Using Self-Organizing Maps to Build an Attack Map for Forensic Analysis

H. Güneş Kayacık, A. Nur Zincir-Heywood
Dalhousie University, Faculty of Computer Science,
6050 University Avenue, Halifax, Nova Scotia. B3H 1W5
{kayacik, zincir}@cs.dal.ca

Abstract

In this work, we focus on developing behavioral models of known attacks to help security experts to identify the similarities between attacks. Furthermore, these attack behavior models can be used to analyze zero-day attacks, which security experts have limited knowledge of. To this end, a Self Organizing Feature Map (SOM) is employed to model the relationship between known attacks and U-Matrix representation is used to create a two dimensional topological map of known attacks. The approach is evaluated on KDD'99 data set. Results show that attacks with similar behavior patterns are placed together on the map. Moreover, when new attacks are presented, SOM assigned similar labels to the attacks that are newer versions of the known attacks.

Keywords

Intrusion detection, KDD 99 intrusion detection datasets, Self-Organizing Map, Neural Networks

1 Introduction

Working in a networked environment exposes us to new threats every day. We discover many attacks both in terms of volume and variety. (Distributed) Denial of Service (DoS) and zero-day attacks (internet worms etc.) are the two most popular of such attacks in today's Internet [16]. To defend against the attacks, security management operations are devised to protect the confidentiality, authentication, integrity and availability of a network. Within the context of security management operations, a lifecycle exists in which new attacks are detected and consequently, suitable defense mechanisms against these new attacks are formulated. Such defensive mechanisms can be categorized in two groups: static and dynamic.

Static defense mechanisms are intended to provide barriers to attacks. Keeping operating systems and other software up-to-date and deploying firewalls at entry points, using cryptographic techniques for secrecy are examples of static defense solutions. Frequent software updates can remove the software vulnerabilities, which are susceptible to exploits. Firewalls provide access control at the entry point; they therefore function in much the same way as a physical gate on a house.

On the other hand, dynamic defense mechanisms are analogous to burglar alarms, which monitor the premises to find evidence of break-ins. These operations aim to catch the attacks and log information about the incidents such as source and nature of the attack. Intrusion detection systems and forensic analysis systems are examples of dynamic defense mechanisms. Thus, dynamic defense mechanisms are especially important in order to understand new attacks, how they work so that such information can be reported and immediate response can be taken to prevent similar attacks that might happen in the future. We consider our approach as a part of dynamic defense mechanisms.

In this work, our objective is to build attack behavior models in order to understand how known attacks are related so that when a new attack is reported we can immediately see if it is related to any known attacks or not. A typical case where such an approach can be useful is when a new attack is discovered. Through forensic analysis, a system/ network administrator will identify which known attack group new attack is most similar to (if at all) because, such an information will be useful in order to decide the most suitable detection/ prevention technique to employ for such an attack. In other words, from a forensic analysis standpoint, we want to find out if there is any relationship between known attacks, i.e. do they share common traits, show similar behavior etc. This type of information not only will help to categorize known attacks but also can help to categorize new attacks.

In order to achieve this, we employed Kohonen's Self Organizing Feature Maps (SOM) [6] to build a model of known attacks and visualize them in a topologically ordered 2-dimensions.

The remainder of the paper is organized as follows. Related work on visualization and forensic analysis of attacks is presented in Section 2. Section 3 provides the methodology of the work. Results are reported in Section 4 and Conclusions are drawn in Section 5.

2 Related Work

Early work on network traffic analysis and visualization employed graphs [7, 8, 9] to visualize the nodes on a network where lines indicate network connections between two hosts. In such approaches, an anomalous traffic was identified by unusual connections from one source or to one destination. Different methods were employed to build visual representations of network state however main emphasis was on the flow information, in other words, destination and source. The main contribution of our work is a methodology that relies on attack behavior rather than flow information. That is to say, instead of looking for abnormal flows, we build behavioral models of known attacks and use the models to discover the similarities between the known attacks. Moreover such a model is crucial to help system administrators to analyze the zero-day attacks.

Self Organizing Maps [6] have been applied to many problems including intrusion detection, pattern recognition and modeling. In particular, exploratory data analysis performed by Kaski et al. [10] bears resemblances to our methodology. Based on a number of welfare indicators, Kaski et al built an SOM, which illustrates the (economic) relationships between countries in a two dimensional projection. Their results indicated that the countries with similar standards of living in OECD countries were mapped to same neighborhood on the SOM.

3 Methodology

Although not without drawbacks [3], KDD 99 intrusion detection [2, 5] (or DARPA 98/99 [1]) datasets are the most comprehensive source of attacks containing 38 different attack types. Datasets are imbalanced, that is to say certain attacks have more samples than the others, therefore the dataset needs to be balanced to eliminate any bias towards majority classes. We trained a Self-Organizing Map on the balanced training data and employed the labels (i.e. attack types) from the same dataset to assign labels to neurons.

3.1 KDD dataset

The KDD 99 intrusion detection datasets are based on the 1998 DARPA initiative, which provides designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies [1]. To do so, a simulation is made of a factitious military network consisting of three 'target' machines running various operating systems and services. Additional three machines are then used to spoof different IP addresses to generate traffic. Finally, there is a sniffer that records all network traffic using the TCP dump format. The total simulated period is seven weeks. Normal connections are created to profile that expected in a military network and attacks fall into one of four categories: User to Root; Remote to Local; Denial of Service; and Probe.

- *Denial of Service (dos):* Attacker tries to prevent legitimate users from using a service.
- *Remote to Local (r2l):* Attacker does not have an account on the victim machine, hence tries to gain access.
- *User to Root (u2r):* Attacker has local access to the victim machine and tries to gain super user privileges.
- *Probe:* Attacker tries to gain information about the target host.

In 1999, the original TCP dump files were preprocessed for utilization in the Intrusion Detection System benchmark of the ACM International Knowledge Discovery and Data Mining Tools Competition [2]. To do so, packet information in the TCP dump file is summarized into connections. Specifically, "a connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows from a source IP address to a target IP address under some well defined protocol" [2]. This process is completed using the Bro IDS [4], resulting in 41 features for each connection. Features are grouped into four categories:

- *Basic Features:* Basic features can be derived from packet headers without inspecting the payload. Basic features are the first six features.
- *Content Features*: Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts;
- *Time-based Traffic Features*: These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over a 2 second interval;
- *Host-based Traffic Features*: Utilize a historical window estimated over the number of connections – in this case 100 – instead of time. Host based features are therefore designed to assess attacks, which span intervals longer than 2 seconds.

The KDD 99 intrusion detection benchmark consists of three components. In the International Knowledge Discovery and Data Mining Tools Competition, only 10% KDD dataset is employed for the purpose of training [5]. This dataset contains 22 attack types and is a more concise version of the Whole KDD dataset. It contains more examples of attacks than normal connections and the attack types are not represented equally. Because of their nature, denial of service attacks account for the majority of the dataset (Table 1). On the other hand the Corrected KDD dataset provides a dataset with different statistical distributions than either 10% KDD or Whole KDD and contains 14 additional attacks. Since 10% KDD is employed as the training set in the original competition, we performed our analysis on the 10% KDD dataset.

Table 1. Basic characteristics of the KDD 99 intrusion detection datasets in terms of number of samples

Dataset	DoS	Probe	u2r	r2l	Normal
10% KDD	391458	4107	52	1126	97277
Corrected KDD	229853	4166	70	16347	60593

3.2 Balancing the Dataset

When dealing with imbalanced datasets where one class represents a large number of samples in training data, the identification accuracy of the model can be affected since the learning algorithm will encounter more samples from the majority class. As detailed in Table 1, KDD 99 datasets are imbalanced where two denial of service attacks (namely, Smurf and Neptune) make up ~70% of the dataset. In order to remove any learning bias towards majority classes, random re-sampling [14] is applied to the 10% KDD. To do so, we randomly select 1000 samples from each class, if the class has less than 1000 samples, it is over sampled (with duplicates) otherwise it is under sampled. Except the two aforementioned denial of service attacks, all attacks have close to or less then 1000 samples hence under sampling applies to only 2 of 24 attacks in the training set.

3.3 Self-Organizing Map

Kohonen's Self-Organizing Feature Map (SOM) algorithm is an unsupervised learning algorithm in which an initially 'soft' competition takes place to provide a topological arrangement between neurons at convergence [6]. The learning process is summarized as follows,

1. Assign random values to the network weights, w_{ij};
2. Present an input pattern, $\boldsymbol{x}$, in this case a series of taps taken from the shift register providing the 'reconstruction' state space on which the SOM is to provide a suitable quantized approximation.
3. Calculate the distance between pattern, $\boldsymbol{x}$, and each neuron weight $\boldsymbol{w}_j$, and therefore identify the winning neuron, or

$$d = \min_j \left\{ \left\| x - w_j \right\| \right\}$$

where $\|\cdot\|$ is the Euclidean norm and $\boldsymbol{w}_j$ is the weight vector of neuron j;

4. Adjust all weights in the neighborhood of the winning neuron, or

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) K(j,t) \left\{ x_i(t) - w_{ij}(t) \right\}$$

where $\eta(t)$ is the learning rate at epoch t; and $K(j, t)$ is a suitable neighborhood function;

5. Repeat steps (2) – (4) until the convergence criterion is satisfied.

Following convergence, presentation of an input vector, $\boldsymbol{x}$, results in a corresponding output vector, $\boldsymbol{d}$, the Euclidian distance between each neuron and input. The neuron with the smallest distance represents the winning or best matching neuron, step (3). The best matching neuron also defines a neighborhood of next nearest neighboring neurons. Once the maps are trained, it is this concept of a best matching node that is used to facilitate the labeling of the map.

3.4 Assigning Labels

In order to assign labels to SOM neurons we maintain a hit score matrix *h(i, j)* where *i* is the neuron index and *j* is the class label index. As neuron *i* gets selected as winning neuron for more input samples from class *j*, *h(i, j)* score increases. After the SOM is trained, inputs from the training set are presented to determine the first 5 winning neurons. Number 5 is selected empirically in order to capture the behavior of the neighborhood rather than a single neuron. Hits score of the winning neuron for the given label (i.e. *h(i, j)*) is incremented by a value inversely proportional to its rank (i.e. 1/*rank* where $5 \geq rank \geq 1$) hence assigning a higher score to a neuron that is closer to the input pattern in terms of Euclidian distance. Neurons are labeled with the class label, which has the highest hit score.

4 Results

As discussed before, the objective of this work is to build a two dimensional topological map of known attacks to help us better understand the known attacks (i.e. the attacks the SOM is trained on) and new attacks (i.e. the attacks the SOM has not seen during training) such as a zero-day attack where the attack information is limited. Table 2 details the training parameters that were used throughout our analysis.

Table 2. SOM training parameters

Parameter	Rough Training	Fine Tuning
Initial η	0.5	0.05
SOM size	10 x 10	
η decay scheme	$f(\text{epoch}^{-1})$	
Epoch Limit	4,000	
Neighborhood Parameters		
Initial Size	4	2
Function	Gaussian	
Relation	Hexagonal	

4.1 Attack Map Visualization

One of the advantages of SOM over other unsupervised learning algorithms such as clustering is its ability to create a two dimensional visualization of high dimensional cluster structures. To visualize the cluster structure of high dimensional weight vectors of SOM neurons, a graphic display called U-Matrix [11] is used.

In U-Matrix visualization, shades of gray are used to show the distances between weight vectors of the neurons. Since in our analysis we are pairing attacks with neurons, if the distance between two neurons (attacks) is small then it is shown with light shades whereas if the distance is large, darker shade is used. In other words, light areas can be viewed as clusters (of attacks) whereas dark areas are cluster separators. U-Matrix representation employs extra hexagons between neurons to show the topology of the clusters.

Figure 1 shows the U-Matrix of the SOM with assigned labels.

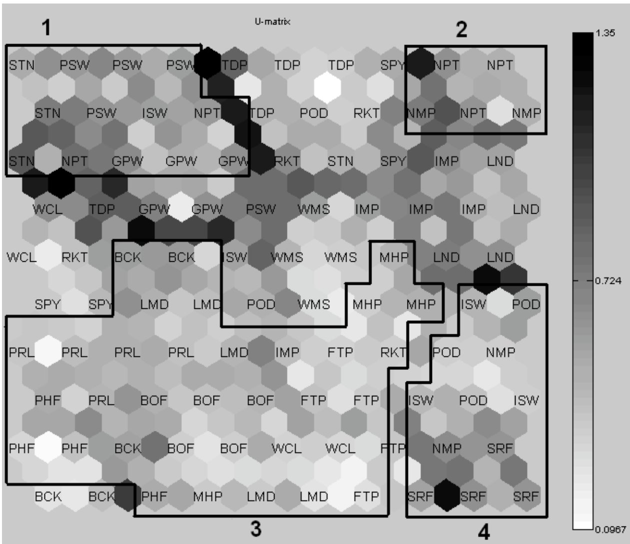

Figure 1. "Attack Map" of the 24 attacks in 10% KDD dataset with clustering shown in shades of gray

Four regions emerge from the SOM shown in Figure 1. Upper left (Region 1), upper right (Region 2), lower left (Region 3) and lower right (Region 4). Dataset labels are converted to three letter acronyms (Table 4) for better visualization. It is important to note that attacks can be assigned neurons on different regions of the SOM since different stages of an attack can exhibit different behavior.

Region 1 is also named as "focused attack group" because it contains attacks that focus on a single host. Of the attacks that appear in this region, Satan (STN) is a vulnerability scanner containing a set of probes, which can be applied to discover vulnerabilities on a victim. Similarly, a portsweep (PSW) looks for open ports on a victim. Neptune (NPT) attack also appears in this region because as a result of this denial of service attack the victim receives a large number of SYN packets. Guess password (GPW) attack uses a brute force technique to discover the password of a local account by trying every word in a "password dictionary", therefore opening many connections to the victim.

Region 2 is the "SYN anomaly group". Within this group, Neptune is a denial of service attack where the attacker sends spoofed SYN packets, which causes victim to initiate partially open connections. Since the number of open TCP connections is limited, server refuses new connections after the limit is reached, hence refusing legitimate users. Nmap (NMP) is an open source port-scanning tool, which aims to detect the open ports on a host and to determine operating system versions. Port scans can be performed using different techniques including SYN scanning.

Region 3, which is the largest Region, is "host-based attack group". This region contains 9 attacks all of which allows the attacker to gain regular or super user privileges. The techniques vary between exploiting bugs, using configuration errors and abusing operating system or application features. It is difficult to further analyze this region because the documentation in DARPA 98 dataset does not provide information on how the attacks are deployed. However the common characteristic of the host based attacks is that they grant a user access to the attacker, which is identifiable because in KDD datasets the 16th and 18th features provide the number of root and regular user shell prompts respectively (further analyzed in Figure 3).

Region 4 is the "ICMP anomaly group". Nmap, which is also associated with Region 2, is also associated with this region because it provides ICMP scanning as well as SYN scanning. Smurf (SRF) is a denial of service attack which involves sending ICMP echo request packets to broadcast addresses with the source address spoofed to victims IP address. Any host that listens to this broadcast sends ICMP echo packets to the victim effectively overwhelming the victim. Similarly ping of death (POD) attack involves sending long ICMP packets to the victim. Moreover, IP sweep is implemented to send ICMP packets to every possible IP address on the victim network.

The neurons that are outside of the four regions do not show any apparent commonalities but it is worth to mention that within a 41 dimensional space, a commonality can exist. Apart from that, generally neighboring neurons are associated with the same attack or similar attacks.

Figure 2 shows the attack map according to the outcome of the attacks. Kendall [13] proposed an attack taxonomy based on the (1) initial privilege of the attacker (2) attack methodology and (3) the outcome of the attack. In Figure 2, four outcomes of an attack are considered:

- *Probe (P):* The objective is to gather information about the victim.
- *Deny (D):* Attack prevents victim to function properly.
- *Super-User Access (S):* Attack provides super user privileges to the attacker on the victim host.
- *User Access (U):* Attack provides regular user access to the victim machine where attacker otherwise did not have.

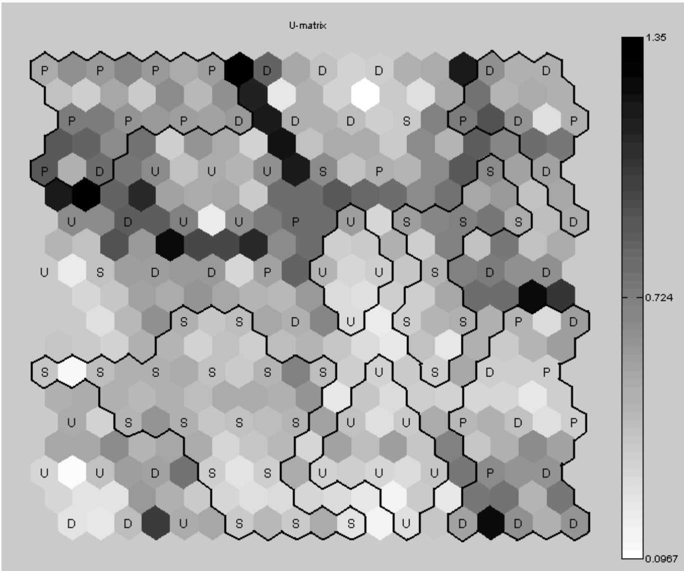

Figure 2. "Attack Map" of the 24 attacks in 10% KDD dataset with clustering shown in shades of gray

Figure 2 shows that probe and denial of service attacks are commonly clustered together (namely Regions 1, 2 and 4 from Figure 1). This is due to the fact that both type of attacks have observable impacts on network traffic such as abnormally short / long packets or a sudden increase in the number of connections that the victim receives. Attacks that affect the network state can be identified by basic features and time based features, which is summarized in Section 2.1. Kayacik et al. [12] provided a list of 41 features and discussed their relevance to intrusion detection.

Furthermore Figure 2 shows that attacks that gain (super) user privileges are clustered together (namely the Region 4, which contains the content based attacks in Figure 1). These attacks do not initiate abnormal network connections but host-based features or content-based features in KDD 99 datasets provide sufficient identification information. In short, the clusters seen in Figure 2 supports the clusters identified in Figure 1, where Region 4 is the cluster of content based attacks and Regions 1-3 are the clusters of different network based attacks.

On the other hand, Figure 3 shows the analysis of the above U-Matrix from the perspective of 41 different features used in the KDD 99 competition. Similar to Figures 1 and 2, light areas represent dense regions – or where little or no change is observed – where dark areas show sparse areas – where the feature exhibits variety. It can be seen that the features which cause/affect certain behaviors also affect similar parts of the map. In case of the densely populated features (e.g. typically a light shade dominant U-Matrix in Figure 3), the region where the change in feature is observed is shaded darker. For example in Region 3, where attacks involve gaining (super) user access on victim host, "logged.in" feature exhibits variety, hence shaded darker whereas for the Regions 1, 2 and 4, it is always zero (i.e. no login to host takes place). Similarly, number of root shells ("root.shell") and number of file modifications ("file.accs") only exhibit change in Region 3. Furthermore, U-Matrix of SYN errors "syn.err" and "srv.syn.err" indicate that Region 2 is the only region where SYN anomalies are observed. Additionally, The neurons, which are assigned to password guessing attacks, stand out in "login.fail" U-Matrix. Features "out.cmd" and "is.hot" features are constant (set to 0) in the training set therefore corresponding U-Matrices do not show any change in color.

4.2 Identification Accuracy

As discussed previously, our objective is to identify the commonalities in the known attacks and provide a forensic analysis tool to analyze new attacks in terms of their similarity to known attacks. Therefore, the identification accuracy results reported in this section should not be interpreted as a detection rate because detection implies separating normal behavior from attacks whereas the model, which is encapsulated by the SOM here, only contains attack information and is not intended for separating normal behavior from attacks. Therefore identification accuracy should be considered as a measure of quality of the model.

Having established the purpose of calculating identification accuracy, now we can describe what we mean by identification accuracy in this context. Identification accuracy of an attack implies recognizing an attack based on a given input to the SOM. Thus, it is calculated by presenting the input to the SOM and finding the winning neuron. If the winning neuron has the same label with the input, it is considered a correct identification. Table 3 summarizes the overall identification accuracy on the attack patterns in 10% KDD and the Corrected dataset.

Table 3. Identification accuracy on training and test set (on attack patterns)

Dataset	Identification accuracy
10% KDD	99.48%
Corrected KDD	89.80%

Table 4 and 5 details the identification accuracy for each attack type that appears in the training set for 10% KDD and Corrected datasets. The detailed results indicate that although the dataset is balanced, the minority attacks (i.e. attacks that involve less than 100 connections) are more difficult to identify because they are content based.

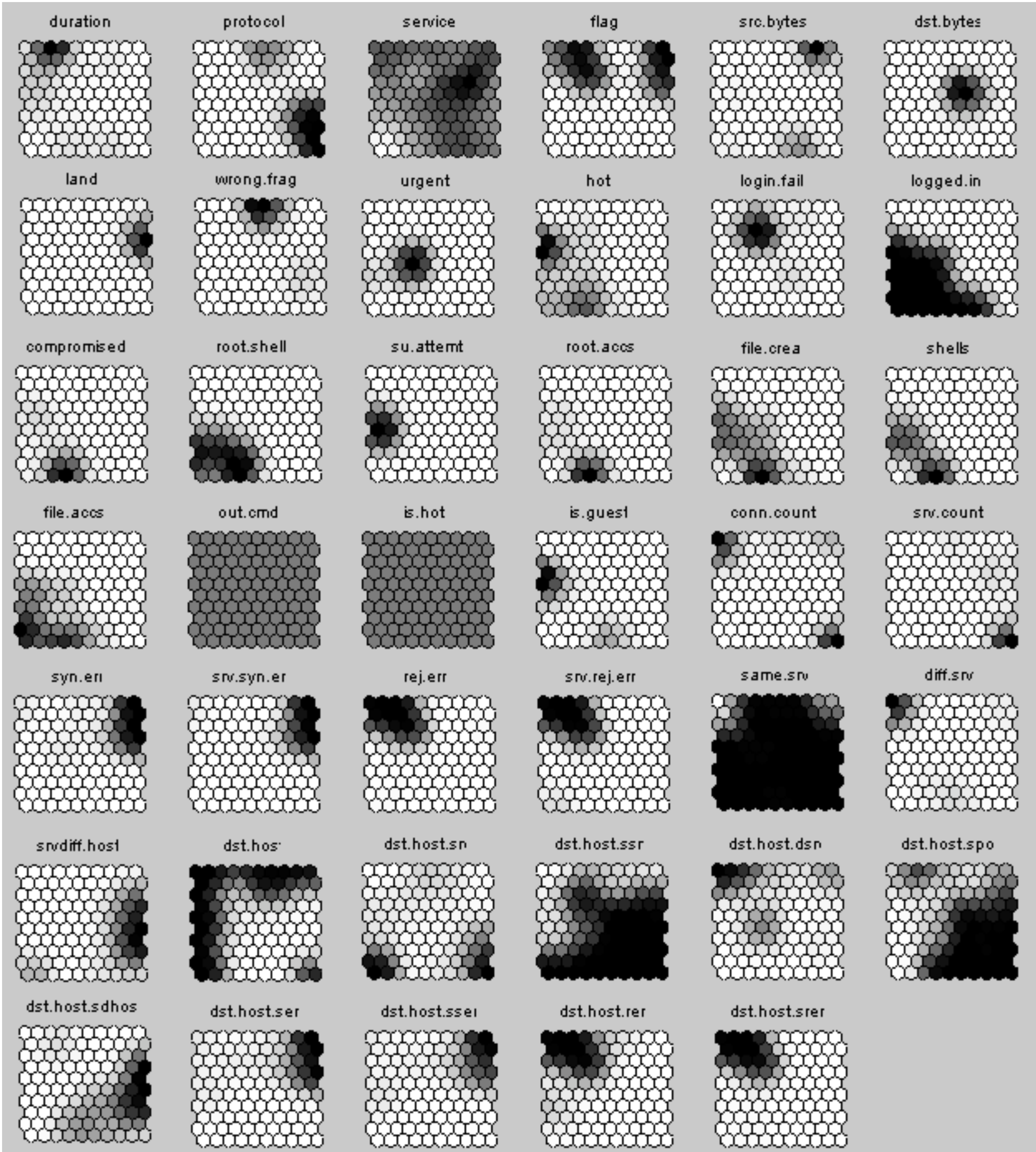

Figure 3. Application of U-Matrix visualization to 41 features separately. Light areas represent high degree of clustering and dark areas represent sparse regions within the feature space.

Table 4. Identification accuracy on the 10% KDD dataset

Attack Type	Label	Total	Accuracy
perl.	PRL	3	100.00%
land.	LND	21	100.00%
smurf.	SRF	280790	99.99%
neptune.	NPT	107201	99.98%
teardrop.	TDP	979	99.80%
guess_passwd.	GPW	53	94.34%
portsweep.	PSW	1040	90.29%
satan.	STN	1589	89.18%
back.	BCK	2203	88.24%
imap.	IMP	12	75.00%
phf.	PHF	4	75.00%
warezmaster.	WMS	20	75.00%
rootkit.	RKT	10	70.00%
pod.	POD	264	67.42%
buffer_overflow.	BOF	30	60.00%
ipsweep.	ISW	1247	59.74%
loadmodule.	LMD	9	55.56%
ftp_write.	FTP	8	50.00%
spy.	SPY	2	50.00%
nmap.	NMP	231	48.48%
multihop.	MHP	7	42.86%
warezclient.	WCL	1020	30.00%

Table 5. Identification accuracy of known attacks on the Corrected dataset

Attack Type	Label	Total	Accuracy	Label Assigned
back.	BCK	1098	100.00%	BCK
nmap.	NMP	84	100.00%	NMP
smurf.	SRF	164091	99.99%	SRF
neptune.	NPT	58001	99.71%	NPT
portsweep.	PSW	354	86.72%	PSW
teardrop.	TDP	12	83.33%	TDP
satan.	STN	1633	81.14%	STN
pod.	POD	87	75.86%	POD
rootkit.	RKT	13	61.54%	RKT
land.	LND	9	55.56%	LND
perl.	PRL	2	50.00%	PRL
phf.	PHF	2	50.00%	PHF
ipsweep.	ISW	306	20.59%	POD
guess_passwd.	GPW	4367	0.27%	BCK
buffer_overflow.	BOF	22	0.00%	PRL
loadmodule.	LMD	2	0.00%	PRL
ftp_write.	FTP	3	0.00%	WCL
imap.	IMP	1	0.00%	NPT
multihop.	MHP	18	0.00%	RKT
warezmaster.	WMS	1602	0.00%	FTP

Corrected dataset contains 14 additional attacks, which did not appear in the training set. We assign labels to these "new" attacks as the way we calculate the identification accuracy. Table 6 provides the list of "new" attacks and the label assigned to each one of them. Saint, which is the more recent version of Satan, is identified as Satan by the SOM. Furthermore Mscan attack, which exhaustively scans IP addresses to discover vulnerabilities, is associated with Neptune that exhibits the same pattern (i.e. high number of connections). Although the reason for the association is not clear, other attacks, which are generally content-based, are associated with the known content-based attack behavior only. Thus, results show that the attacks with similar behavior are placed in the same or neighboring clusters on the map.

Table 6. Assigned labels for new attacks in Corrected dataset

Attack Type	Label Assigned
mailbomb.	BCK
xterm.	BOF
apache2.	GPW
mscan.	NPT
httptunnel.	NPT
xsnoop.	PRL
sqlattack.	PRL
xlock.	RKT
sendmail.	RKT
udpstorm.	RKT
processtable.	RKT
ps.	RKT
worm.	SPY
snmpgetattack.	STN
named.	STN
saint.	STN
snmpguess.	STN

5 Conclusion

In this paper, we employed Kohonen's Self-Organizing Feature Map (SOM) algorithm to build a topological model of known attacks for forensic analysis of suspicious network traffic. SOM is selected as the analysis technique because it places similar patterns to contiguous locations in the output space (i.e. neighboring neurons) and provides projection and visualization options for high dimensional data. The main focus of the SOM, in this work, is to create a summary of known attacks, while preserving topological relationships. Consequently, not only SOM forms a forensic analysis tool for post mortem analysis of the known attacks but also can be employed to analyze new attacks or suspicious network behavior.

Results demonstrate that attacks with similar behavior patterns are placed together on the SOM. Four regions emerged on the SOM, Regions 2 and 4 are perceptive to abnormal use of network features such ICMP and SYN flag in TCP. Region 1 summarizes the abnormal traffic that a host receives either in terms of volume or number of open connections both of which are the cause of denial of service attacks or attacks that involve exhaustive search. Region 3 is the largest region on the SOM and covers most of the content-based attacks. The common trait of the attacks in Region 3 is that they all involve interaction with the victim on the application layer such as sending malicious URL to the web server or exploiting a vulnerable application on the host to gain access.

Results on the test data indicate that known attacks are identified with a relatively high identification accuracy although SOM employs unsupervised learning. Furthermore when attacks that did not appear on the training set (i.e. new attacks) are presented, SOM assigned similar labels to the attacks that are newer versions of known attacks, such as identifying the new version of a vulnerability scanner with its predecessor and recognizing anomalies that appear on network traffic. However, label assignments for new content-based attacks are not always as effective but at least, they are all assigned with a content-based attack label. This indicates that the SOM needs more information to identify the specific attack type but it can provide an approximate identification (i.e. network based attack versus content based attack) using existing features. For instance most of the denial of service and probe attacks would not result in a (super) user shell where most of the content-based attacks would.

We employed attack data from KDD 99 intrusion detection datasets because, despite its drawbacks [3], which are tied to the original data source (i.e. DARPA 98 and 99 datasets), it is the most comprehensive source of attacks. Future work will include collecting attack data from a live network by employing Honeypots [15] and using different features to characterize attacks.

Acknowledgments

This work was supported in part by Killam, NSERC and CFI. All research was conducted at the NIMS Laboratory, http://www.cs.dal.ca/projectx/.

References

[1] The 1998 intrusion detection off-line evaluation plan. MIT Lincoln Lab., Information Systems Technology Group. http://www.11.mit.edu/IST/ideval/docs/1998/id98-eval-11.txt, 25 March 1998.

[2] Knowledge discovery in databases DARPA archive. Task Description. http://www.kdd.ics.uci.edu/databases/kddcup99/task.html

[3] J. McHugh, "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," ACM Transactions on Information and System Security, 3(4), pp. 262-294, 2001.

[4] Paxson V., "Bro: A System for Detecting Network Intruders in Real-Time", Computer Networks, 31(23-24), pp. 2435-2463, 14 Dec. 1999.

[5] S. Hettich, S.D. Bay, The UCI KDD Archive. Irvine, CA: University of California, Department of Information and Computer Science, http://kdd.ics.uci.edu, 1999.

[6] T. Kohonen, Self-Organizing Maps. 3rd Ed., Springer-Verlag, ISBN 3-540-67921-9, 2000.

[7] R. Becker, S. Eick, and A. Wilks, "Visualizing network data", IEEE Transactions on Visualization and Computer Graphics, vol. 1, pp. 16-28, March 1995.

[8] K. Cox and S. Eick, "Case study: 3d displays of Internet traffic", in Proceedings of Information Visualization (INFOVIS), IEEE Computer Society, pp. 129-131, Oct. 1995.

[9] K. Abdullah, C. Lee, G. Conti and J. Copeland; "Visualizing Network Data for Intrusion Detection" IEEE Information Assurance Workshop (IAW) June 2002.

[10] S. Kaski & T. Kohonen. Exploratory Data Analysis by the Self-Organizing Map: Structures of Welfare and Poverty in the World, Proceedings of the Third International Conference on Neural Networks in the Capital Markets, World Scientific, Singapore, 1995, 498-507.

[11] Ultsch A. and Siemon H. "Kohonen's self-organizing feature maps for exploratory data analysis", In Proceedings of the International Neural Network Conference (INNC'90), Dordrecht, Netherlands, pages 305–308. Kluwer, 1990.

[12] Kayacik H. G., Zincir-Heywood A. N., Heywood M. I., "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets", Proceedings of the Third Annual Conference on Privacy, Security and Trust (PST-2005), October 2005.

[13] K. Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems", S.M. Thesis, MIT Department of Electrical Engineering and Computer Science, June 1999. http://citeseer.ist.psu.edu/kendall99database.html

[14] Han J., Kamber M., "Data Mining: Concepts and Techniques", Morgan Kaufmann, 2000, ISBN 7-04-010041, Ch. 5.

[15] Spitzner L., "The Honeynet Project: Trapping the Hackers", IEEE Security & Privacy March-April 2003 (Vol. 1, No. 2) pp. 15-23

[16] Symantec Internet Security Threat Report, Trends for July 05–December 05, Volume IX, Published March 2006 http://www.symantec.com/enterprise/threatreport/index.jsp