The Korean Institute of
Communications and
Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 4 (2018) 95–99

www.elsevier.com/locate/icte

Intelligent intrusion detection systems using artificial neural networks

Alex Shenfield[a,*], David Day[b], Aladdin Ayesh[b]

[a] *Department of Engineering and Mathematics, Sheffield Hallam University, Sheffield, UK*
[b] *Department of Computing, De Montfort University, Leicester, UK*

Received 6 February 2018; accepted 9 April 2018
Available online 1 May 2018

Abstract

This paper presents a novel approach to detection of malicious network traffic using artificial neural networks suitable for use in deep packet inspection based intrusion detection systems. Experimental results using a range of typical benign network traffic data (images, dynamic link library files, and a selection of other miscellaneous files such as logs, music files, and word processing documents) and malicious shell code files sourced from the online exploit and vulnerability repository exploitdb [1], have shown that the proposed artificial neural network architecture is able to distinguish between benign and malicious network traffic accurately.

The proposed artificial neural network architecture obtains an average accuracy of 98%, an average area under the receiver operator characteristic curve of 0.98, and an average false positive rate of less than 2% in repeated 10-fold cross-validation. This shows that the proposed classification technique is robust, accurate, and precise. The novel approach to malicious network traffic detection proposed in this paper has the potential to significantly enhance the utility of intrusion detection systems applied to both conventional network traffic analysis and network traffic analysis for cyber–physical systems such as smart-grids.

Keywords: Machine learning; Intrusion detection systems; Computer security; Artificial Intelligence

1. Introduction

Network Intrusion Detection Systems (NIDS) are essential in modern computing infrastructure to help monitor and identify undesirable and malicious network traffic (such as unauthorised system access or poorly configured systems). The majority of commercial NIDS are signature based, where a set of rules are used to determine what constitutes undesirable network traffic by monitoring patterns in that traffic. Whilst such systems are highly effective against known threats, signature based detection fails when attack vectors are unknown or known attacks are modified to get around such rules [2].

As well as struggling to identify unknown or modified threats, signature based detection in NIDS in real-world scenarios are frequently plagued by false positives. This is particularly problematic in the detection of malicious shellcode – a high impact threat vector allowing attackers to obtain unauthorised commandline access to both conventional computer systems and cyber–physical systems such as smart grid infrastructure – as shellcode patterns can be difficult to distinguish from benign network traffic [3]. For example, while working as a network security consultant for the Shop Direct Group (UK) using the network intrusion detection tools. Sguil and Snort from the Debian based Linux distribution Security Onion, it was noticed that signatures designed to match shellcode frequently also matched other non shellcode binaries e.g. DLLs as well as jpg image files. The frequency of these false positives was such that the signatures themselves ultimately had to be disabled, rendering them useless. This experience with the false positive problem with shellcode and signature based systems is very common, Microsoft discuss

[*] Corresponding author.
E-mail addresses: a.shenfield@shu.ac.uk (A. Shenfield), david.day@dmu.ac.uk (D. Day), aayesh@dmu.ac.uk (A. Ayesh).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2018.04.003
2405-9595/© 2018 The Korean Institute of Communications and Information Sciences (KICS). Publishing Services by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

this at length in their patent of methods to detect malicious shellcode with reduced false positives in memory [3].

Shellcode is frequently used as a payload in system penetration tools due to the enhanced access and further leverage they offer to an attacker [4].

This paper outlines a non-signature based detection mechanism for malicious shellcode based around Artificial Neural Networks. Results presented show that this novel classification approach is capable of detecting shellcode with extremely high accuracy and minimal numbers of false positives. The proposed approach is validated using repeated 10-fold cross-validation and is then tested with respect to creation of false positive alerts on a large dataset of typical network traffic file contents (achieving a false positive rate of less than 2%).

The rest of this paper is organised as follows: Section 2 provides a background to intrusion detection systems and artificial neural networks, before Section 3 provides a brief introduction to the particular instances that motivated the creation of this system and the results achieved by the proposed AI based intrusion detection system. Section 4 then concludes with the main achievements of this research and some potential avenues for further work.

2. Background and previous work

2.1. Intrusion Detection Systems

The primary aim of an Intrusion Detection System (IDS) is to identify when a malefactor is attempting to compromise the operation of a system. That is to say, cause the system to operate in a manner which it was not designed to do. This could take the form of a compromise to the confidentiality, availability and integrity of the system and the data stored and controlled by it. Systems could be hosts, servers, Internet of Things (IoT) devices, routers or other intermediary devices [5]. Traditionally, at the highest level, intrusion detection systems fall into one of the following two categories, host based intrusion detection systems (HIDS) and network based intrusion detection systems (NIDS). The former being an individual device detecting a compromise and the latter detecting a compromise in transit over a network [6]. NIDS can be further categorised into anomaly and signature based systems. Signature based systems form the mainstay of commercial network intrusion detection systems with anomaly based still largely a research concept [7] with only a few practical vendor backed examples. Increasingly alerts and other incident information generated via an IDS act as a feed into security information and event management (SIEM) systems, along with other logs and feeds allowing a more complete view of a potential incident to be recorded.

2.2. Artificial Neural Networks

Artificial Neural Networks (ANNs) are a form of machine learning algorithm inspired by the behaviour of biological neurons located in the brain and central nervous system [8,9]. Inputs to the ANN are typically fed to the artificial neurons in one or more hidden layers, where they are weighted and processed to decide the output to the next layer. ANNs make use of a “learning rule” (often gradient descent based back-propagation of errors) that allows the set of weights and biases for the hidden layer and output layer neurons to be adaptively tuned. This self-adaptive nature means that ANNs are capable of capturing highly complex and non-linear relationships between both dependent and independent variables without prior knowledge [10].

ANNs have been used in a wide variety of classification tasks across many application domains. In contrast to traditional classification methods, such as logistic regression and discriminant analysis, which require a good understanding of the underlying assumptions of the probability model of the system that produced the data, ANNs are a “black-box” technique capable of adapting to the underlying system model [11]. This makes them particularly useful in fields such as decision support for concealed weapons detection [12], prediction and classification of Internet traffic [13], and signature verification [14] where their ability to adapt to the data, especially in high dimensional datasets, overcomes many of the difficulties in model building associated with conventional classification techniques such as decision trees and k-nearest neighbour algorithms [15].

ANNs have also been used in several computer security domains, including the analysis of software design flaws [16] and computer virus detection [17]. ANN approaches to detection of multiple types of network attacks have also been shown to be effective [18], though their application to the detection of shellcode was not considered.

3. Detecting shellcode in complex network traffic

3.1. Problem domain

Detecting shellcode within complex network traffic poses many challenges for network intrusion detection systems due to the low-level code (usually machine code), small size, and frequently obfuscated nature of the exploits. This is further complicated by the observation that, to signature based detection methods, the binary patterns in shellcode often look indistinguishable from many other forms of benign network traffic.

The work presented in this paper was motivated by the experience of one of the authors working as a network security consultant for a major UK online retailer. Using conventional network intrusion detection tools such as Snort [19] and Sguil [20] to provide event driven analysis of NIDS alerts produce a high level of false positives — with many of these alerts being produced by benign binary and image files. A frequent culprit of false positives was found to be the delivery of files such as DLLs via Windows Update.

3.2. Artificial neural network design

The byte level data from the network traffic dataset used was converted into integer values to feed into the artificial neural network. Care was taken to avoid the “magic numbers” often present at the start of files, as these would be deceptively easy for the classifier to find and are possible to spoof

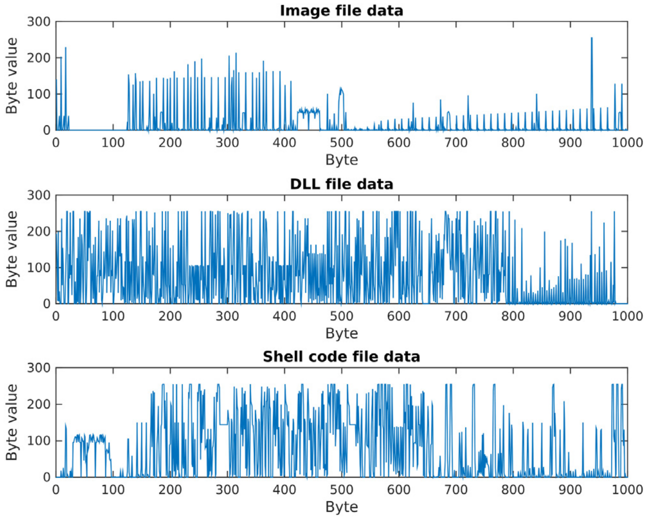

Fig. 1. Byte value data of three different file types: top: images, middle: DLLs, bottom: shellcode.

Table 1
Results of malicious file content detection.

Accuracy	**0.98** (0.01)
Precision	**0.97** (0.01)
Sensitivity	**0.95** (0.04)

(especially when designing obfuscated shellcode). 1000 bytes of contiguous data was extracted and used as an input to the ANN (using zero padding where necessary). Initial exploration and visualisation of the data showed definite patterns within different file types (as shown in Fig. 1), although there was considerable variability between files of the same class.

The ANN for these experiments was implemented using the MATLAB (2016b) Neural Network Toolbox [21]. The optimal structure of the ANN was found through a grid search process, with the best structure (in terms of classification accuracy) for the ANN found to be a multi-layer perceptron (MLP) with two hidden layers of 30 hidden neurons each. The ANN structure optimisation used repeated 10-fold crossvalidation to evaluate the classifier designs. An overview of the final optimised classifier design is shown in Fig. 2. The resilient backpropagation learning strategy (using a default learning rate of 0.01 and training for a maximum of 1000 epochs) was used to train the neural network, with Xavier Glorot initialisation [22] used to set the initial values of the weights.

3.3. Results

The artificial neural network classifier outlined in Section 3.2 above was applied to the network traffic dataset contain- ing both benign and malicious files. Repeated 10-fold crossval- idation was used to ensure that the classifier generalises well to unseen data. Table 1 shows the mean (in bold) and standard deviation of the accuracy, precision, and sensitivity obtained over 1000 iterations of repeated 10-fold crossvalidation.

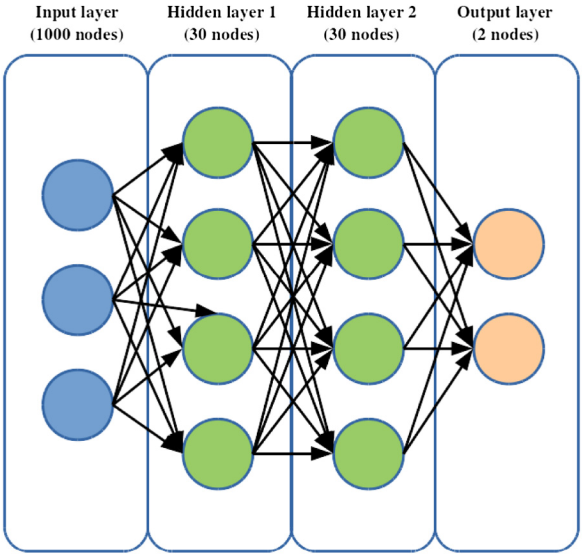

Fig. 2. Final artificial neural network design.

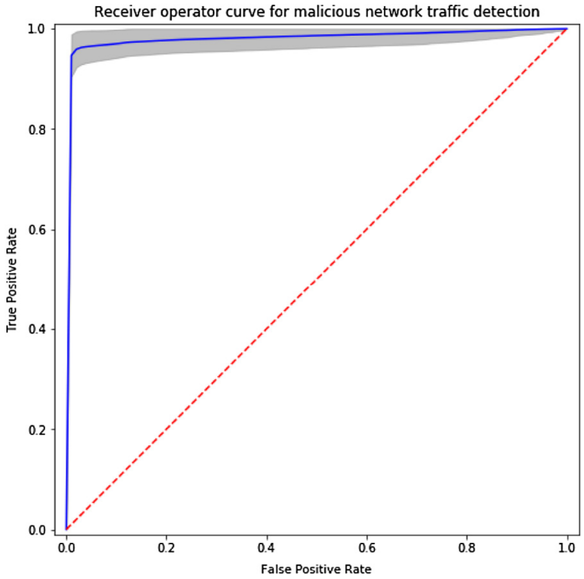

Fig. 3. Receiver-Operator Characteristics Curve for Malicious File Content Detection.

Fig. 3 shows a Receiver-Operator Characteristics (ROC) curve generated using the data for all 1000 iterations of the repeated 10-fold crossvalidation process. ROC curves are commonly used to analyse the trade-off between sensitivity and specificity of classifiers across different classification thresholds. The area under the ROC curve (reported in Table 2) can be used to characterise the overall discrimination of a classification model (with a higher value for the area under the ROC curve indicating that the classifier is better at distinguishing between the two different classes). The bold blue line in Fig. 3 indicates the average ROC curve across all 1000 iterations of the repeated 10-fold crossvalidation, and the grey shaded area indicates the range of ROC curves produced over the course of all 1000 iterations. The dashed red line indicates the performance of

Table 2
Metrics for the area under the ROC curve (AUROC).

Average AUROC	0.98
Standard deviation AUROC	0.02
Maximum AUROC	1.00
Minimum AUROC	0.82

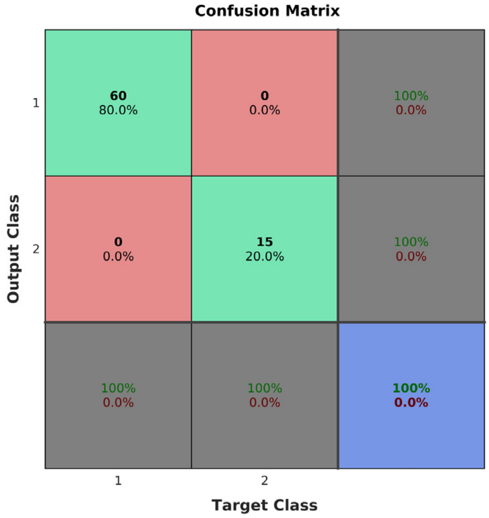

Fig. 4. Confusion Plot for Completely Unseen Test Data.

a classifier which chooses which class a file belongs to at random (this is considered as a baseline for the “worst case” classification performance).

Fig. 4 shows the performance of one of the best performing trained artificial neural network designs on a completely unseen test set (the file contents in this dataset were not used either for training or in the crossvalidation process). As you can see the best performing trained classifier has correctly identified 100% of malicious file contents in the test set, without any false positives!

The performance of the best trained classifier was also tested with regards to flagging up false positives on an extremely large dataset of candidate network traffic data contents. A key driver of this is that, if a network intrusion detection system flags up too many false positives, it becomes useless because any true malicious code is drowned out by benign traffic that has been misidentified. To test this, data from 400,000 random files (consisting of a mixture of text files, log files, compressed and uncompressed music, executables, office documents, and other miscellaneous file data) was extracted into the same format as expected by the artificial neural network and the classifier ran on this benign data. Across this large scale dataset the classifier misidentified 7337 samples (approximately 1.8% of all the data samples).

4. Conclusions and further work

The intelligent intrusion detection system outlined in this paper significantly improves upon the performance of signature based detection methods by utilising an artificial neural network classifier for the identification of shellcode patterns in network traffic. The ANN based classifier not only achieves perfect sensitivity on the test dataset (identifying all instances of shellcode), it also exhibits excellent precision (minimising the number of false positives identified). The performance of the proposed approach was then further evaluated with respect to the false positive rate by testing on an extremely large (400,000 samples) set of benign network traffic file content — where the proposed approach achieved a false positive rate of less than 2%. Minimising the false positive rate is a major concern for the application of network intrusion systems in the real-world, as high levels of false positives result in an extremely poor signal-to-noise ratio and often render the system useless.

The research presented in this paper describes an offline approach to detecting shellcode patterns within data. Work is currently ongoing to integrate the approach proposed in this paper into online network intrusion detection systems and to test on real-time network data, with further real-time optimisations for live network traffic an active area of development. Another area identified for further work is the application of the intelligent approach to intrusion detection outlined here to other areas of network security such as the detection of cross-site scripting attacks and SQL injection attacks on web applications.

Conflict of interest

The authors declare that there is no conflict of interest in this paper.

References

[1] Exploit database. http://www.exploit-db.com/shellcode/, accessed: 2017-11-30.

[2] D. Stiawan, A.H. Abdullah, M.Y. Idris, The trends of intrusion prevention system network, in: 2010 2nd International Conference on Education Technology and Computer, Vol. 4, June 2010, pp. V4–217–V4–221.

[3] J. Shin, J.J. Lambert, J. Lackey, Evaluating shellcode findings, Apr. 2 2013. US Patent 8,413,246.

[4] M. Polychronakis, K.G. Anagnostakis, E.P. Markatos, An empirical study of real-world polymorphic code injection attacks, in: LEET, 2009.

[5] R. Singh, H. Kumar, R.K. Singla, R.R. Ketti, Internet attacks and intrusion detection system: A review of the literature, Online Inform. Rev. 41 (2) (2017) 171–184.

[6] H.-J. Liao, C.-H.R. Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: A comprehensive review, J. Netw. Comput. Appl. 36 (1) (2013) 16–24.

[7] J.M. Vidal, A.L.S. Orozco, L.J.G. Villalba, Quantitative criteria for alert correlation of anomalies-based nids, IEEE Latin Amer. Trans. 13 (10) (2015) 3461–3466.

[8] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (4) (1943) 115–133.

[9] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (6) (1958) 386–408.

[10] J.V. Tu, Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes, J. Clin. Epidemiol. 49 (11) (1996) 1225–1231.

[11] G.P. Zhang, Neural networks for classification: A survey, IEEE Trans. Syst. Man Cybern. C 30 (4) (2000) 451–462.

[12] S. Rostami, D. O’Reilly, A. Shenfield, N. Bowring, A novel preference articulation operator for the evolutionary multi-objective optimisation of classifiers in concealed weapons detection, Inform. Sci. 295 (2015) 494–520.

[13] T. Auld, A.W. Moore, S.F. Gull, Bayesian neural networks for internet traffic classification, IEEE Trans. Neural Netw. 18 (1) (2007) 223–239.

[14] K. Huang, H. Yan, Off-line signature verification based on geometric feature extraction and neural network classification, Pattern Recognit. 30 (1) (1997) 9–17.

[15] S. Dreiseitl, L. Ohno-Machado, Logistic regression and artificial neural network classification models: A methodology review 35 (5–6) (2002) 352–359.

[16] A. Adebiyi, J. Arreymbi, C. Imafidon, A neural network based security tool for analyzing software, in: Doctoral Conference on Computing, Electrical and Industrial Systems, Springer, 2013, pp. 80–87.

[17] G. Liu, F. Hu, W. Chen, A neural network ensemble based method for detecting computer virus, in: 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering, Vol. 1, Aug 2010, pp. 391–393.

[18] J. Wu, D. Peng, Z. Li, L. Zhao, H. Ling, Network intrusion detection based on a general regression neural network optimized by an improved artificial immune algorithm, PLOS ONE 10 (3) (2015) 1–13.

[19] Snort. http://www.snort.org/. Accessed: 2017-11-30.

[20] Sguil: The analyst console for network security monitoring, http://bammv.github.io/sguil/index.html, accessed: 2017-11-30.

[21] Mathworks. Matlab neural network toolbox. https://uk.mathworks.com/products/neural-network.html, 2016.

[22] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y.W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 9, PMLR, 2010, pp. 249–256 Chia Laguna Resort, Sardinia, Italy, 13–15 May.