tables include both attack sessions and normal data.

Table 3. Mahalanobis Distances for GroupSample Data Set

Type	Clust1	Clust2	Clust3	Clust4	Clust5
Portscan	281	>27118	>160650	>66050	>587882
Ipscan	256	>26887	>159752	>65862	>587901
Datascan	234	>26656	>158860	>65672	>587932
Ackscan	68	642146	>390663	>16158	>138241
Neptune	256	>26887	>159752	>65862	>587901
Mailbmb	13.3*	66881	469193	>20766	162987
https	2371	2370	114	4.6*	21602
ssh	3.7*	927	15155	93888	5331
smtp	4.9*	68431	497244	>22471	184608

* Indicates distance is not significant[7]

[6] A Data Portscan is defined as a Portscan where the attacker sends data in an effort to bypass potential filtering rules

[7] Distance translates to a Chi-squared distribution with 5 degrees of freedom, .001 significance level, equals 20.5

Table 4. Mahalanobis Distances for TypeSample Data Set

Type	Clust1	Clust2	Clust3	Clust4	Clust5
Portscan	681	>112709	>139081	>473376	39
Ipscan	631	>112406	>139003	>473385	39
Datascan	586	>112103	>138924	>473394	39
Ackscan	138	27747	326809	>113839	40
Neptune	631	>112406	>139003	>473385	39
Mailbmb	19.5*	36475	348528	>145099	36
https	2371	30115	487	108509	79
ssh	1.6*	53277	643863	>221000	23
smtp	4.9*	64285	861963	>264696	36

* Indicates distance is not significant

Examining the Mahalanobis distances, it appears that these results are similar to the results obtained from the Lincoln Labs data. The new attacks are significantly different from all clusters in both data sets. The two flood attacks also compare similarly to results from the Lincoln Labs data [27]. Neptune which resembles a Syn scan once the time element is removed, appears anomalous as can be seen from the significant Mahalanobis distances. However, Mailbomb appears to be normal when examined at the session level. Again, this particular attack floods by overloading the mail service with numerous mail sessions each of which appears normal.

Legitimate TCP sessions selected from the unsampled TCP data were also compared against the normal databases. There were some differences in the correct identification of normal TCP sessions between the two data sets. With the GroupSample data set, all normal sessions were correctly identified as demonstrated by insignificant Mahalanobis distances for one or more clusters (Table 5). For the TypeSample data set, one http normal session was misidentified since its distances are significantly different from all clusters (Table 6). Additional testing of normal sessions resulted in more misidentification of normal sessions. Normal sessions that appear anomalous are considered to be false positives and should be minimized to reduce the problem of sounding false alarms. The reason for the greater incidence of false positives with the TypeSample data set will be covered later in Section 6, Discussion. The normal sessions included in the tables, represented either extremely large or extremely small valued sessions in an effort to identify normal sessions that could cause false positives.

Euclidean distance results are presented in Tables 5 and 6. Results from this measure are nearly identical with the Mahlanobis distance.

Table 5. Euclidean Distances for the TypeSample Data Set

Type	Clust1	Clust2	Clust3	Clust4	Clust5
Portscan	214	948	320	782	571
Ipscan	214	947	319	779	570
Datascan	214	913	319	781	572
Ackscan	83*	949	243	741	523
Neptune	214	947	319	779	570
Mailbomb	13*	883	199	700	483
https	760	545	493	21*	349
ssh	159*	723	107*	439	348
smtp	45*	784	98*	618	406

*Indicates distance is not significant, < Chebechev inequality limit

One attack, the Ack Portscan is not anomalous for Cluster 1 in both data sets (Tables 5 and 6). However, the remaining scans and probes are all identified as anomalous with Euclidean distances outside of the Chebyshev bound. The two DOS attacks produced identical results to the Mahalanobis distance. Neptune is correctly identified as anomalous while Mailbomb appears to be normal as seen from the non-significant Euclidean distances. Euclidean distances for normal sessions were all non-significant for both data sets.

Table 6. Euclidean Distances for GroupSample Data Set

Type	Clust1	Clust2	Clust3	Clust4	Clust5
Portscan	194	357	533	922	689
Ipscan	194	357	532	921	688
Datascan	193	357	531	920	687
Ackscan	173*	264	477	886	645
Neptune	193	357	532	921	688
Mailbomb	15*	213	437	855	603
https	722	657	347	533	58*
ssh	117*	164	252	694	34
smtp	13*	112*	335	755	523

* Indicates distance is not significant, < Chebechev inequality limit

6. DISCUSSION

So far, our lightweight ID approach has yielded reasonable results where it appears that we can reliably distinguish scans, probes and DOS types of attacks from normal TCP traffic. This section discusses implications of the results from the analysis of real data.

Previous studies have suggested that TCP network traffic is distinguishable by type [22]. Consequently, our first sampling attempt treated each type as a separate group from which samples were taken. Yet, our cluster results contradicted the generally accepted notion that TCP traffic types are distinguishable. We observed that most of the clusters contained mixtures of traffic types as opposed to single type clusters. One explanation for the lack of TCP type grouping is the choice of attributes used to cluster the data. Attributes were selected that distinguished between normal and anomalous sessions. However, these attributes may not be suitable for discriminating between the various TCP traffic types. The selection of attributes and further characterization of TCP network traffic is a topic for further research.

Our initial cluster results led us to try sampling based on a strategy of ignoring TCP types and sampling the traffic as a single population. This appeared to provide a more even distribution of data points among the clusters but did not affect the attack recognition results, which were nearly identical for the two data sets. However the sampling method does appear to affect formation of the cluster database which relates to the accuracy of normal traffic identification. Normal traffic can fall outside the bounds of all the clusters and be mistaken for anomalous traffic. When the data shows little separation with most points lumped together as in TypeSample, cluster creation is more difficult with cluster boundaries created arbitrarily. One solution is to choose a more complex clustering methods such as the Twostage Density method from SAS. This method was tried along with increasing the number

of clusters which decreased the cluster variability. This eliminated the problem with misidentified normal sessions. Another solution is to sample by attribute distribution, which appears to create some natural groupings. Thus, a simpler cluster approach with fewer clusters produced no false positives from the GroupSample database.

Sampling network traffic is an area in need of further study, but these initial results suggest that sampling according to attribute distribution is a good alternative to sampling by TCP type. Besides creating natural groups, another advantage that attribute distribution sampling has over TCP type sampling is a much simpler sampling process since the sample doesn't have to include every TCP type. As noted with the Lincoln Labs data set, obtaining adequate samples of the least frequent types can be difficult [27].

Mahalanobis distance previously produced good results in identifying both TCP and UDP attacks [27,28]. However, mapping the distance to a chi-square distribution requires a normally distributed data set. Normal distribution is not always attainable which prompted the investigation of an alternate distance measure with fewer requirements. Euclidean distance appeared to be a good choice since an empirical distribution could be calculated via the Chebyshev inequality. Yet, results were less robust for attack session identification than with Mahalanobis distance. Since the basis for Euclidean distance is to calculate the distance of every point from its cluster mean, the measure appears to be sensitive to outlier points. The more widely dispersed clusters will produce a larger standard deviation and consequently a larger bound for normal behavior. This bound can exceed the distance of an anomalous vector. This occurred in Cluster 1 of both data sets. Comparing results from both distance measures showed that Mahalanobis distance was slightly better at distinguishing between normal and attack sessions. Mahalanobis distance incorporates the covariance structure of the attributes which adds information. For normal TCP sessions where few attributes are missing, medium to strong relationships exist between the attributes. Several attributes are correlated above .85. Scan and probe attack sessions are characterized by mostly missing attributes and a lack of attribute relationships. Thus, it is worthwhile to explore attribute transformation or some other method of approximating normality in order to satisfy the requirements for using the chi-square distribution.

To date, we have used both simulated and real data in developing NATE. This experience provides us with several useful insights regarding IDS development. The advantage of a simulated data set is total control of the data. Attacks can be injected at known intervals and manipulated to suit individual research needs. Another benefit from using a public, widely distributed data set is it represents a standard against which IDS's can be compared. A number of studies based their research on this data set, which in theory allows them to compare results [5,8,9,15]. Yet, given that problems were identified in the Lincoln data [18], relying on it as a data source may not be desirable for ID development. The inherent danger in relying on a simulated data set for any type of research is it may not be representative of the real word. For us, the real data obtained from a small special purpose network was substantially more variable than the simulated data. Screening the data for attacks highlighted numerous instances of normal TCP sessions that appeared anomalous. The presence of legitimate traffic that appears anomalous can be resolved in several ways. Rules can be added during both screening and operation to filter out this traffic. Or, additional attributes can be measured to distinguish between similar normal and anomalous traffic. The important point is that if we had developed NATE based only on the simulated data we would have obtained a distorted view of the data regularity. Subsequent use of NATE in a real environment would have resulted in many false positives. Thus, for IDS development it is important to not only conduct empirical tests but ideally to test the IDS's under conditions or with data that will be encountered in the intended operating environment.

7. CONCLUSION AND FUTURE WORK

This paper presented our continuing work with NATE, a lightweight anomaly based ID tool. A summary of our conclusions from this research include the following observations:

- TCP type can be ignored[8] in obtaining a representative sample of network traffic. Of more importance are the ranges of the measured attributes.
- Using a distance measure that captures relationships between TCP session attributes adds information since the attributes are correlated to some extent.
- Validating an IDS with real data or under an actual operating environment should be an important step in IDS development.
- Encrypted traffic types such as ssh and https can still be analyzed since header information is not encrypted.
- Sampling appears to be important in creating a good cluster database which in turn affects NATE's power to discriminate between normal and anomalous sessions.

At this point, we need to ask what does a tool such as NATE add to the security of a system and is it enough to continue investigating this particular approach? A number of security products already provide screening of anomalous traffic. Many firewalls and most popular routers allow users to set filtering rules. The key distinction that can be made between these products and NATE appears to be anomaly based detection. Firewalls and routers filter by rules, which translate to attacks that can bypass the device by targeting services not in the rule set. Anomaly based techniques will function in the event of new exploit attempts. NATE can contribute to attack detection by filtering those attacks that evade the rules of firewalls or routers. NATE can sit on either side of a firewall and provide additional filtering capabilities. Inside the firewall, NATE can detect internal machines that have been compromised and are now attacking other internal and external hosts. Outside the firewall, NATE can catch incoming traffic that would have bypassed the firewall. Currently, we feel that there is enough promise shown by NATE to continue pursuing this research. We envision NATE as just one of many tools or probes that can be used by system administrators to enhance the security of their systems.

[8] Given our selection of TCP header attributes.

Future research needs to address several unresolved areas. TCP traffic characterization is an area in need of further study. Basic research needs to be done to capture essential qualities of network traffic. Parameters relating to traffic variability in terms of traffic types are unknown for TCP and UDP traffic. Most traffic studies have been conducted for performance and not specifically aimed at understanding network traffic from a security perspective [22].

Clustering the raw data without performing the cleanup step may prove a viable alternative to our present approach. Anomalous appearing data should cluster together enabling future identification via a match with these anomalous traffic clusters.

Currently, NATE monitors few attributes. This set should be expanded to see if additional attributes result in better attack detection. One obvious attribute to add is time since DOS attacks are typically noticed in relation to some time element. Time will allow the detection of individual sessions that appear normal such as Mailbomb but flood by sending multiple sessions. Other attributes extracted from packet headers needs to be researched for their detection potential.

Nate's performance in terms of false negatives and false positives is an area of future research. More testing with a wider range of attack and normal data needs to be done.

Another area to investigate is the examination of some packet payload features. While a full payload analysis would take too long for network traffic, examination of the payload would allow NATE to detect more serious attacks and may not significantly slow down the detection process.

Finally, an actual prototype of NATE needs to be constructed and deployed on a high band width network to assess NATE's realtime performance under actual working conditions.

8. ACKNOWLEDGEMENTS

We would like to thank the reviewers and all of the participants of NSPW 2002 for all of their suggestions for this paper. Most of the comments were helpful and assisted with the production of a much better paper.

9. REFERENCES

[1] J. Allen et al. State of the practice intrusion detection technologies. Carnegie Mellon, SEI, Tech Report, CMU/SEI-99-TR-028, ESC-99-028, January 2000.

[2] M. Asaka, O. Takefumi, T. Inoue, S. Okazawa, S. Goto. A new intrusion detection method based on discriminant analysis. *IEICE, Transactions of Information and Systems*, Vol. E84-D, No. 5, May 2001.

[3] S. Axelsson. Intrusion detection systems: A survey and taxonomy. Technical Report 98-17, Department of Computer Engineering, Chalmers University. 1999.

[4] Cert, www.cert.org, 2001.

[5] Chapple, M.J. Network intrusion detection utilizing classification trees. Masters Thesis, Computer Science Department, University of Idaho, 2000.

[6] T. Daniels and E. Spafford. Identification of host audit data to detect attacks on low-level IP vulnerabilities. COAST Lab., Purdue University, COAST TR98/10, 1998.

[7] H. Debar, M. Dacier, A. Wespi. Towards a taxonomy of intrusion detection systems. *Computer Networks*, 31 (1999) pg. 805-822.

[8] E. Eskin, M. Miller, Z. Zhong, G. Yi, W. Lee, S. Stolfo. Adaptive model generation of intrusion detection. In *Proceedings of the ACM CCS Workshop on Intrusion Detection and Prevention*. Athens, Greece, 2000.

[9] A. K. Ghosh, A. Schwartzbard, M. Schatz. Learning program behavior profiles for intrusion detection. In *Proceedings of the Workshop on Intrusion Detection and Network Monitoring, April 9-12, 1999,* Santa Clara, CA, Usenix Assoc.

[10] L. T. Heberlein, G. V. Dias, K. N. Levitt, B. Mukherjee, J. Wood, and D. Wolber.A network security monitor. In *Proceedings of the IEEE Symposium on Research in Security and Privacy,* Oakland, CA, April 1990.

[11] V. Jacobson, C. Leres, and S. McCanne. *tcpdump.* LBNL, University of California, June 1997, www.tcpdump.org.

[12] H. S. Javitz, and A. Valdes. The NIDES statistical component: description and justification. Tech. Report, Computer Science Lab., SRI-Int., Menlo Park , CA, March 1994.

[13] L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Series in Probability and Mathematical Statistics, John Wiley and Sons, Inc., 1990.

[14] K. Kendell. A database of computer attacks for the evaluation of intrusion detection systems. Masters Thesis, MIT, June 1999

[15] W. Lee. A data mining and CIDF based approach for detecting novel and distributed intrusions. In Proceedings of the 3rd International Workshop on Recent Advances in Intrusion Detection, Oct 2-4, 2000, Toulouse, France, 2000.

[16] R. Lippmann and M. Zissman. Intrusion detection technical evaluation – 1998 project summary. http://www.darpa.mit/ito.

[17] D. Marchette. A statistical method for profiling network traffic. In *Proceedings of the Workshop on Intrusion Detection and Network Monitoring, April 9-12, 1999,* Santa Clara, Calif.

[18] J. McHugh. 1998 Lincoln Lab intrusion detection evaluation a critique. *In Proceedings of the 3rd International Workshop on Recent Advances in Intrusion Detection, Oct 2-4, 2000,* Toulouse, France, 2000.

[19] K. Monroe. BlackIce.

[20] P. G. Neumann, P. A. Poras. Experiences with Emerald to date. *Proceedings of 1st Usenix Workshop on Intrusion Detection and Network Monitoring,* Santa Clara, CA, Apr. 11-12, 1999.

[21] nmap. www.insecure.org. 2000.

[22] V. Paxon. Empirically derived analytic models of wide-area TCP connections. IEEE Transactions on Networking, Vol. 2, No. 4, 1994.

[23] L. Portnoy. Intrusion detection with unlabelled data using clustering. Undergraduate Thesis. Columbia University, Dept. of Computer Science, 2000.

[24] J. A. Rice. Mathematical Statistics and Data Analysis. Wadsworth Publ. Co., 1995.

[25] S. E. Smaha. Haystack: an intrusion detection system. *Proceedings IEEE Fourth Aerospace Computer Science Applications Conference,* Orlando, FL, Dec. 1988.

[26] K.M.C. Tan and B.S. Collie. Detection and classification of TCP/IP Network Services, IEEE Network, 1997.

[27] C. Taylor and J. Alves-Foss. NATE – Network analysis of anomalous traffic events, a low-cost approach. In Proceedings of New Paradigms in Security Workshop, Cloudcroft, New Mexico, Sept. 2001.

[28] C. Taylor. NATE – Network Analysis of Anomalous Traffic Events – A Low-cost Approach. Masters Thesis, Computer Science Department, University of Idaho, 2001.

[29] H.S. Vaccaro and G.E. Liepins. Detection of anomalous computer session activity. *In Proceedings of the 1989 IEEE Symp. on Sec. and Privacy. pg. 280-289, Oakland, CA 1-3* May, 1989.

[30] A. Valdes and K. Skinner. Adaptive, model-based monitoring for cyber attack detection. In *Proceedings of the 3rd International Workshop on Recent Advances in Intrusion Detection, Oct 2-4, 2000, Toulouse,* France, 2000.

[31] S.F. Wu, H.C. Chang, F. Jou, F. Wang, F. Gong, C. Sargor, D. Au, R. Cleaveland. Ji Nao: Design and implementation of a scalable intrusion detection system for the OSPF routing protocol. www.anr.mcnc.org, 1999.