Author(s): Beqiri, Elidon; Lee, Sin Wee; Draganova, Chrisina.
Article title: A Neural Network Approach for Intrusion Detection Systems
Year of publication: 2010
Citation: Beqiri, E.; Lee. S. W.; Draganova, C. and Palmer-Brown, D. (2010) 'A Neural Network Approach for Intrusion Detection Systems', *5th Conference in Advances in Computing and Technology* (London, United Kingdom, 27th Jan), pp. 209 -217.
Link to published version: http://www.uel.ac.uk/act/proceedings/index.htm

A NEURAL NETWORK APPROACH FOR INTRUSION DETECTION SYSTEMS

Elidon Beqiri, Sin Wee Lee, Chrisina Draganova, Dominic Palmer-Brown
School of Computing, Information Technology and Engineering
University of East London
Docklands Campus
University Way
London E16 2RD
e.*beqiri@uel.ac.uk, s.w.lee@uel.ac.uk, c.draganova@uel.ac.uk*
d.palmer-brown@londonmet.ac.uk

Abstract

Intrusion detection systems, alongside firewalls and gateways, represent the first line of defense against computer network attacks. There are various commercial or open source intrusion detection systems in the market; nevertheless they do not perform well in various situations including novel attacks, user activity detection, generating in some cases false positive or negative alerts. The reason behind such performance is probably due to the implementation of merely signature based checks and a high degree of dependence on human interaction. On the other hand, a neural network approach might be the right one to tackle these issues. Neural networks have already been applied successfully to solve many problems related to pattern recognition, data mining, data compression and research is still underway with regards to intrusion detection systems. Unsupervised learning and fast network convergence are some features that can be integrated into an IDS system using neural networks. The networks can be designed to process a variety of data, although there are some constraints regarding input formatting. For this reason, data encoding represents a challenging task in the integration process since it needs to be optimised for the IDS domain. This paper will discuss the integration of IDS and neural networks, including data encoding and performance issues.

Keyword: Neural network, snap-drift, intrusion detection system, encoding, performance

1. Introduction

Computer networks and electronic communications have changed considerably almost every aspect of our daily lives, business, work and entertainment. They have influenced positively research and development, information sharing and globalization. As the dependency on these technologies is increasing, a range of new and old electronic attacks, vulnerabilities and intrusions is thriving. This is mainly due to the availability of attacking tools, automation and action at distance (Schneier, 2004). Therefore, protecting computer networks against malicious attacks becomes really important since even single intrusions can result in loss, corruption or destruction of electronic property.

Various technologies such as firewalls, antivirus packages, honeypots, and intrusion detection systems have been developed to shield computer systems against such electronic attacks. Amongst all, intrusion detection systems (IDS) attempt to achieve information security, by monitoring, analyzing and detecting suspicious intrusions both at the host or network level

(Rehman, 2003). IDS can be considered as the "burglar alarms" adapted to computer networks (Plante, 2004) since they use similar techniques to detect system integrity breaches.

2. Intrusion detection systems

The idea behind intrusion detection systems was initially introduced in 1980 by James Anderson (Anderson, 1980). He was the first to propose the idea of using audit trails to track computer misuse and match patterns of use behaviour. In 1987, Dorothy Denning proposed the first intrusion detection model which became the baseline for most of the intrusion detection systems of today (Denning, 1987). Since then several IDS approaches were introduced and implemented either for commercial or educational use.

An intrusion detection system fundamentally is a computer program (or set of programs) that collects and analyses a range of parameters or metrics related to computer networks so that it can ascertain if the security is compromised or not. The used parameters vary according to IDS models; however they can be classified in one of these categories: event counters, time intervals and resource measurements (Pervez at al, 2006). Event counters consist of occurrences of specific events over a period of time (ex. login attempts, incorrect login attempts etc); time intervals consist of intervals between events while resource management includes the expenditure of CPU time, number of records written to a database, number of files transmitted over a network etc (Pervez at al, 2006). IDS systems might also make use of other parameters including program signatures, which essentially are well known lines of code associated with malicious software or attacks.

IDS use probes or sensors installed in computer networks to passively sniff the data passing through; once the data is gathered a sets of programs known as the IDS engine carries the analysis against known attack signatures, thresholds or patterns of behavior (Werlinger at al, 2008). Subsequently, according to the result of the analysis the right course of action is taken by generating alerts or ignoring events. Generally, IDS deploy three distinct methodologies to detect possible intrusions: anomaly detection, misuse detection and hybrid detection.

a) Anomaly detection: the IDS respond if a deviation from a previously defined computer system state is detected. The system must be configured manually or automatically in advance with a set a parameters and values in order to create a knowledge base state, which is considered to be normal (Allen at al, 2000). IDS analyze system event streams, authentication logs, file access and other data to detect deviation from pre-established thresholds.

b) Misuse detection: intrusion detection is performed by comparing attack behaviors used to penetrate systems, against recorded user activity (Cannady at al, 1998). Signatures of well known hacking tools are usually stored in a database and then used by the engine during the processing stage for occurrences. Misuse detection approach is used by most of the commercial intrusion detection systems.

In terms of architecture, IDS can be classified as follow: 1) host-based IDS ,2) network-based IDS and vulnerability–assessment tools. Host-based attempts to

unveil intrusions on individual machines ,network-based evaluate information captured from network communications while vulnerability assessment IDS detect vulnerabilities on internal networks and firewalls (Planquet, 2001).

2.1 Issues with intrusion detection systems

An exploration of commercial IDS reveals that most of the products do not perform well in at least one of these categories: false positives, vulnerability to new attacks, continuous human interaction and anomaly detection.

False positives: Currently available IDS systems quite often misinterpret genuine user actions as malicious therefore producing false positives (false alerts). High rates of false positives are not productive because of the costs associated with them (administration, resources etc).

Vulnerability to new attacks: Most commercial IDS systems operate using misuse detection techniques. Misuse detection, also known as the signature-based method, makes use of well-known attack signatures. Attack signatures are developed by examining known malicious intrusions, analyzing their code, the content of IP packets and various parameters with the main objective of creating rules that automatically detect their presence. A typical misuse detection rule checks IP packets contents for occurrences of known attack strings

Continuous human interaction: The application of computer systems on almost every single scenario requires human interaction. In the case of currently present IDS systems human interaction on daily basis is fundamental to guarantee normal operation. IDS systems capture and analyze daily vast amounts of IP packets and generate thousands of possible alerts which are then assessed and evaluated by the user. Commercial IDS systems lack autonomy in operation and updating.

Anomaly detection not implemented: The majority of available IDS systems deploy misuse detection techniques to identify intrusion events based on well-known attack signatures. Anomaly detection, if implemented in conjunction with the misuse approach, would provide better recognition of novel intrusion attacks, decrease false positives and possibly increase overall system autonomy.

False negatives: On some occasions IDS systems fail to identify intrusion events or system data alteration. The term “false negative” is used to define real intrusion events not reported as such by the defence system (in this instance the intrusion detection system). Attackers make use of freely available packet crafting tools to fool IDS and compromise the system. A typical example is the use of data fragmentation to defeat IDS systems that do not perform seamless state checks (Northcutt at al, 2003).

3. Neural networks

Neural networks terminology refers to the cluster of neurons that function or act together to solve a particular task and process information. These networks are also capable of learning through supervision or independently. Artificial neural networks (ANN) as processing models are inspired by the way nervous system work and they attempt to implement in computer systems neuron like capabilities. Three layers are present in a typical ANN: input layer, hidden layer and output layer. Each layer is

composed of one or more nodes (neurons) and communication paths between them (Smith, 1998). All layers connected together form a network of nodes (or neurons). Typically information flows from the input to the output layer, although in some ANN architectures a feedback flow is present. The input layer represents the stimulus or information forwarded to the network, while the output layer is the final product of the neural processing. Input layer nodes often carry out hidden relationships amongst them producing "hidden" nodes. The hidden nodes and the interaction weight between input nodes compose the hidden layer.

The performance of neural networks depends on the architecture, algorithms and learning model chosen to collect and process data. Neural networks main features:

a) Architecture: Layer feed forward, multiple layer feed-forward and recurrent networks. Single layer networks have only one layer of input connections while recurrent networks use multiple layers and back propagation for learning (Hagan at al, 1996).

b) Learning algorithms: There is a variety of algorithms used for learning including: error correction learning, Hebbian learning, competitive learning, self organizing maps, back propagation ((Hagan at al, 1996), and snap-drift (Palmer-Brown at al, 2004)

1. c) Learning model: Supervised or unsupervised. Supervised models have been the mainstream of neural development for some time. The training data consist of many pairs of input/output training patterns and the learning process relies on assistance (Kung,1993).While in the learning phase the neural network learn the desired output for a given input. Multiple layer perceptron (MLP) (some reference needed e.g. D.E. Rumelhart, G.E. Hinton, and R. J. Williams, "Learning representations by back-propagation errors", Nature, vol. 323pp. 533-536, 1986.

) algorithm is used often with supervised models. The self-organizing map (SOM) algorithm is associated frequently with unsupervised models (Planquet, 2001). (reference to Kohonen)

3.1 Application of Neural Networks for Intrusion Detection Systems

In 1998 two researchers of UBILAB labs, used neural networks to detect intrusions and perform clustering of network traffic. They deployed an IDS based on a Self Organizing Map (SOM) neural network, associated to a visual approach of network traffic to achieve their objective (Girardin at al, 1998). This type of IDS performed anomaly detection checks against firewall logs and network traffic events; SOM was used to project network events in a visual format allowing the administrator to identify possible intrusions. This model managed to detect successfully a range of attacks including network scanning, IP spoofing, FTP password guessing (Planquet, 2001).

Multi-Layer Perception model and the Lucky Bucket algorithm were used by RST researchers for anomaly and misuse detection. Their IDS performed checked on program behaviour profiles which were build by monitoring system calls made by programs (Planquet, 2001).The Lucky

Bucket algorithm was used to store in memory recent abnormal events through counter management. IDS used the DARPA database (a database containing well known attack signatures) for misuse detection.

A Multi-Level Perceptron (two layers) neural network capable of detecting misuse intrusion detection on root-privilege attacks was deployed by MIT labs. Back propagation facilitated learning by detecting neural network weights. The model included a k input node, 2k hidden notes and 2 outputs for normal or attack state. The IDS searched for attack-specific keywords in the network traffic to detect intrusions (Planquet, 2001). The application of the neural network model increased the detection rate up to 80%, reduced false alarms and even managed to detect new attacks (Cunningham at al, 1999).

Georgia University researchers introduced a misuse detection system that used a combination of Self-Organizing Maps (SOM) and Multi-Level Perceptron (MLP) .The MLP/SOM prototype consisted of: 9 input layers, 4 fully tied layers and 2 output nodes representing normal and attack states (Planquet, 2001). The neural network used a feed-forward model with back propagation for learning (Cannady at al, 1999).

Hierarchical Intrusion Detection (HIDE) is another proposed model that employs neural networks for anomaly detection. HIDE makes use of statistical models and neural network classifiers to detect attacks (Zhang at al, 2001). The IDS architecture is hierarchical, with several tiers of intrusion detection agents.

Neural network models of back propagation (BP) and perceptron back propagation (PBH) were tested and selected for the proposed statistical anomaly intrusion detection system (Zhang at al,2001).This model does not deal with misuse detection and false positives.

South Korean researchers at Yonsei University have used evolutionary learning neural networks (ENN) to improve anomaly detection performance based on learning program's behaviour (Han at al,2004). Their model made use of system-call audit data to build ENN normal behaviour profiles. One neural network was used per program with 10 input nodes, 15 hidden and 2 output nodes (normal and attack). Anomaly detection performance reached almost 100% with very low false positive rates, whereas the structure and the weights of neural network were learned simultaneously.

In 2007, Stefano Zanero of the PMTU presented a study on unsupervised learning for anomaly intrusion detection (Zanero, 2007) and proposed a model that uses host anomaly detectors. The anomaly detector analysis system calls arguments and behaviours. Markovian modelling, clustering and behaviour identifiers were used also. The intrusion detections system used an unsupervised payload clustering and classification techniques to find anomalies.

4. Proposed approach

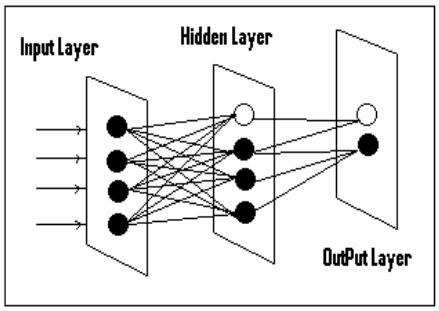
Although the above mentioned neural network approaches contribute seriously to the intrusion detection domain, a final approach that combines together misuse and anomaly detection, is far from being provided. Snap-drift, a novel neural network learning algorithm developed by Palmer-Brown and Lee (Palmer-Brown at al, 2004), might be a good candidate to tackle IDS issues. Snap-drift has already been used in the study domains of phonetic feature discovery of speech in waveforms, cluster

analysis and phrase recognition (Palmer-Brown at al, 2004). This is a novel approach for real time learning and mapping of new patterns which makes it suitable for use in frequently changing environments such as computer networks. Snap-drift algorithm will alter between minimalist learning when network performance is down and cautious learning when network is performing well; the detection system will be able to detect new attacks and learn patterns of behaviour.

The "snap" component is based on a modified form of ART while "drift" is based on Learning Vector Quantization (Palmer-Brown at al, 2004). Adoptive Resonance Theory (ART) , developed by Grossberg and Carpenter , is an unsupervised learning approach that achieves well at pattern matching (Carpenter at al, 1998). The Learning Vector Quantization (LVQ) (some reference needed T. Kohonen T., "Improved vesrions of learning vector quantization". In 1990 *Proc. Int. Joint Conf. Neural Networks*, vol.1, pp. 545-550.) approach exploits the underlying structure of input vectors to achieve data compression. It is a supervised learning technique that uses class information (Haykin, 1999).The two components are combined within a semi-supervised learning system that shifts its learning style whenever it receives a drop in the performance feedback. This learning algorithm can be explained better with this equation:

W= α (Fast Learning ART) + σ (LVQ)

The values α and σ, can be modified according to changes in performance (Palmer-Brown at al, 2004).



(Neural network diagram)

The neural network component based on the snap-drift learning algorithm can be integrated in an IDS model that already performs misuse detection. The neural network component will deal with anomaly detection, identification of attack patterns and learning so that new threats are detected.

4.1 Data encoding and formatting

Data encoding is often required for a variety of reasons including saving space, better understanding, or interpretation. This is also true in the case of IDS domain and in particular our selected data set. The data set that will be used as input by the neural network component of the IDS will be extracted from IP packets collected by the packet capturing engine. These IP packets data fields with be processed by snap-drift to identify patterns of attach or possible malicious activity:

Data set		
IP field	Data type	Length
Source Port	Decimal	>= 5 digits
Destination port	Decimal	>= 5 digits
Protocol	ASCI	>= 4 characters
TCP header length	Decimal	>= 5 digits
Sequence No	Hexadecimal	>= 8 characters
Acknoledgement No	Hexadecimal	>= 8 characters
TCP flag	ASCI	>= 4 characters
TTL	Decimal	>= 5 digits
IP ID	Decimal	>= 5 digits
Window Size	Decimal	>= 5 digits

(Data set table)

The selected neural network algorithm (snap-drift) performs calculations and learning based on binary input format; therefore the selected data set must be encoded and formatted to binary before use. Decimal and hexadecimal input must be converted to binary, while an encoding scheme must be selected for the Protocol and TCP flag data fields. TCP, UDP and ICMP are the transport layer protocols used randomly on top of the IP protocol. The data collected at the Protocol section might probably be encoded in this format: TCP: 00000001, UDP: 00000011, ICMP: 00000100. The TCP flag field encoded schema is presented in the table below. The basic TCP flags like ACK or PUSH are presented by a selected binary value, while the combinations of various flags by a binary adding of basic TCP flags.

TCP flag encoding			
Flag	Value	Flag	Value
ACK	00010000	F , A	00010001
SYN	00000010	R , A	00010100
FIN	00000001	S , F	00000011
URG	00100000	S , P	00001010
PUSH	00001000	S , F , P	00001011
RST	00000100	S , F , R	00000111
NULL	00000000	F , P	00001001
F , U , P	00101001	F , R	00000101
S , A	00010010	P , A	00011000

(TCP flag encoding table)

Another problem that might arise with input data is formatting. The collected and encoded data must be formatted to provide consistency and simplicity. Consistency might be achieved by padding with zeros the data fields in order to get equal length for each field. For example, some of the selected data set fields (such as Sequence No) will require a 32 bit binary representation, while other fields (such as TCP header length) only 8 bit one. Padding data set fields with zeros to achieve a standard 32 bit encoding seems like a necessary step to achieve consistency. Furthermore, the 32 bit binary fields must be formatted with space delimiters (such as commas, space or dots) since the algorithm is designed to operate on binary (pattern of zeros and ones). Encoding and formatting data uniformly simplifies processes and reduce calculation times. One of the other challenges concerning input is data set length. As mentioned above a typical standardised data field would contain 32 bits; data length will be affect to a certain extent feeding input to snap-drift and calculation times, especially if the intrusion detection system operates in a very busy environment.

4.2 Performance Issues

The purpose of the proposed neural network component is to deal with the above identified IDS issues. Although, the integration of this component will likely improve detection, it is wise to take in consideration possible performance issues caused by this process. First of all, the process of adding another layer (neural network) to the existing system will increase IDS complexity. This will require a careful design and implementation so that the overall performance is not affected. Another possible performance issue is related to the selected data set. Complex calculations on long data sets might affect detection time. Detection time is a really important factor, since malicious activities should be detected ideally in real time. Furthermore, some of the preferred attacking strategies against IDS systems are based on packet flooding in order to confuse detection systems. In this context, the added component should not affect in any way the overall detection or reaction speed.

The introduction of the neural network component might improve system self learning amongst other features. On the other hand, the neural network component will require tuning especially in the early stages of the integration. As such, overall system performance might be affected during the training stage.

Conclusion

A neural network approach might improve intrusion detection system performance and deal with the existing issues. However, a careful design and selection of data set is required so that overall performance is not affected. Data encoding, formatting and performance issues are some of the topics that require particular attention during the design of the neural network and the integration process. The selected learning neural network algorithm (snap-drift) seems to fulfil the selection requirements, because it performs well quickly changing environments similar to computer networks.

References:

Allen, J.,Christie, A.,Fithen, W.,McHugh, J.,Pickel, J.,Stoner, E.,2000. State of the practice of intrusion detection technologies, Available at: http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr028.pdf

Anderson, J., 1980. Computer security threat monitoring and surveillance, Available at: http://csrc.nist.gov/publications/history/ande80.pdf

Cannady, J., 1998. Artificial neural network for misuse detection, Available at: http://csrc.nist.gov/nissc/1998/proceedings/paperF13.pdf

Cannady, J., Mahaffey, J., 1999. The application of artificial neural networks to misuse detection, Available at: http://freeworld.thc.org/root/docs/intrusion_detection/nids/Application-of-ANN-to-Misuse-Detection.pdf

Carpenter, G. Grossberg, S. 1998, Adaptive Resonance Theory, Available at: http://cns-web.bu.edu/Profiles/Grossberg/CarGro2003HBTNN2.pdf

Cunningham, R .,Lippmann , R .,1999. Improving intrusion detection performance using keyword selection and neural networks, Available at: http://www.raid-symposium.org/raid99/PAPERS/Lippmann1.pdf

Denning, D., 1987. An intrusion-detection model, IEEE Transactions on software engineering, v.13 n.2, p.222-232, Available at: http://ieeexplore.ieee.org

Girardin, L., Brodbeck , D., 1998 . A visual approach for monitoring logs, Available at http://www.ubilab.org/publications/index.html

Hagan, T., Demuth, H ., Beale, M., 1996. Neural network design, PWS Publishing, USA

Han, S., Kim, K.,Cho, S., 2004. Evolutionary learning program's behavior in neural networks for anomaly detection, Available at: http://www.springerlink.com/content/g807h4u4q3700529/

Haykin, S, 1999. Neural Networks, A Comprehensive Foundation, Prentice Hall Publishing, USA, page 466

Kung, S., 1993. Digital neural networks, Prentice Hall, USA (page 27)

Northcutt, S.,Novak, J., 2003, Network Intrusion Detection ,New Riders Publishing ,USA(Page 53)

Palmer-Brown, D.,Lee, S., 2004. Continuous reinforced snap-drift learning in a neural architecture for proxylet selection in active computer networks, Available at: http://www.scs-europe.net/services/esm2004/pdf/esm-45.pdf

Pervez, S.,Ahmad, I., Akram, A., Swati, S., 2006. A comparative analysis of artificial neural network technologies in intrusion detection systems, Available at: http://www.labplan.ufsc.br/congressos/WSEAS/papers/517-423.pdf

Rehman, R., 2003. Intrusion detection systems with Snort, Pearson Education .Inc, USA

Schneier, B., 2004. Secrets and Lies: Digital security in a networked world, John Wiley and Sons .Inc, USA (page 30)

Smith, S., 1998 . The Scientist & Engineer's Guide to Digital Signal Processing, California Technical Publishing ,USA (page)

Werlinger, R., Hawkey, K.,Muldner, K.,Jaferiaan, P.,Beznosov,K., 2008. The challenges of using an intrusion detection system: is it worth the effort? Available at: http://cups.cs.cmu.edu/soups/2008/proceedings/p107Werlinger.pdf

Zanero, S., 2007. 360 degree anomaly based unsupervised ids, Available at: http://www.blackhat.com/presentations/bh-dc-07/Zanero/Paper/bh-dc-07-Zanero-WP.pdf

Zhang, Z., Li, J., Manikopoulos,C., Jorgenson, J., Ucles, J., 2001. HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification, Available at: http://www.itoc.usma.edu/Workshop/2001/Authors/Submitted_Abstracts/paperT2A2(19).pdf