Smart Computing Review

Smart Detection and Classification of Application-Layer Intrusions in Web Directories

Goran Bujas, Marin Vuković, Valter Vasić*, and Miljenko Mikuc

University of Zagreb, Faculty of Electrical Engineering and Computing / Unska 3, 10000 Zagreb, Croatia / {goran.bujas, marin.vukovic , valter.vasic, miljenko.mikuc}@fer.hr

*Corresponding Author: Valter Vasić

Received October 4, 2015; Revised November 5, 2015; Accepted December 10, 2015; Published December 31, 2015

Abstract: The Republic of Croatia homepage and directory of Croatian web servers (www.hr) attracts several thousand visitors daily, which makes it the target of various attacks. In order to lower the risk from such attacks, we propose a concept for an intrusion detection system and a classifier of detected intrusions. We first examined the concepts of existing intrusion detection systems and combined their individual benefits into a concept best suited for protecting web services on the application layer. The proposed concept uses machine learning techniques for both intrusion detection and classification. Intrusion detection, observed through analysis of requests, is implemented by a feed-forward neural network, while intrusion classification is done using self-organizing maps. The case study and preliminary evaluation is presented on the Republic of Croatia homepage (www.hr), followed by guidelines for further research.

Keywords: Intrusion Detection, Dntrusion Classification, Machine Learning, www.hr

Introduction

The Internet has become an integral part of society, and the services it provides are expanding into every aspect of our lives. These services span from social interactions to transportation, and are also used to plan and enjoy leisure time. With the expansion of the Internet of Things, even the most personal of data are shared over the Internet to provide a wide

DOI: 10.6029/smartcr.2015.06.002

range of services. With the increasing number of services and users, the economy is also shifting towards a networked environment. This increases the interest some people have in exploiting and attacking Internet services for their own interest and benefit. Attackers usually try to gain control of vulnerable systems to obtain access to confidential data or to conduct further attacks on other targets. Recently, attacks have been more focused on reducing the availability of Internet services through denial of service attacks. Attacks can be conducted by a wide range of attackers, from "script kiddies" to automated systems targeting a wide range of devices with the aim of forming large controlled networks, such as botnets. To efficiently counter most attackers, a sophisticated and automated system should be deployed in different segments of the network. Such an automated system is usually referred to as an intrusion detection system (IDS).

An IDS can be classified into one of three major groups: a network-based IDS (NIDS), a host-based IDS (HIDS), or a hybrid IDS. An NIDS is a traditional IDS implemented mostly as a hardware device. It can work in passive or active mode. In passive mode, a mirrored traffic stream is delivered to the device to be classified as legitimate or malicious. In active mode, sometimes referred to as inline mode, the device is placed in the traffic stream. In this mode, an IDS can detect anomalous traffic, or it can act after detection and prevent potential attacks. In the latter case, the system is called an intrusion prevention system (IPS). In network implementations, prevention is usually achieved by blocking the IP address of the source of the attack for a certain period of time. This is achieved either by silently discarding the rest of the traffic sent by that particular IP address, or by actively sending TCP_RST packets to the sender. The latter obviously works only for TCP traffic. In passive mode, IPS systems can also mitigate attacks indirectly, through reconfiguration of active network equipment, e.g. firewalls. A HIDS is implemented on a single host and protects it from attacks. The main disadvantage is that a HIDS is not scalable to a larger environment [1]. Finally, a hybrid system is a combination of NIDS and HIDS that aims to combine the advantages of both approaches.

This article proposes a novel IDS concept to be used on the Republic of Croatia homepage (www.hr). This site is visited by several thousand users on a daily basis and serves as a directory of Croatian web pages which are categorized and automatically rated based on the number of visits. As such, it is constantly under attack by entities that are trying to gain administrator access. This would allow attackers to distribute various malicious applications to the daily users of www.hr. In addition, the directory database consists of around 25,000 web sites with direct contacts, and they could be targeted by malicious actions, such as spam. The Republic of Croatia homepage project places a high priority on protecting its visitors and customers that are registered in the directory. For this purpose, a signature-based IDS system is used to protect the www.hr service from attacks. As the amount and complexity of attacks increase, the deployed system needs to be updated and refined to successfully counter them. Therefore, an analysis of state-of-the-art IDS concepts is provided in this article, followed by the proposed application-layer IDS concept appropriate for the Republic of Croatia homepage, as well as other web servers. Furthermore, the paper focuses on a broader view of the problem, and provides insight into the key aspects needed to protect currently deployed Internet services.

In Section 2 we give a detailed overview of currently available IDS concepts, and describe the latest research in the area. Section 3 describes our novel IDS concept for application layer protocols. Section 4 demonstrates integration of the proposed concept on the Republic of Croatia homepage, and we conclude the paper in Section 5.

Related work

When it comes to detection, IDS systems can be generally classified into three major groups: signature-based, traffic validation, and anomaly-based.

A signature-based [1] or pattern-matching IDS uses a predefined set of rules to match specific patterns of traffic. It is the most common IDS solution on the market. Typically, keywords or regular expressions are used for detection. The main disadvantage of a signature-based IDS is that it cannot detect attacks that are not in the signature database. When new attacks are discovered, the signature database is updated with the new signatures. A signature-based IDS usually works on the network and transport layers, but it can also work on upper network layers with adequate traffic preprocessing and normalization.

Traffic validation or a protocol-behavior IDS [1], usually complements a signature-based IDS. Traffic validation IDS is built for one or more known network protocols. The traffic stream for a particular protocol is compared to the Request for Comments (RFC) specification of that protocol. Any deviation from normal behavior is considered anomalous, and as a result, an alert is generated. However, attacks that do not violate a protocol specification can be missed, e.g. attacks using the HTTP protocol that manipulate URLs cannot be detected because they heavily depend on a particular web application implementation, and not a protocol specification. The traffic validation IDS usually works on a layer where a specific protocol resides (e.g. TCP on the transport layer, HTTP on the application layer). A similar approach is employed in next generation firewalls, where firewall rules are constructed on a per protocol basis and are not TCP or UDP port–dependent. However, as technology advances and new protocols are adopted (specifically on the application layer), modeling protocol behavior becomes difficult and time-consuming.

The third type of intrusion detection system is an anomaly-based IDS. The anomaly-based IDS [1] takes a different approach and mainly focuses on unknown attacks. An anomaly-based IDS has two main techniques for detection: statistical and machine learning. Both require a certain period for learning. In statistical models, the learning period usually produces acceptable statistical levels for certain traffic parameters or baselines. Any deviation above the defined baseline produces an anomaly alert. The disadvantage with statistical anomaly detection is a high false positive rate. Any deviations from the baseline, i.e. traffic bursts or other unexpected but valid traffic patterns are labeled misbehavior. Machine learning techniques have similar periods of learning, but detection is based on machine learning models. The main advantage of machine learning techniques lies in a more precise network behavior model, compared to statistical techniques. Both the statistical and machine learning anomaly-based IDS have to be retrained from time to time as network conditions change. Commonly, an anomaly-based IDS has poorer performance characteristics than other IDS types, so most of the research is focused on improving performance of such systems. The main research objectives are:

- minimize false positive and false negative rates
- increase true positive and true negative rates
- optimize computational performance to decrease network latency and system complexity
- minimize training and retraining periods

Comprehensive research was done by Shah et al. [2], where they referred to studies researching machine learning techniques from 2000 to 2012. They compared different approaches to the machine learning IDS over the course of 12 years. The study identified three main approaches in classifying traffic: single, hybrid and ensemble. Most researched single classifiers are Support Vector Machine (SVM), K-Nearest Neighbor (k-NN), Decision Tree (DT) and Genetic Algorithm (GA). A lot of hybrid systems have been researched. Hybrid systems are systems where two or three machine learning methods have been used as classifiers to enhance the accuracy and precision of the IDS. Hybrid systems introduce additional machine learning methods, whereas ensemble methods focus on combining the most efficient single classifiers. In conclusion, the authors state that the main methods of research are increasingly done with an SVM method used as a single classifier. However, since none of the single classifiers can identify all types of attack, and there are many false alarms, so further research should put more emphasis on hybrid classifiers.

Another comparative study by Juma et al. [3] focused on the IDS with a hybrid classifier over the period from 1993 to 2014. In conclusion, the authors stated that using a hybrid classifier in an IDS with an ineffective combinational can actually impact detection performance. They concluded that there is still no efficient intrusion detection system. Each hybrid classifier method has more promising results than the others, but only under specific conditions. There is no single method that suites all situations.

A study by Choudhury and Bhowal [4] empirically compared several machine learning techniques. The data set used in this research was NSL-KDD. The study compared nine classification techniques and measured their performance. The metrics used in the study were sensitivity (true positive rate), specificity (true negative rate), precision (probability of a positive prediction being correct), accuracy (number of correct predictions), training time, and mean absolute error. The classifiers used in the research were BayesNet, Logistic, IBK, J48, PART, JRip, Random Tree, Random Forest and REPTree. The authors' conclusion states that the Random Forest and BayesNet classifiers are the most suitable.

A study by Kumari and Kumari [5] determined that machine learning techniques used in an IDS produce static intrusion detection systems that poorly adapt to new network behavior and are vulnerable to new attacks. A static IDS is periodically trained because it needs to be updated regularly in offline mode. The authors proposed Adaptive Anomaly Intrusion Detection, based on a stream mining ensemble classifier. They used an adaptive size Hoeffding tree, an online boosting algorithm, and an adaptive sliding window algorithm, ADWIN. The empirical results of their model was compared to unsupervised machine learning techniques (K-means, self-organizing maps, and a farthest first algorithm) on an NSL-KDD data set showed an accuracy increase and a false positive rate decrease.

Haque and Alkharobi [6] showed that reducing the dimensionality of a large dataset can provide an accurate and lightweight intrusion detection system that can be embedded into vulnerable systems with improvement in execution time. They divided machine learning techniques into three categories:

- supervised anomaly detection—both normal and abnormal traffic is used to train and make a predictive model
- semi-supervised anomaly detection—only one type of traffic is used (normal or abnormal)
- unsupervised anomaly detection—this does not require training data, and assumes that the frequency of normal traffic is much greater than the frequency of abnormal traffic

A study by Choudhary and Dalal [7] proposed a feature-based intrusion data classification technique in which reduction of feature attributes (service or dst_host_count parameters) improves the classification of intrusion data, which in turn, decreases the execution time of classification and increases IDS performance. In their research, attacks are broadly divided into four groups: denial of service (DoS) attacks, remote-to-local (R2L) attacks, user-to-root (U2R) attacks, and probing attacks.

The research of Gupta and Shrivastava [8] focused on an SVM machine learning technique and clustering based on bee colonies. The SVM learning technique gives good results with a small data volume. A system that is built on SVM has to

be updated as soon as new attack information is available. The training of the system with new data can take weeks, or longer. With clustering based on bee colonies, an adaptive model can be generated based on old models and new information.

Gaikwad and Thool [9] proposed an ensemble method of machine learning that is a combination of Bagging and REPTree. They used metrics such as classification accuracy, model building time, and the number of false positives. The experimental results on an NSL-KDD data set showed that this combination gives the highest classification accuracy and takes less time to build a model than an AdaBoost algorithm with Decision Stump base classifiers.

The research by Hussain et al. [10] suggested that a single classification technique fails to provide the best attack detection rate. They proposed a two-stage hybrid classification model using SVM for anomaly detection in the first stage and an artificial neural network (ANN) for misuse detection in the second stage. The main objective was to combine the best of both algorithms for better classification accuracy and a lower false positive rate. Anomaly detection (SVM) was used to classify data into normal and attack traffic, whereas misuse detection (ANN) classified attack data into four classes: DoS, R2L, U2R, and probe. Empirical results on an NSL-KDD data set showed that the hybrid system has better performance and reduces computational complexity better than a conventional IDS or single classification with an SVM or ANN method.

A study by Zuech et al. [11] proposed an approach in monitoring security events from many different network sources. Their presumption was that correlation of security events from heterogeneous sources can give a more complete view of cyber threats and cyber intelligence. This research focused on big heterogeneous data, data fusion, heterogeneous intrusion detection architectures and security information and event management (SIEM) systems. They concluded that academia should expand its research into more diverse event sources, not just NIDS or HIDS. They put weight on the industry and advancement of SIEM technology where the next field of research should focus.

Research by Moghaddam and Calix [12] focused on machine learning techniques implemented in hardware. A network IDS has to perform efficiently to minimize network latency. They implemented KNN and Restricted Coulomb Energy (RCE) machine learning methods in the hardware. The research consisted of preprocessed and normalized network data, which was delivered to a single cognitive processor, CM1K. The primary disadvantage of hardware implementations is a limited number of neurons due to hardware constraints. Therefore, for more complex implementations, more processors need to be used. They concluded that hardware implementation holds promising results, and that the research should continue, because having a single chip IDS with a complex classification model would present a significant advantage. The authors are planning future work on comparison of hardware and software implementations in terms of classification accuracy and performance.

Proposed application-layer IDS and classifier system

After reviewing current best practices from related work, we propose an application layer intrusion detection and classifier system. The main components of the proposed system are a pattern generator, a request analyzer and an intrusion classifier. Pattern generation is an important step for presenting requests as inputs to the request analyzer. It is implemented by a knowledge-based algorithm that derives specific parameters from the request. The request analyzer is implemented by a feed-forward neural network that uses generated patterns for training and detection of malicious requests. Its output is the estimated threat level, ranging from zero to one. The intrusion classifier is a self-organizing map (SOM) that uses the same input patterns as the request analyzer, whereas the output corresponds to the threat type. Classifying the threat type is important because it allows the system to take action, depending on the severity of the incoming threat—from blocking a single IP address, to raising firewall rules to a higher security level. The proposed request analysis and intrusion classification is presented in Figure 1.

When a new request is received, it is initially checked against flagged request records. Flagged request records enable time-dependent analysis and detection by marking possible threats within previous requests sent from a certain IP address or address range. If the originating IP address or range was not flagged as suspicious, the request is considered harmless and processed as such. However, because every new request can be an attack from a previously unknown source, it still needs to be analyzed, which is done in parallel to request processing (the upper part of Figure 1). Parallel processing and analysis is done in order to not slow down the server response, which would be the case if each request was analyzed prior to processing. On the other hand, if the initial check shows that the request might be malicious, it is sent directly for analysis, prior to any processing. This results in slowed down processing of the request, but enables rejection of the request if it turns out to be malicious.

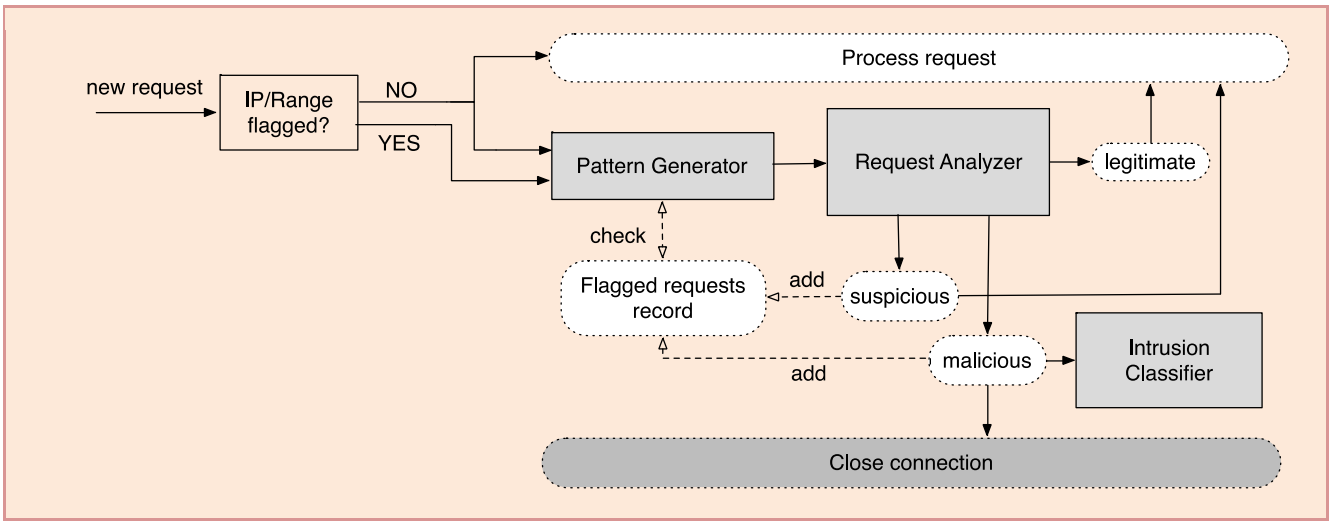

Figure 1. Proposed request analysis and intrusion classification

Request analysis is performed as follows. Because analysis is done by a neural network, the request needs to be transformed into patterns. The pattern generator is based on predefined rules that determine what components of the request should be taken into consideration when generating patterns. Besides the current request data, the patterns should contain information about previous potentially malicious requests from the same or a similar origin (e.g. an IP address or range). Furthermore, flagged requests allow observing more than one request, rather than a single request, which enables detection of attacks that combine several requests in a time series. It is reasonable to assume that some of the attacks, especially scanning and denial of service attacks, will be executed with more than one request.

Generated patterns are sent to the request analyzer's neural network input. If the analysis determines that the request is suspicious, its data is added to the flagged request records, but the request is still sent for processing, if not already in the process (depending on the initial check). However, if the request is determined to be malicious, the connection needs to be closed, and the request data should be added to the flagged request records for future use.

One issue with the proposed approach is that if the malicious request in question passed the initial IP address and range check, it might have already been processed as harmless. This is the tradeoff between security and availability of the server. In other words, the first malicious attempt will be processed as harmless, but further requests from the same or a similar origin will be rejected. Finally, a malicious request is sent to the intrusion classifier, which tries to determine the type of intrusion and performs certain actions to protect the server according to the intrusion type.

Subsections below give more details about each key component of the proposed application-layer IDS.

Pattern Generator

The pattern generator needs to transform raw requests into a format suitable for neural networks. This task is crucial because the patterns should represent real requests as much as possible. But it must also remove data not necessary for the given purpose without losing any information that might distinguish harmless requests from malicious ones. The patterns consist of three groups of data: existing flagged requests, protocol headers and methods, and resource/destination type identification, which are illustrated in Figure 2.

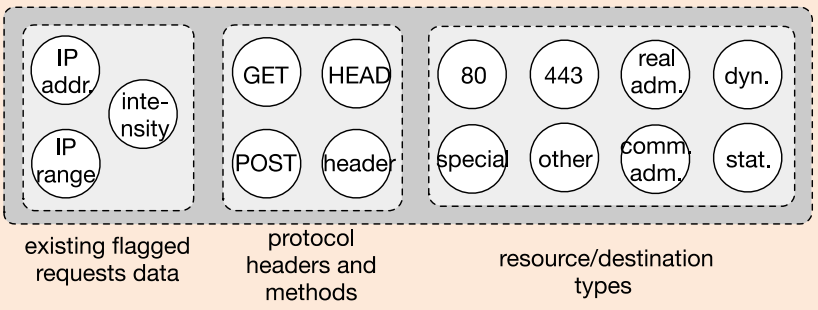

Figure 2. Input pattern representing request data

Existing flagged request data need to indicate whether the used IP address or range should be considered suspicious, and need to include information about the intensity of requests from the source in question. This information is represented by activation of three neurons: one representing a suspicious IP address (0 or 1), one representing a suspicious IP range (0 or 1), and another indicating the intensity by adjusting its activation with a value between 0 and 1.

The protocol headers and methods section deals with methods and headers from the protocol HTTP or HTTPS. Once again, one neuron indicates the POST method, the second the GET method, and the third the HEAD method, all by neuron activation 0 (not present) or 1 (present). The *headers* neuron indicates whether the headers are malformed, as is common with some types of attack.

Finally, a third section of neurons represents resource and destination type. For this purpose, we defined several classes of resource and destination type. Destinations are observed through ports, and we distinguish four classes: 80, 443, Special and Other. Ports 80 and 443 are common web application ports, whereas the Special class represents known ports commonly used by server administration tools, such as Webmin (10000), WebSphere (9060), etc. The Other class indicates any port other than 80, 443, or the ports from the Special class list. Resources are also divided into classes as follows: real administration, common administration, dynamic, and static. Real administration indicates that the request is pointed towards one of the existing server administration interfaces. Common administration is activated when the request is attempted at one of the well-known administration interfaces from existing commercial and open-source content management systems, administration tools, and the like. These include tools like phpMyAdmin, the Wordpress administration interface, and similar tools, and they are often a target of automated malicious scripts and tools. Lastly, the dynamic or static classes indicate whether the request is aimed at static resources (e.g. images) or dynamic resources (e.g. scripts), the latter being a more common target of malicious requests.

Request Analyzer

The request analyzer neural network is a feed-forward architecture trained with a back-propagation algorithm, presenting a supervised training method, as mentioned by Haque and Alkharobi [6]. Prior to request analysis, the network has to be trained with representative gathered requests, which are combined into the training set. The training set needs to consist of both legitimate and malicious requests with assigned threat levels. Requests should be selected from raw server access logs which need to be carefully examined and assigned a realistic threat level. This type of network can be trained with a back-propagation algorithm. After successful training, indicated by a near-zero mean square error, the network can be used to analyze incoming requests. The network used for request analysis is presented in Figure 3.

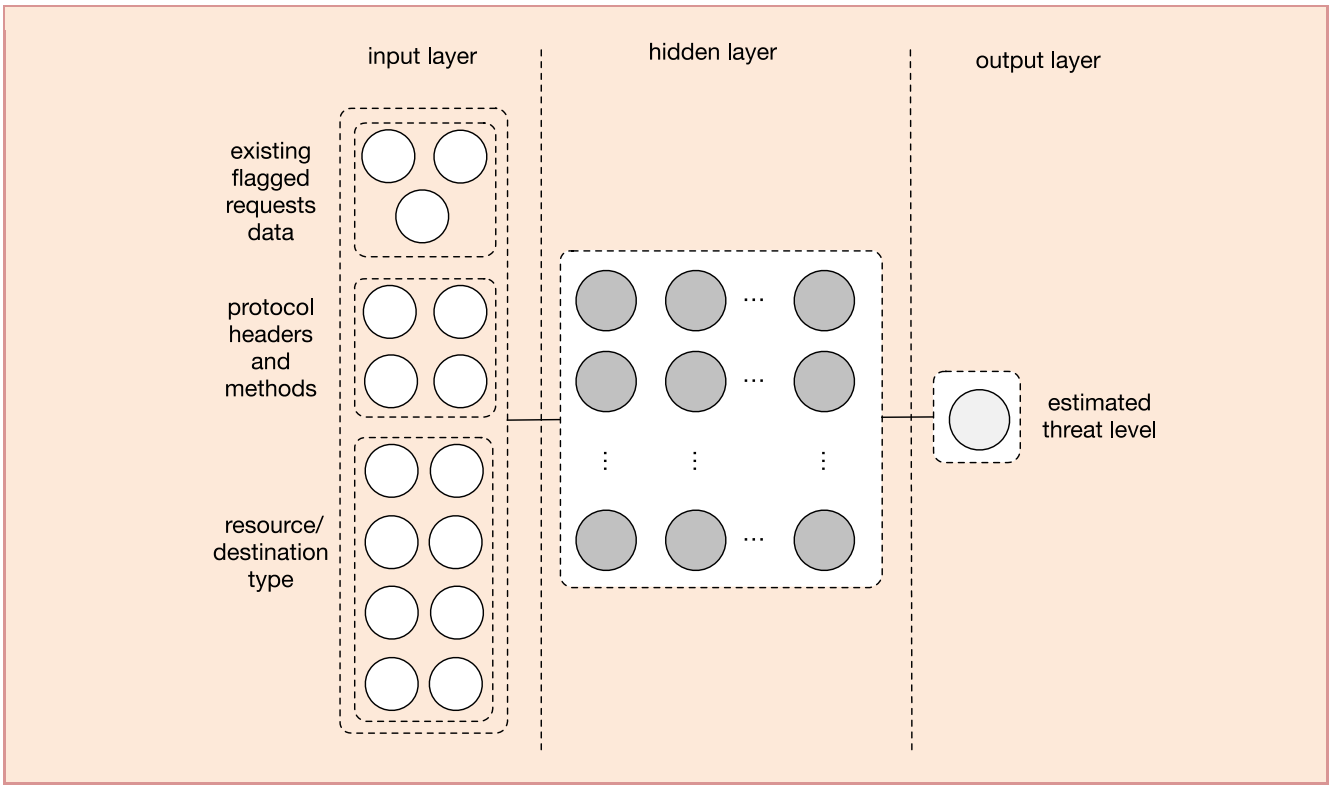

Figure 3. Feed-forward network for request analysis

The input layer consists of 15 neurons, according to the described input patterns, whereas the output layer has one neuron that indicates the threat level, ranging from 0 to 1. The size of the hidden layer is highly dependent on the correlation between input patterns, and should be determined experimentally.

The proposed neural network architecture is suitable for the given purpose, because the threat level of new, unknown, requests will be estimated according to their similarity to data from the training set. Although we tried to encompass the usual attack types and access to sensitive web application resources through pattern generation, a probable drawback to this approach is that completely new and non-typical attacks may not be detected. Evaluation and eventual adaptation to new types of attacks in real time is the focus of ongoing and future work.

Intrusion Classifier

The final key component is the intrusion classifier. It is used after the request has been detected as malicious in order to classify the type of intrusion. The classifier is implemented via SOM, an unsupervised neural network architecture [6]. SOM networks are classifiers that group input patterns according to their similarity on a two-dimensional (or more) output layer. Similar input patterns will be mapped more closely on an output layer forming clusters, while new and different patterns will activate the neurons farthest from the neurons activated by known patterns. In this sense, distance is observed as a Euclidean distance in a two-dimensional output array. The SOM architecture is presented in Figure 4.

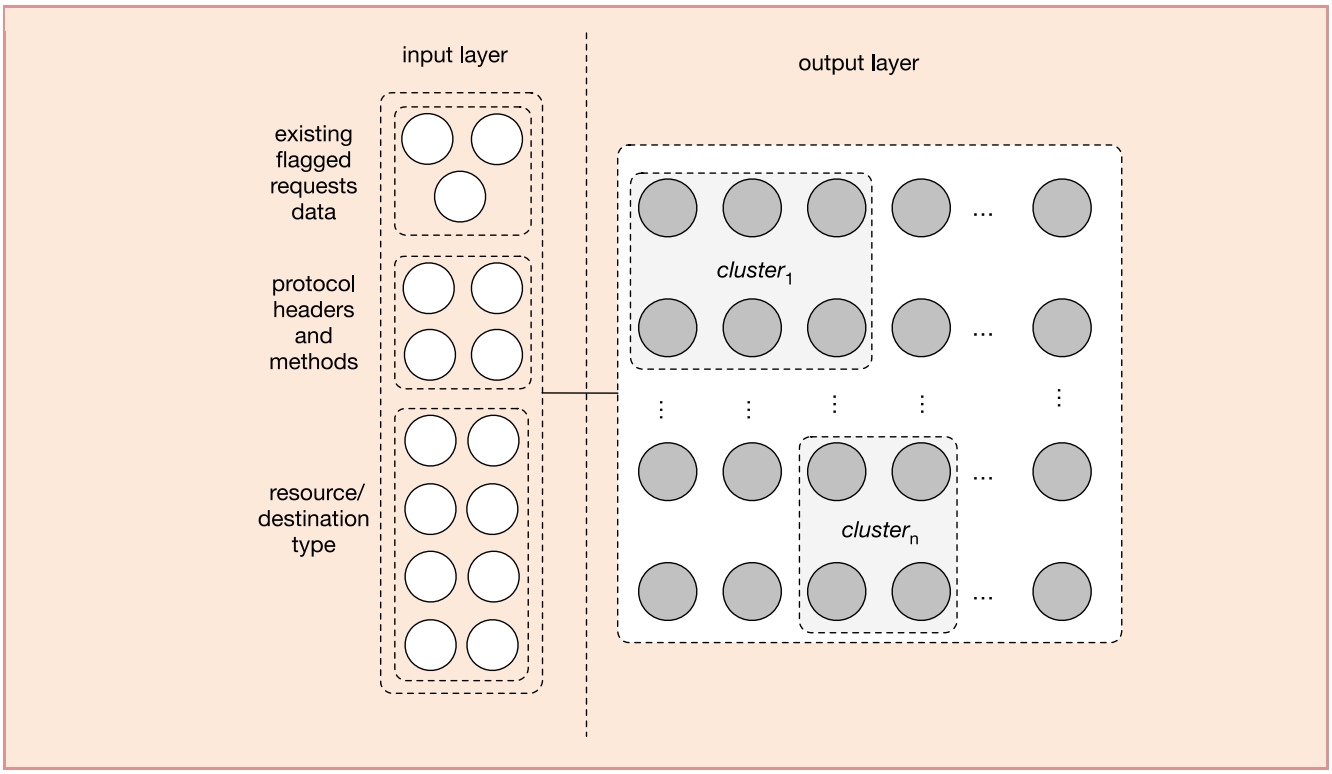

Figure 4. SOM architecture used for intrusion classification

In the proposed system, we focus on a winner-takes-all training method, where only a single neuron is activated for each input pattern. The SOM is first trained with the subset of malicious patterns from the patterns used for training the request analyzer, but without output values, because SOM uses unsupervised learning. After the network maps inputs to the output layer, clusters and individual neurons that represent single types of attack can be determined by probing the network with input patterns. The level of precision for this purpose can be fine-tuned by adjusting the size of the output layer. Because we know what type of attack is represented by each input pattern used during the training, we assign each cluster or neuron a single attack type based on its activation during probing. During usage, activation of a neuron within a certain premarked cluster indicates the type of attack as marked during probing.

The purpose of determining the attack type is to take certain action according to the attack in progress. Actions may vary, from blocking a certain IP address for limited time (e.g. for a probe of the server for administrative tools) to blocking whole ranges of IP addresses, and even slowing response times in case of distributed attacks.

Application of the proposed concepts on the Croatian homepage

The Republic of Croatia homepage (www.hr) offers two basic services: the homepage of the Republic of Croatia with basic information about Croatia, and an official directory of Croatian webservers. It is a project funded by the Croatian Academic and Research Network (CARNet) developed at the Department of Telecommunications in the Faculty of Electrical Engineering and Computing (FER) at the University of Zagreb.

The architecture of www.hr is shown in Figure 5. Basic information about Croatia is stored in the content database and can be managed through the content management system admin interface. The official directory of web pages is stored in a database that can be administered through the web page directory admin interface. For regular users, both content and web page databases are accessed through a unified www.hr service. This service is, by far, the most accessed of the interfaces. The architecture also includes four additional services that are needed to give a richer user experience and to protect users from malicious content. User experience modules generate page thumbnails for easier searching and looking up real-world locations of pages in the directory. A search indexing service speeds up search times. Malicious content detection is periodically done for all categorized web pages in order to prevent users from accessing potentially dangerous sites.

The architecture also shows where the proposed IDS system is placed in the architecture. It partially filters incoming requests to prevent slowing down server responses. The placement of the IDS system in the architecture demonstrates the flexibility and adaptability of the proposed solution to a large number of systems with both simpler and more complex architectures.

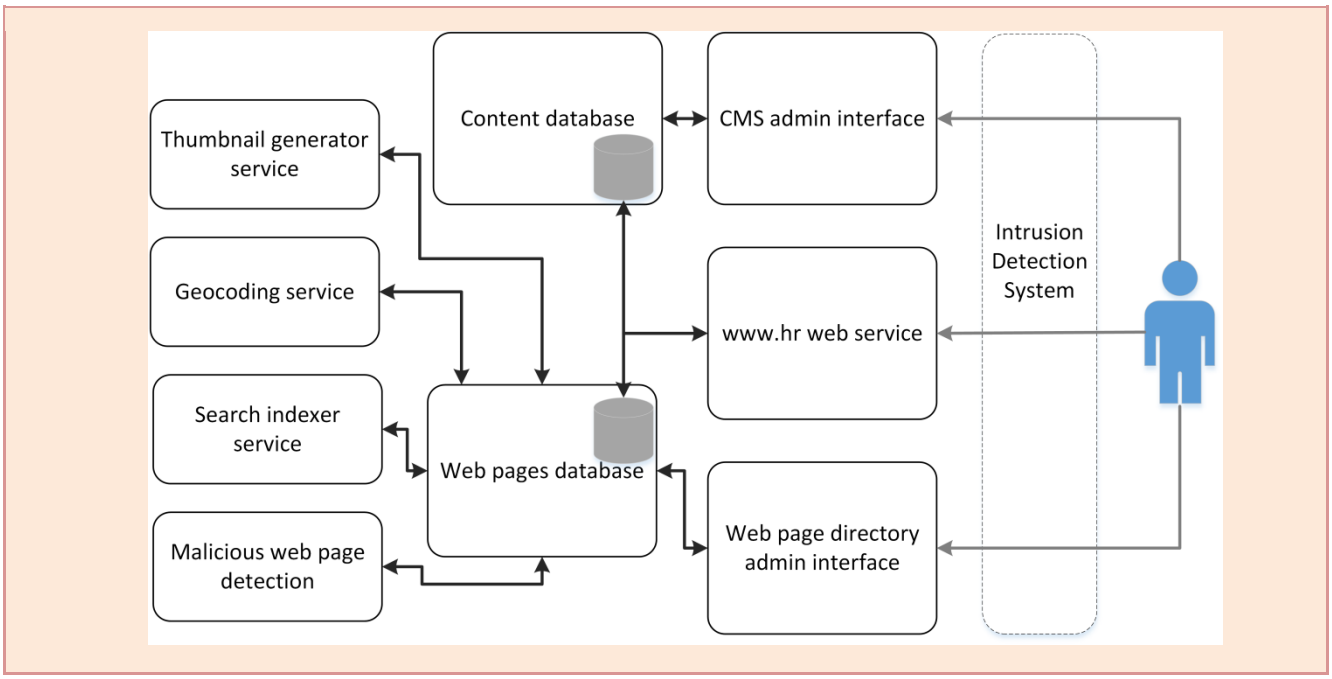

Figure 5. Architecture of the www.hr service

Prior to deployment of the proposed IDS, we analyzed one week of raw www.hr access logs in order to deduce the IP addresses and IP ranges we should filter from the start. Furthermore, we manually observed requests and selected typical representations of malicious requests, along with legitimate ones, in order to form a training set of patterns. Threat levels were evaluated by administrators of www.hr.

When first deployed, the proposed IDS contained 3208 IP ranges and 3670 IP addresses that were initially blocked, based on the examined one-week period. These addresses and ranges were automatically blocked for two days and were added to initial flagged request records.

Regarding protocol headers and methods, the GET method is the most common on www.hr, whereas POST is used for search, new site registration, and modification of existing sites in the directory. Malicious GET requests typically focused on mirrored Cross-site scripting (XSS) attacks, whereas POST method requests tried injecting custom JavaScript and SQL. Malicious HEAD requests were used by custom scripts that probed the server, similar to port scans on lower layers.

Most of the examined traffic was targeted at port 80, while most of the malicious traffic tried to access both ports 80 and 443 in order to access administration interfaces. Interestingly, very few requests were aimed at special ports, but a high

number of requests, probably originating from scripts, tried to access common interface URLs. As expected, no malicious traffic was addressed at the real administration interface because the link is not publicly available. The relation of dynamic and static resources accessed did not show any unexpected variations, although malicious attempts were almost always aimed at dynamic resources (e.g. php files).

Upon examining raw access logs and training the SOM we could distinguish four main groups of access types: administration probing, injection, XSS, and common vulnerability checks. However, this should be examined in more detail, and will be the focus of future work.

After deployment, the proposed concept managed to filter out requests that were most similar to the requests used in the training process. Nevertheless, every first malicious request can still access the server and will not be filtered. Therefore, it can be concluded that the proposed system is suitable for attacks that use more requests (which is the most common case) but individual request probing is a problem for the proposed solution. We estimate that the proposed solution can filter up to 80% of known malicious packets. Such a rate is the result of not checking the first request and, to a higher extent, due to the fact that the proposed request analyzer must be periodically trained with data on new malicious requests. All these issues are a focus of ongoing and future work.

Conclusion

In this paper, we analyzed existing state-of-the-art intrusion detection systems and evaluated how the existing concepts could be engaged for protecting the Republic of Croatia homepage (www.hr). We propose an IDS concept based on a feed-forward neural network, with emphasis on a pattern generation process, a critical prerequisite for most machine learning solutions. Alongside detection, we try to classify intrusions using self-organizing maps in order to take action according to intrusion type. We estimate that the proposed concept is able to filter up to 80% of malicious requests and successfully classify the intrusions based on their similarity to the malicious requests from the training set. The drawback of the proposed concept is that every first malicious request will be processed as harmless, a tradeoff implemented in order to not slow down the server significantly. Another challenge is the required adaptability of the concept, which would allow it to respond to new types of malicious requests. In future work we plan to address the existing drawbacks and challenges, especially the issue of automatic adaptability. The concept is currently deployed in a testbed environment and is continuously being analyzed for performance and upgraded.

References

[1] S. Harris, “All in one CISSP Exam Guide,” *McGraw-Hill*, 2015.

[2] A. A. Shah, M. S. H. Kiyhal, M.D. Awan, “Analysis of Machine Learning Techniques for Intrusion Detection System: A Review,” *International Journal of Computer Applications*, vol. 119, 2015. Article (CrossRef Link)

[3] S. Juma, Z. Muda, M. A. Mohamed, W. Yassin, “Machine learning techniques for intrusion detection system: a review,” *Journal of Theoretical and Applied Information Technology*, vol. 72, no. 3, pp. 422-429, 2015. Article (CrossRef Link)

[4] S. Choudhury, A. Bhowal, “Comparative Analysis of Machine Learning Algorithms along with Classifiers for Network Intrusion Detection,”*2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials*, IEEE, pp. 89-95, 2015. Article (CrossRef Link)

[5] S. R. Kumari, P. K. Kumari, “Adaptive Anomaly Intrusion Detection System Using Optimized Hoeffding Tree and Online Adaboost Algorithm,” *World Applied Sciences Journal*, vol. 33, no. 1, pp. 102-108, 2015. Article (CrossRef Link)

[6] M. E. Haque, T. M. Alkharobi, “Adaptive Hybrid Model for Network Intrusion Detection and Comparison among Machine Learning Algorithms,” *International Journal of Machine Learning and Computing*, vol. 5, no. 1, Feb. 2015. Article (CrossRef Link)

[7] S. Choudhary, P. Dalal, “An Architecture for Network Intrusion Detection System Based on Dag Classification using Hybrid Enesemble and Ensemble Method,” *International Journal for Innovative Research in Science & Technology*, vol. 1, no.11, Apr. 2015. Article (CrossRef Link)

[8] M. Gupta, S. K. Shrivastava, “Intrusion Detection System based on SVM and Bee Colony,” *International Journal of Computer Applications*, vol. 111, no. 10, Feb. 2015. Article (CrossRef Link)

[9] D. P. Gaikwad, R. C. Thool, “Intrusion Detection System Using Bagging Ensemble Method of Machine Learning,” *International Conference on Computing Communication Control and Automation*, pp. 291-295, 2015. Article (CrossRef Link)

[10] J. Hussain, S. Lalmuanawma, L. Chhakchhuak, "A Novel Network Intrusion Detection System using Two-stage Hybrid Classification Technique," *International Journal of Computer & Communication Engineering Research*, vol. 3, no. 2, pp. 16-27, Mar. 2015. Article (CrossRef Link)

[11] R. Zuech, T. M. Khoshgoftaar, R. Wald, "Intrusion detection and Big Heterogeneous Data: a Survey," *Journal of Big Data*, vol. 2, no. 1, pp 1-41, 2015. Article (CrossRef Link)

[12] M. H. Moghaddam, R. A. Calix, "Network Intrusion Detection Using a Hardware-Based Restricted Coulomb Energy Algorithm on a Cognitive Processor," *28th International Florida Artificial Intelligence Research Society Conference*, July 2015. Article (CrossRef Link)

Goran Bujas received MS degree in Electronic Engineering from the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia in 2006. In 2015 he started pursuing his Ph.D. degree in Electrical engineering on the same University. He has six years extensive technical experience in implementation of security solutions in computer networks, vulnerability assessment, incident handling, incident investigations and information security governance.

Marin Vuković is Associate Professor at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia (UNIZG-FER). He has been affiliated with the Department of Telecommunications at UNIZG-FER since 2006. He received his diploma degree (Dipl.-Ing.) in electrical engineering with a major in telecommunications and informatics from the UNIZG-FER in April 2006. He successfully finished PhD studies in June 2011. Marin Vuković has co-authored over 30 journal and conference papers and reviewed a number of papers for international conferences. He is a co-author of the patent at the Croatian Institute for Intellectual Property P20080303A. He is a member of IEEE Communications Society, Royal Institute of Navigation and KES International.

Valter Vasić received his M.Sc. in Computer Science in 2010 from the Faculty of Electrical Engineering and Computing, University of Zagreb. In 2010 he started pursuing his Ph.D. degree in Computer science on the same University. He is currently employed as a research associate at the same faculty within the E-IMUNES project funded by Ericsson Nikola Tesla. His research interests include security, network simulation and virtualization. He is an author of 5 conference papers and 1 journal paper. He is a member of IEEE.

Miljenko Mikuc received his PhD in Electrical Engineering from University of Zagreb, Croatia, in 1997. He is currently Associate Professor at the Faculty of Electrical Engineering and Computing, Department of Telecommunications within the same university. His area of interest includes network protocols, network simulation and security. He is an author of more than 20 conference papers and 4 journal papers.

Copyrights © 2015 KAIS