International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 12, pp: 70-81 www.ijaem.net ISSN: 2395-5252

A Comprehensive Review of Intrusion Detection System Approach and Technique

Ohuabunwa Augustine Ebere[1*], Icheke Led[2], Ukaegbu ThankGod Ekene[2], Dilibe Godson Chijioke[2], Kadiri Ramotu Ochuwa[1]

[1]Information Communication Technology Unit, Electronics Development Institute, Awka, Nigeria

[2]Research and Development Unit, Electronics Development Institute, Awka, Nigeria

Corresponding Author: Ohuabunwa Augustine Ebere

Date of Submission: 14-11-2021 Date of Acceptance: 29-11-2021

ABSTRACT: The problem of securing information and data flows has existed from the origin of network communication. Diverse approaches have been developed to protect and transfer such information safely. Conversely, as communication technology advance and information management systems become more powerful and distributed, attack on network system and web application has taken on new and more complex dimensions posing a great challenge to both government and business organization. Security breaches almost occur on daily basis; hence, the need for adequate information security. Presently, security is implemented using firewalls, encryption, intrusion detection systems (IDS), authentication and other software and hardware solutions. This paper provides an up-to-date comprehensive overview of IDS approach and technique through a systematic survey of the major work done in this area to provide researchers with adequate information needed to design robust and efficient IDS.

KEYWORDS: Intrusion, Detection, Security, Model, Machine learning, Anomaly, attack Vulnerability, Pattern

I. INTRODUCTION

Cyberspace plays a key role in modern societies and economies. Recently, there has been a tremendous rise in the use of web-based applications [1]. Apart from personal users, companies and governments have become highly dependent on the cyberspace for their daily activities. Applications such as social networking sites, e-banking, online blogs and e-commerce have become a general platform for conveying information and providing online services. Critical tasks such as communication activities and the control of critical infrastructures now depend on the Internet. According to Cisco Visual Networking Index 2017, there will be almost 106 Terabytes per second of global Internet traffic in 2021.

Although networks and web applications offer great digital experiences, only the secured ones can deliver the services safely. Since these applications deal with sensitive data and operations, they appear to be a major target to attackers with the intention to acquire confidential data, earn monetary gain, and perform several unlawful activities [2]. Hence, the security of networks and web application from various threats has become considerably important to ensure system confidentiality, availability and integrity.

The major principles of information security guarantee that only authenticated and authorized users are capable of reliably accessing secure information. However, these principles can be violated when vulnerabilities exist in complex software system and computer network [2]. These can be exploited by attackers to gain unauthorized access into the system. To prevent these security compromises, several defense mechanisms have been adopted by organizations. This include: cryptography, Firewall, Intrusion Detection System (IDS) etc. IDS provide the ability to identify security breaches in a system. Current methods used for these systems include anomaly and signature base detection. Signature detection makes use of databases which hold “definitions” of known attacks. An anomaly is any unusual event that occurs in the network environment. This method is used to detect unknown attacks. By presenting a literature survey of research on IDS, this paper aim at exploring the measures which have been suggested in existing studies to deal with the issues of network security and identify the current state of the art and future perspectives. The rest of the paper is arranged as follows; Section 2 focused on the key approaches for detecting intrusions and presents a comprehensive literature survey conducted to

explore various intrusion detection systems proposed by researchers, Section 3 presented the various Intrusion detection techniques while Section 4 is the conclusion.

II. INTRUSION DETECTION APPROACHES

There are basically four major approaches towards intrusion detection, namely: misuse, anomaly, policy and hybrid. Every detection approach works on a given set of principles. The current section briefly states the mechanism of each detection technique along with the challenges faced by them and a comprehensive review of the work done in this area.

2.1. Misuse-Based Intrusion Detection

The misuse-based approach utilizes a set of signatures which represent the patterns of known attacks to sieve harmful activities. It have the capacity of detecting known attacks more accurately with less false-positive rate but prove to be ineffective for detecting zero-day or unknown ones. The signature database needs to be up to date for identifying novel attacks which is very tedious and rigorous process since novel attacking techniques are being regularly discovered [3]. Another problem arises due to the vulnerabilities caused by the in-house developed web applications. Such vulnerabilities require specialist knowledge to produce their signatures. Also the variant of attack vector is unlimited; a slight variation in attack vector might easily deceive the signature-based Detector. Thus, signatures need to be more general to cover these variants. Nevertheless, there is a trade-off between deciding on detectors sensitivity and specificity level. Sensitive signatures increase the risk of having high false-positive alerts whereas highly selective ones are unable to detect attack variants.

The signature-based systems detect intrusions by using the knowledge of existing attacks. However, the data being monitored can vary. For example, the signature-based intrusion detection system by Almgren [4] examines log entries to identify malicious activities on the net server. The authors of the paper [5] came up with WebSTAT, an IDS based on STAT framework [6] that is capable of monitoring each HTTP requests and logs. The system depends on state transition analysis model to depict the attack scenario and have the capacity to detect multistep attacks. It allows the detection of attacks variant which have similar feature with the specified malicious behavior, in addition to being a misuse-based system. The study by García [7] solved the problem of manually writing the attack patterns by automating the entire process. The work basically employs text mining techniques to learn the behavior of the benign as well as malicious user from the web log file and generate the attack model.

2.2. Anomaly-Based Intrusion Detection

The anomaly-based technique is base on the assumption that malicious activities considerably differ from expected behavior and can be studied quantitatively [1]. The incoming events are examined to detect deviate from the normal ones. An anomaly-based system is capable of detecting unknown and novel attacks and handles the problems caused by custom vulnerabilities when properly trained. Besides having great prospect, there are some vital issues associated with the approach. Most anomaly-based IDS model desired behavior by using various machine learning techniques [8], and selecting the most suitable ones among them is a difficult issue. Besides, there is a challenge of deciding the optimal thresholds of the machine learning parameter. Choosing a high threshold value may raise the total number of undetected attacks while moderate configuration may result to higher false-positive alerts. Nevertheless, the solution is a function of the variability of the attribute being observed. The generalized model can handle the situation better when the web traffic is highly variable whereas low variable traffic needs a strict model to detect suspicious movement. Anomaly-based systems often produce high false-positive rates which may result to the blocking of legitimate requests. The assumption that attacks manifest unusual behavior is responsible for such abnormality since at times a benign user does exhibit strange behavior which might not have been recorded during the training phase.

Among the earliest work on anomaly-based system which solely focused on the detection of web-based attacks is the research by Kruegel and Vigna [9]. The authors developed multiple anomaly models to characterize the existing relationship between the server-side programs, client-side parameter, and the resultant parameter values. Several other studies [10 - 12] leveraged on this work. Kruegel [10] introduce additional features to capture more granular request behavior like the frequency of invoking a program by the client, whether an automated source issued the request or the client has circumvent any functionality.

To identify malicious behavior, Wressnegger [13] proposed a classification learning scheme. In this work, both the malicious and benign data was used to train the classifier. The use of Deterministic Finite Automata (DFA) as an anomaly detection scheme to learn normal request behavior was introduced by Ingham [14]. Since user requests are extremely variable in content, the researcher applied transformation rules in order to reduce the variability of the request as well as the complexity of the DFA model. The n-gram technique [15] which can analyze the structure of requests is seen as another promising anomalies detection mechanism. However, the complexity of the model using n-gram technique increases exponentially with the increasing size of the gram. Song [16] proposed Spectrogram model that reduce the complexity of n-gram-based technique by incorporating Markovian property with the assumption that the probability of future states depends solely on the present state. The authors in study [17] used a group of Hidden Markov Model to detect attacks related to the input validation. The research clearly handled the case of noisy data in the training set. In article [18], a comparative analysis of DFA and N-grams anomaly-based detection was conducted based on their effectiveness and efficiency.

Also, some researchers have attempted to add the context knowledge into the anomaly-based detector. The study by Düssel et al. [19] introduced the concept of a context-aware anomaly detection system by integrating the structural information of HTTP protocol into the scheme. The anomaly detection model was built using one-class support vector machine (OC-SVM) [20]. In [21], an innovative method for representing the packet payload via contextual n-grams was presented. The n-grams technique incorporates the structural attribute of protocol and their individual byte sequences in a single feature space. In addition, to confine suspicious pattern and identify the cause of an anomaly, the concept of feature shading [22] was introduced. Authors in work [11] addressed the inefficient of anomaly-based approach in providing attack description.

Normally, Anomaly-based models produce high false-positive alerts and this attracted the attention of many researchers. The study in [23] addressed the issue of false alerts produced by request which are both anomalous and benign using the concept of data compartmentalization. Rather than treating every anomalous request as malicious, it categorized requests base on their anomaly score and give clients limited or no access to the resources base on the score assigned to the request. Abnormal request are sent to the server which does not have privilege to access sensitive information. Paper [24] proposed Double Guard, an IDS which track malicious request by analyzing both the incoming user request at the web server and the request from the back end of the database. The model was developed by mapping client requests to their specific database queries. Researchers in [25] came up with ToKDoc by introducing a mechanism for repairing malicious request automatically in IDS. The proposed IDS have the ability to substitute the section of the request that appear suspicious with a safe value learned in the training stage. The work in [26] suggested an anomaly-based IDS called SWADDLER; by profiling the server-side program variables, the IDS is capable of analyzing the different states of an application within a session as the attack is carried out.

2.3. Policy-Based Intrusion Detection

Both the Misuse IDS and Anomaly-Based IDS have inherent limitations. It is impossible to maintain data of all likely attack vectors in Misuse detection. Similarly, anomaly detection cannot capture all valid user behaviours. Policy-based intrusion detection proffers solution to this challenge; by enforcing a set of rules, it set up boundaries between acceptable and unacceptable
events [27]. This approach help to solve the problem associated with the detection of unknown attacks as well as the classification of ordinary behavior into attack class. Despite its good promises, Policy-based detection suffers two major setbacks; the task of defining effective policies which is consistent (and in a logically correct state all through the system) is quite challenging and required the service of a security specialist. Since policies are establish by considering associated conditions, they are interrelated and an incoming event have the tendency to trigger more than one rule either within a policy or among policies thereby leading to internal conflict. This inter-policy and intra-policy relationship make policy management task very cumbersome. To

mitigate this problem, ontology-base system which has the capacity to simplify management task and policy specification was proposed [28-30]. Authors in [31] highlighted two ontology models: attack-centric and protocol – centric model. By providing the system with inference and reasoning capacity to chart different scenarios of security breakdown to a general semantic rule, Protocol ontology offers a baseline for the construction of the security measures of the attack model and reduces the number of signatures in detection process. Study in [32] inserted the Bayesian filter into an ontology-based system to mitigate attack. The model was trained using both malicious and benign data to boost the detection ability of the system. Further work on the enhancement of policy base detection efficiency was done by researchers in [33]. The study combined both policy-based Generic Authorization and Access control API (GAA-API) structure. It specified security policies for monitoring access to resources and responding to threatening activity using an Extended Access Control List (EACL) language [34].

2.4. Hybrid Intrusion Detection

In hybrid system, two or more intrusion detection mechanism are integrated into a single unified detection system [35-36] to achieve a higher performance by using the strength of more than one approach to overcome the limitations of individual system. Two important issues needed to be critically considered before choosing this method. First, hybrid systems are implemented using either single or double layer architecture. There is a challenge of establishing the right order for processing events in multiple components when using layered architecture. Resolving conflicts between the results classified by these components is another issue as a situation might arise where one component classifies an event into a safe group whereas the other labels the same as an intrusion [37]. The performance of hybrid-based systems depends on how well the different components are integrated.

A lot of research has been carried out based on Hybrid Intrusion Detection System. Paper [36] proposed an intelligent Intrusion Detection and Prevention System (IDPS) which integrates both the anomaly-based and signature-based detection approaches together with additional response action mechanism to handle the invader. To estimate the threat risk and design the response policies based on the level of severity, the authors leverage on DREAD model. The IDS proposed in [37] provides architecture which categorize events into safe, intrusive, or unknown class. Researchers in [38] used the features of attacks carried out by script kiddies in their proposed hybrid detection system. While they employed weighted graph in the anomalies detection, the signature component of the system utilize attack pattern presented by some web applications which participate mutually in the detection process. Using this system, the list of harmful request as created by the web application is forwarded to the collaborating web applications to boost detection.

III. INTRUSION DETECTION TECHNIQUES

The major intrusion detection techniques are: Machine Learning, Specification, Provenance graph and Statistical-based technique.

3.1 Machine Learning Technique

Machine learning techniques deals with the establishment of an explicit or implicit model which enables the patterns analyzed during detection process to be classified. It makes use of labeled data in training the behavioural model and has the ability to adjust its execution strategy as it acquires new information. Generally, the major drawback of this scheme is the resource expensive nature. The various machine learning-based techniques, their merit and demerit are discussed below.

A Bayesian model encodes probabilistic relationships among variables of interest. This technique is used in combination with statistical schemes. It has the ability to incorporate both prior knowledge and data, encode interdependencies between variables and accurately predict event [39]. Although, Bayesian networks requires higher computational effort [40]; the result obtain are similar to that of threshold-based system and to a large extent depend on the assumptions about the behaviour of the targeted system. Thus, a deviation in these hypotheses will lead to detection errors.

The two main approaches in Markov model are Markov chain and hidden Markov model. A Markov chain is a set of states joined together by certain transition probabilities, which determine the topology and the capacity of the model. The probabilities associated to the transitions are estimated from the normal behaviour of the target system in the first training phase. Then, anomaly is detected by comparing the given probability (anomaly score) from the observed sequences [41]. It was also used in packets inspection in the network IDS proposed by [42-43]. In either case, the model provided a good approach for the claimed profile; however, the result obtained is dependent upon the assumptions about the acceptable behaviour of the system.

Neural network (NN) is capable of simulating the operation of human brain. It is highly flexible and easily adapt to environmental changes. Researchers employed this detection mechanism in creating user profiles [44], predicting the next command from a sequence [45], identifying the intrusive behaviour of traffic sample [46], etc. The major problem associated with this system is the inability to provide a descriptive model that explains the reason for making a particular detection decision. Chonka et al. [47] proposed an intrusion detection and prevention mechanism to discover XML-Dos and HTTP-Dos attacks based on a back-propagation NN named Cloud Protector. The system was able to detect between 98% and 99% of the XML-Dos traffic within an average of 10-135ms. Vieira et al. [48] provided a detection system for cloud and grid computing, using an Artificial Neural Network (ANN) to identify attack in the anomaly component of the hybrid detection system. The model performance is satisfactory for real-time implementation with low data volume and minimal data volume and complexity requirement. Xiong et al. [49] introduced an anomaly detection method based on synergetic NN [50] for a cloud environment. The synergetic NN in a pattern recognition process perform anomaly detection by matching the testing data with the training data. This approach led to an improvement in the rate of false alerts and detection probability over the baseline used in evaluation.

Fuzzy logic was derived from fuzzy set theory in which reasoning is approximately inferred from classical affirm logic. When using Fuzzy technique in anomaly detection, the various features are considered as fuzzy variables [51] and an observation is labeled normal if it lies within a given interval [52]. Fuzzy logic has been found to be effective against port scans and probes attack but the resource consumption is high. Chan et al. [53] presented a fuzzy logic IDPS named FAP and FAR deployed over a public cloud platform for protection against web service attacks. These fuzzy-based systems have the capacity to detect and prevent famous web attacks with false alarm rate less than 1% and close to 100% detection accuracy rate. Wang et al. [54] provided a botnet detection system based on fuzzy pattern recognition of network traffic behaviour. The authors improved the detection accuracy of earlier systems detecting about 95% bot with less than 3.5% false positive alarm rate by accurately altering the membership functions utilized by the fuzzy pattern. Iyengar et al. [55] proposed a DDoS attack detection system using fuzzy logic. The authors devised an IDS by installing a fuzzy system in the cloud which checks the input traffic to discover DDoS attacks. This system specify the traffic type, help in traffic selection, analyze traffic, triggers an alarm and send a request to the routers for rejection of malicious packet entry. The system was able to reduce the cost of data transmission and storage functionality.

Genetic algorithms (GA) are subsets of evolutionary computation categorized as global search heuristics that use such technique as inheritance, mutation, selection and recombination. It is capable of developing classification rules [56] and choosing the most suitable features for the detection process [51].The advantage of Genetic algorithm is that it is flexible and employ a robust global search scheme that arrive at solution from various directions even without prior knowledge of the system. Pitropakis et al. [57] introduced a defense mechanism that uses Genetic algorithm. The detection scheme monitors the system calls produced in the various steps of an attack and compare the system calls with other executions of the same attack in addition to the normal system state before the attack took place. Like the Fuzzy logic, the resource consumption is high.

Clustering techniques group observed data into clusters, based on the similarity or distance

measure. It first select a representative point for each cluster and then classify each new data point as a member of a given cluster base on the proximity to the corresponding representative point [58]. The points that do not belong to any cluster (outliers) represent anomalies in the detection process. Several variant of clustering and outliers has been adopted by Researchers [59] developing a detection system. [60] applied the k-nearest neigbour approach of the Euclidean distance to define the membership of data points to a given clauster, whereas other systems use the Mahalanobis distance.[61] went a step further to associate certain degree of being an outliers to every point. The main advantage of clustering technique is that it required minimal effort to tune IDS since the occurrence of intrusion event is determine only from the raw data. In [62], the authors presented an IDS based on clustering utilizing learning automata to boost detection performance. The learning automata form the leadership of clusters by employing aggregate relative velocity and connectivity degree. Using this technique, about 93% of malicious activities were detected with lower false-positive rate. Also, it is easy for the system to adapt to changes of the nodes in the network. A clustering anomaly detection scheme which uses Local Outlier Factors (LOFs) and dimension reasoning rule was introduced by Huang at al.[63]. This technique is an effective method for Virtual Machine (VM) management. It detects anomalies, their source and behaviour through VM performance profile with the detection rate of 98% and also decreases false alert to 16.9%.

Anomaly intrusion detection system that employs more than one machine learning techniques is refer to as Hybrid machine learning technique IDS. Ganeshkumar and Pandeeswari [64] presented a hybrid system named adaptive neuro-fuzzy inference system (ANFIS) which is based on NNs and fuzzy logic. This is a Hypervisor Introspection designed for big data application and has the capacity to detect both the network-based and host-based activities even when it is not directly deployed in the virtual machines. The performance of the system was evaluated by comparing it with Naives Bayes, NBRF and ANN-based system using precision, recall and F-measure values. ANFIS gave higher detection accuracy and F-measure with a recall comparable with than other systems. In [65], the authors introduced an intrusion detection approach with the combination of GA and Fuzzy NN. The researchers applied Fuzzy NN to distinguish between remote-to-local and users-to-root attack while overcoming the low detection rate limitation of Fuzzy NN with GA. The system has the highest detection accuracy when compared with most other approaches derived from the individual machine learning scheme using a standard IDS benchmark data.

3.2. Statistical-based techniques

The Statistical techniques capture network traffic activity and generate a profile representing its stochastic behaviour. Metrics such as traffic rate, number of packets for every protocol, connection rate and number of dissimilar IP addresses are use in creating the profile. The two main network traffic dataset considered during the anomaly detection process using statistical-based technique are: dataset for the currently observed profile over time and that of the previously trained statistical profile. While the network events occur, the present profile is determined and the anomaly score which indicates the degree of irregularity for a given event (Such that the intrusion detection system will flag the occurance of anomaly when the score exceeds a given threshold) is estimated by comparison of the two behaviours. The earliest statistical approaches are univariant models which represent the independent variables as Gaussian random variables [66], thereby defining an acceptable range of value for each variable. Later, [67] proposed a multivariate model that considers the correlations existing between two or more metrics; thus, achieving a high level of discrimination through the combinations of related measures rather than considering them individually.

Statistical-based techniques have a lot of advantages. It has the capacity to learn the expected system behaviour from observation and as such does not require past knowledge of the normal activity of the target system. Also, statistical method can track malicious activities occurring over a long period of time and provide accurate notification. Nevertheless, it suffers some drawback. Setting the values of the various parameters in statistical-based IDS is quite challenging task. In addition, not every behaviour can be modeled by using stochastic methods. Furthermore, a statistical-base IDS is susceptible to invasion.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 12, pp: 70-81 www.ijaem.net ISSN: 2395-5252

3.3. Provenance graph based IDS

Provenance refers to meta-data describing how digital objects came to be in their current state [68].It gives a comprehensive, structured outlook of the activity going on in the system [69] by presenting as a directed acyclic graph, complex dependencies and causality relationships between digital objects. An inconsistent interdependency between data objects that deviate from those present in non-malicious execution signifies an intrusion. Provenance graphs show long range correlations and dependencies which allow for causal reasoning about intrusion [70] and strengthens adversarial robustness. This makes it possible to detect sophisticated attacks that remain undetected for extended length of time. To detect an intrusion using provenance graphs requires analyzing dynamic, attributed and steaming graphs. The development of fine-grained, whole-system provenance capture systems [71] has made this task more challenging since the graphs swiftly become extraordinarily large [69]. Though, the domain-specific knowledge of provenance graph, when applied, help to simplify the challenge of identifying anomalies via graph analysis.

In computing, data provenance is use to model syscall audit logs and several other event sequences. This process can also be reversed (to obtain provenance graph from audit data) in order to better understand system execution [71], though the precision of the graph generated in this case might be less [72]. Syscall-based provenance frameworks generate multiple disconnected graphs instead of a single graph of system execution because it lacks the ability to trace the interrelationships of kernel threads that do not utilize the syscall interface; thus the framework is not suitable for the detection of stealthy malicious events found in Advance Persistent Attacks (APTs). To handle this issue, Whole-system provenance was proposed. It runs at the operating system level, capturing all the activities going on in the system and the interactions among them [72]. For instance, Hi-Fi [72] and CamFlow [73] offer strong security and guarantees completeness with respect to information flow capture. This is highly desirable in the case of APT attack as it captures long-distance causal relationships which make contextualized analysis possible, even when the security-sensitive kernel objects are manipulated by an attack in order to hide its presence. CamFlow [73] ensures high quality, reliable recording of information flows among data objects [74-75] by implementing Linux Security Modules (LSM) framework [76]. LSM is capable of eliminating race conditions (TOCTTOU attacks) by inserting mediation points inside the kernel instead of placing it at the system call interface [77]. In building provenance base IDS it is assume that no component of the internal network is actively hostile and there exist related networks within a single organization that has an independent infrastructu

3.4. Specification Technique

Specification technique is classified into two: static specification and dynamic specification. Static specification mining examines source code to extract specifications and infer correct behaviour of the system [78-82]. For instance, Karim et al. [78] proposed synchronized pushdown systems (SPDS), and demonstrated how SPDS identifies security vulnerabilities due to exploited Crypto Application Programming Interface (APIs) in Android apps and Maven Central repositories. Wagner and Dean [82] also mine an automaton model from source code for their FSM-based intrusion detection system. They came up with a non-deterministic pushdown automaton (NDPDA), which develop an extensive model of the software system based on system calls. While NDPDA show high accuracy, the high memory overhead made them to be slow.

Dynamic specification mining scheme employ diverse system execution traces to identify inherent program rules [83-89]. For instance, Daikon is a dynamic analysis-based mechanism that generate (likely) invariants representing constraints on data value relations [90]. DIDUCE [91] merge data invariant inference and inspection in one tool for fault diagnosis purposes. It does not only monitor software dynamically, but also extends dynamic invariant detection to big programs. DySy[92] deduced a more accurate invariant than Daikon by combining the advantages of dynamic invariant inference and static analysis utilizing symbolic execution. Aliabadi et al. [93] proposed a dynamic specification mining technique which extracts specifications following the three dimensions of data, event, and time, and produce a 3D model for the system. This technique is quite accurate for detecting

security attack in Cyber Physical Systems (CPSs), but lead to noticeable overheads for complex CPS platforms[93].

IV. CONCLUSION

Cyber attacks have become a major concern for information security. IDS is one of the security mechanisms used to guard networks and applications against attacks. However, the methodology has been mostly used for monitoring the network-based attacks. Designing suitable IDS to prevent attacks still needs more focus by the research community. The paper provides essential information about IDS approach and techniques at a single place as a recipe to design a robust intrusion detection system which is capable of detecting and preventing attack.

REFERENCE

[1]. Nancy Agarwal and Syed Zeeshan Hussain (2018) A Closer Look at Intrusion Detection System for Web Applications**.** Security and Communication Networks 2018 (27).
[2]. Dieter Gollmann (2008) Securing Web applications. Information Security Technical Report 13(1):1-9
[3]. "Symantec - ISTR-Internet Security Threat Report" (2017) https://www. symantec. com /content/dam/symantec/docs/reports/istr-22-2017-en.pdf
[4]. M. Almgren, H. Debar, and M. Dacier (2000) A Lightweight Tool for Detecting Web Server Attacks. NDSS.
[5]. G. Vigna, W. Robertson, V. Kher, R. Kemmerer et al. (2003) A stateful intrusion detection system for world-wide web servers. IEEE Proceedings of the 19th Annual Computer Security Applications Conference, 34–43.
[6]. G. Vigna, S. T. Eckmann, and R. A. Kemmerer (2000) The STAT tool suite. IEEE Proceedings of the DARPA Information Survivability Conference and Exposition, DISCEX 2: 46–55.
[7]. J. J. García Adeva and J. M. Pikatza Atxa (2007) Intrusion detection in web applications using text mining. Engineering Applications of Artificial Intelligence 20 (4): 555–566.
[8]. C.F. Tsai (2009) Intrusion detection by machine learning: A review. Expert Systems with Applications 36 (10): 11994–12000.
[9]. C. Kruegel and G. Vigna (2003) Anomaly detection of web-based attacks. ACM Proceedings of the 10th Conference on Computer and Communications Security, 251–261.
[10]. C. Kruegel, G. Vigna, and W. Robertson (2005) A multi-model approach to the detection of web-based attacks. Computer Networks 48 (5): 717–738.
[11]. W. Robertson, G. Vigna, and C. Kruegel (2006) Using generalization and characterization techniques in the anomaly-based detection of web attacks. NDSS
[12]. F. Maggi, W. Robertson, C. Kruegel et al. (2009) Protecting a moving target: Addressing web application concept drift. Proceedings of the International Workshop on Recent Advances in Intrusion Detection, 21–40.
[13]. C. Wressnegger, G. Schwenk, D. Arp, and K. Rieck (2013) A close look on n-grams in intrusion detection: anomaly detection vs. classification. ACM Proceedings of the 2013 Workshop on Artificial Intelligence and Security, 67–76.
[14]. K. L. Ingham, A. Somayaji, J. Burge, and S. Forrest (2007) Learning DFA representations of HTTP for protecting web applications. Computer Networks 51 (5): 1239–1255.
[15]. R. Hamid, A. Johnson, S. Batta et al. (2005) Detection and explanation of anomalous activities: Representing activities as bags of event n-grams. IEEE Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition 5(1): 1031–1038.
[16]. Y. Song, A. D. Keromytis, and S. J. Stolfo (2009) Spectrogram: A Mixture-of-Markov-Chains Model for Anomaly Detection in Web Traffic. NDSS 9: 1–15.
[17]. I. Corona, D. Ariu, and G. Giacinto (2009) HMM-Web: A framework for the detection of attacks against web applications. IEEE Proceedings of the International Conference on Communications 9: 1–6.
[18]. L. Lin, C. Leckie, and C. Zhou (2010) Comparative Analysis of HTTP Anomaly Detection Algorithms: DFA vs N-Grams. IEEE Proceedings of the Fourth International Conference on Network and System Security, 113–119.
[19]. P. Düssel, C. Gehl, P. Laskov et al. (2008) Incorporation of application layer protocol syntax into anomaly detection. Proceedings of the International Conference on Information Systems Security, 188–202

[20]. D. M. J. Tax and R. P. W. Duin (1999) Data domain description using support vectors. ESANN 99.
[21]. P. Duessel, C. Gehl, U. Flegel et al. (2016) Detecting zero-day attacks using context-aware anomaly detection at the application-layer. International Journal of Information Security, 1–16.
[22]. K. Rieck and P. Laskov (2009) Visualization and explanation of payload-based anomaly detection. IEEE Proceedings of the European Conference on Computer Network Defense, 2: 29–36.
[23]. F. Valeur, G. Vigna, C. Kruegel et al. (2006) An anomaly-driven reverse proxy for web applications. ACM Proceedings of the Symposium on Applied Computing, 361–368.
[24]. M. Le, A. Stavrou, and B. B. Kang (2012) DoubleGuard: Detecting intrusions in multitier web applications. IEEE Transactions on Dependable and Secure Computing 9 (4): 512–525
[25]. T. Krueger, C. Gehl, K. Rieck et al. (2010) TokDoc: A self-healing web application firewall. ACM Proceedings of the Symposium on Applied Computing, 1846–1853.
[26]. M. Cova, D. Balzarotti, V. Felmetsger et al. (2007) Swaddler: An approach for the anomaly-based detection of state violations in web applications. Proceedings of the International Workshop on Recent Advances in Intrusion Detection, 63–86.
[27]. S. N. Chari and P. C. Cheng (2003) BlueBox: A policy-driven, host-based intrusion detection system. ACM Transactions on Information and System Security 6 (2): 173–200.
[28]. F. Abdoli, N. Meibody, and R. Bazoubandi (2010) An attacks ontology for computer and networks attack. Innovations and Advances in Computer Sciences and Engineering, 473–476.
[29]. J. Undercoffer, J. Pinkston, and A. Joshi (2004) A target-centric ontology for intrusion detection. Proceedings of the 18th International Joint Conference on Artificial Intelligence, 9–15.
[30]. F. Abdoli and M. Kahani (2009) Ontology-based distributed intrusion detection system. IEEE Proceedings of the 14th International CSI Computer Conference, 65–70.
[31]. A. Razzaq, K. Latif et al. (2014) Semantic security against web application attacks. Information Sciences 254: 19–38.
[32]. A. Razzaq, H. F. Ahmed, A. Hur et al. (2009) Ontology based application level intrusion detection system by using bayesian filter. IEEE Proceedings of the 2nd International Conference on Computer, Control and Communication, 1–6.
[33]. T. Ryutov, C. Neuman, K. Dongho et al. (2003) Integrated access control and intrusion detection for web servers. IEEE Transactions on Parallel and Distributed Systems 14 (9) 841–850.
[34]. T. Ryutov and C. Neuman (2002) The specification and enforcement of advanced security policies. IEEE Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks, 128–138.
[35]. S. Peddabachigari, A. Abraham, C. Grosan et al. (2007) Modeling intrusion detection system using hybrid intelligent systems. Journal of Network and Computer Applications 30 (1) 114–132.
[36]. A. Alazab, M. Hobbs, J. Abawajy et al. (2014) Using response action with intelligent intrusion detection and prevention system against web application malware. Information Management and Computer Security 22 (5): 431–449.
[37]. E. Tombini, H. Debar, L. Mé et al. (2004) A serial combination of anomaly and misuse IDSes applied to HTTP traffic. IEEE Proceedings of the 20th Annual Computer Security Applications Conference, pp. 428–437
[38]. M. Zachara (2016) Identification of Possible Attack Attempts Against Web Applications Utilizing Collective Assessment of Suspicious Requests. Transactions on Computational Collective Intelligence 12: 45–59.
[39]. D. Heckerman (1995) A tutorial on learning with Bayesian networks. Microsoft Research Technical Report MSRTR-95-06.
[40]. C. Kruegel , D. Mutz, W. Robertson and F. Valeur (2003) Bayesian event classification for intrusion detection. Proceedings of the 19th Annual Computer Security Applications Conference.
[41]. D. Yeung and Y. Ding (2003) Host-based intrusion detection using dynamic and static behavioral models. Pattern Recognition 2003; 36(1): 229–43

[42]. M. Mahoney and P.K. Chan (2003) An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection. Florida technology report CS-2003-02.
[43]. J.M Este´vez-Tapiador, P. Garcı´a-Teodoro and J.E Dı´az-Verdejo (2005) Detection of web-based attacks through Markovian protocol parsing. Proc. ISCC0 5: 457–62.
[44]. K. Fox, R. Henning, J. Reed and R. Simonian (1990) A neural network approach towards intrusion detection. Proceedings of 13th National Computer Security Conference, 125–34.
[45]. H. Debar, M. Becker and D. Siboni (1992) A neural network component for an intrusion detection system. IEEE Symposium on Research in Computer Security and Privacy, 240–50.
[46]. A.M Cansian, E. Moreira, A. Carvalho, and J.M. Bonifacio (1997) Network intrusion detection using neural networks. Proceedings of International Conference on Computational Intelligence and Multimedia Applications 97: 276–80
[47]. A. Chonka, Y. Xiang, W. Zhou, and A. Bonti (2011) Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks. Journal of Network and Computer Applications 34 (4) 1097-1107.
[48]. K. Vieira, A. Schulter, C. Westphall, and C. Westphall (2010) Intrusion Detection for Grid and Cloud Computing. IT Professional 12: 38-43.
[49]. W. Xiong, H. Hu, N. Xiong, L. T. Yang, W.-C. Peng, X. Wang, et al. (2014) Anomaly secure detection methods by analyzing dynamic characteristics of the network traffic in cloud communications. Information Sciences 258: 403-415.
[50]. H. Haken (2013) Synergetic computers and cognition: A top-down approach to neural nets. Springer Science & Business Media.
[51]. S.M Bridges and R.B. Vaughn (2000) Fuzzy data mining and genetic algorithms applied to intrusion detection. Proceedings of the National Information Systems Security Conference, 13–31.
[52]. J. E. Dickerson (2000) Fuzzy network profiling for intrusion detection. Proceedings of the 19th International Conference of the North American Fuzzy Information Processing Society, 301–6.
[53]. G.Y. Chan, F.F. Chua, and C.S. Lee (2016) Intrusion detection and prevention of web service attacks for software as a service: Fuzzy association rules vs fuzzy associative patterns. Journal of Intelligent and Fuzzy Systems 31 (2): 749-764.
[54]. K. Wang, C.Y. Huang, L.Y. Tsai, and Y.D. Lin (2014) Behavior-based botnet detection. Parallel Security and Communication Networks 7(11): 1849-1859.
[55]. N. C. S. Iyengar, A. Banerjee, and G. Ganapathy (2014) A fuzzy logic based defense mechanism against distributed denial of service attack in cloud computing environment. International Journal of Communication Networks and Information Security, 6 (3): 233-245.
[56]. Y. Liao and V.R Vemuri (2002) Use of K-nearest neighbor classifier for intrusion detection. Computers and Security 21: 439–48.
[57]. N. Pitropakis, D. Anastasopoulou, A. Pikrakis, and C. Lambrinoudakis (2014) If you want to know about a hunter, study his prey: detection of network based attacks on KVM based cloud environments. Journal of Cloud Computing 3(1) 20
[58]. L. Portnoy, E. Eskin and S.J Stolfo (2001) Intrusion detection with unlabeled data using clustering. ACM Proceedings of the Workshop on Data Mining Applied to Security.
[59]. K. Sequeira and M. Zaki (2002) ADMIT: anomaly-based data mining for intrusions. ACM Proceedings of the 8th SIGKDD International Conference on Knowledge Discovery and Data Mining, 386–95.
[60]. Y. Liao and V.R Vemuri (2002) Use of K-nearest neighbor classifier for intrusion detection. Computers and Security 21: 439–48.
[61]. M. Breunig, H.P Kriegel, R.T Ng and L.J Sander (2000) Identifying density-based local outliers. ACM Proceedings of the International Conference on Management of Data, 93–104.
[62]. N. Kumar, J. P. Singh, R. S. Bali, S. Misra, and S. Ullah (2015) An intelligent clustering scheme for distributed intrusion detection in vehicular cloud computing. Cluster Computing 18(3):1263-1283.
[63]. T. Huang, Y. Zhu, Y. Wu, S. Bressan, and G. Dobbie (2016) Anomaly detection and identification scheme for VM live migration in cloud infrastructure. Future Generation Computer Systems 56:736-745
[64]. P. Ganeshkumar, and N. Pandeeswari (2016) Adaptive neuro-fuzzy-based anomaly

detection system in cloud. International Journal of Fuzzy Systems 18 (3): 367-378.

[65]. S. Raja and S. Ramaiah (2016) An Efficient Fuzzy-Based Hybrid System to Cloud Intrusion Detection. International Journal of Fuzzy Systems 19(1):1-16.

[66]. D.E Denning and P.G Neumann (1985) Requirements and model for IDES – a real-time intrusion detection system. Computer Science Laboratory, SRI International Technical Report #83F83- 01-00.

[67]. N. Ye, S.M Emran, Q. Chen and S. Vilbert (2002) Multivariate statistical analysis of audit trails for host-based intrusion detection. IEEE Transactions on Computers, 51(7).

[68]. X. Han, T. Pasquier and M. Seltzer (2018) Provenance-based Intrusion Detection: Opportunities and Challenges. ACM ISBN 123-4567-24-567.

[69]. A. M Bates, D. Tian, K.R. Butler, and T. Moyer (2015) Trustworthy Whole-System Provenance for the Linux Kernel. USENIX Security Symposium, 319–334.

[70]. L. Akoglu, H. Tong, and D. Koutra (2015) Graph based anomaly detection and description: a survey. Data Mining and Knowledge Discovery 29(3) 626–688

[71]. S. Chen, J. Xu, E. C. Sezer et al. (2005) Non-Control-Data Attacks Are Realistic Threats. USENIX Security Symposium, 5.

[72]. D. J. Pohly, S. McLaughlin, P. McDaniel, and K. Butler (2012) Hi-fi: collecting high-fidelity whole-system provenance. ACM Computer Security Applications Conference, 259–268.

[73]. T. Pasquier, X. Han, M. Goldstein, T. Moyer, D. Eyers, M. Seltzer, and J. Bacon (2017) Practical whole-system provenance capture. ACM Symposium on Cloud Computing, 405–418.

[74]. L. Georget, M. Jaume, F. Tronel, G. Piolle, and V. V. T. Tong (2017) Verifying the reliability of operating system-level information flow control systems in linux. IEEE/ACM International Workshop on Formal Methods in Software Engineering, 10–16.

[75]. T. Pasquier, X. Han, T. Moyer, A. Bates, O. Hermant, D. Eyers, J. Bacon, and M. Seltzer (2018) Runtime analysis of whole-system provenance. ACM, Conference on Computer and Communications Security (CCS'18)

[76]. J. Morris, S. Smalley, and G. Kroah-Hartman (2002) Linux security modules: General security support for the linux kernel. USENIX Security Symposium.

[77]. T. Jaeger, A. Edwards, and X. Zhang (2004) Consistency analysis of authorization hook placement in the linux security modules framework. ACM Transactions on Information and System Security (TISSEC), 7(2):175–205

[78]. J. Späth, K. Ali, and E. Bodden (2019) Context-flow and field-sensitive data-flow analysis using synchronized pushdown systems. ACM Proceedings of Programming Language. 3 (POPL) 1–29

[79]. S. Shoham, E. Yahav, S.J. Fink and M. Pistoia (2008) Static specification mining using automata-based abstractions. IEEE Trans. Software Engineering 34 (5) 651–666.

[80]. M. Gabel, and Z. Su (2008) Symbolic mining of temporal specifications. ACM Proceedings of the 30th International Conference on Software Engineering, 51–60.

[81]. J.T. Giffin, S. Jha, and B.P. Miller (2004) Efficient context-sensitive intrusion detection. NDSS.

[82]. D. Wagner, R. Dean (2001) Intrusion detection via static analysis. IEEE Proceedings on Security and Privacy,156–168.

[83]. P. Bian, B. Liang, W. Shi, J. Huang, Y. Cai and Nar-miner (2018) discovering negative association rules from code for bug detection. ACM Proceedings of the 26th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 411–422.

[84]. P. Bian, B. Liang, Y. Zhang, C. Yang, W. Shi , and Y. Cai (2018) Detecting bugs by discovering expectations and their violations. IEEE Trans. Software Engineering.

[85]. B. Liang, P. Bian, Y. Zhang, W. Shi, W. You, and Y. Cai (2016) Antminer: mining more bugs by reducing noise interference. ACM Proceedings of the 38th International Conference on Software Engineering, 333–344.

[86]. C. Lemieux, D. Park, and I. Beschastnikh (2015) General specification mining. IEEE/ACM 30th International Conference on Automated Software Engineering, 81–92

[87]. I. Beschastnikh, Y. Brun, J. Abrahamson, M.D. Ernst and A. Krishnamurthy (2015) Using declarative specification to improve the understanding, extensibility and comparison of model-inference algorithms, IEEE Trans. Software Engineering 41 (4) 408–428.

[88]. J. Abrahamson, I. Beschastnikh, Y. Brun, and M.D. Ernst (2014) Shedding light on distributed system executions. ACM Companion Proceedings of the 36th International Conference on Software Engineering, 598–599.
[89]. M.D. Ernst, J.H. Perkins, P.J. Guo, S. McCamant, C. Pacheco, M.S. Tschantz, and C. Xiao (2007) The daikon system for dynamic detection of likely invariants. Society of Computer Programming. 69 (1–3).
[90]. M.D. Ernst, J. Cockrell, W.G. Griswold, and D. Notkin (2001) Dynamically discovering likely program invariants to support program evolution. IEEE Trans. Software Engineering. 27 (2) 99–123.
[91]. S. Hangal, and M.S. Lam (2002) Tracking down software bugs using automatic anomaly detection. IEEE Proceedings of the 24rd International Conference on Software Engineering, 291–301.
[92]. C. Csallner, N. Tillmann, and Y. Smaragdakis (2008) Dysy: dynamic symbolic execution for invariant inference. ACM Proceedings of the 30th International Conference on Software Engineering, 281–290.
[93]. M.R. Aliabadi, A.A. Kamath, J. Gascon-Samson, and K. Pattabiraman (2017) Artinali: dynamic invariant detection for cyber-physical system security. ACM Proceedings of the 11th Joint Meeting on Foundations of Software Engineering, 349–361.