Journal of AL-Qadisiyah for computer science and mathematics Vol.10 No.2 Year 2018
ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

Comp Page 88 - 96

An Adaptive Intrusion Detection System by using Decision Tree

Osamah Adil Raheem **Esraa Saleh Alomari**
College of Computer Science and Information Technology
University of Wasit
oalmusawi@uowasit.edu.iq ealomari@uowasit.edu.iq

Recived : 8\3\2018 **Revised : 14\3\2018** **Accepted : 9\4\2018**

Available online : 17/5/2018

DOI: 10.29304/jqcm.2018.10.2.387

Abstract:

In recent decades, data security has turned into a new vision in data innovation as the quantity of PC protection breaks are presented to an explosion in the number of security incidents. An assortment of Sequence Discovery System have been utilized for shielding PCs and systems from pernicious system based or have based assaults by utilizing conventional measurable strategies to new information mining approaches in a years ago. In any case, the present monetarily accessible interruption identification frameworks are mark based that are not equipped for identifying obscure assaults. In this paper, we introduce another learning calculation for abnormality based system interruption identification framework utilizing choice tree calculation that recognizes assaults from ordinary practices and distinguishes diverse kinds of interruptions. Test comes about on the KDD-99 bench-mark organize interruption discovery data-set exhibit that the suggested algorithm resulted 98.5% Discovery rate in comparing with other executing techniques.

Key Words. Network Security, Sequence Discovery System, and decision tree.

1.Introduction. As propels in PC organize innovation grow for interchanges and business as of late, the rate of interruptions increment more than twofold consistently. Interruption discovery is the way toward distinguishing activities that endeavor to bargain the privacy, uprightness or accessibility of PCs or systems.

The utilization of information digging calculations for recognizing interruptions is currently considered to manufacture productive and versatile interruption identification frameworks (Sequence Discovery System) that identify unapproved exercises of a PC framework or system. Sequence Discovery System was first presented in [1], and later in 1986 an author [2] proposed a few models for Sequence Discovery System in light of insights,

Markov chains, time-arrangement, and so on [2]. Irregularity based interruption identification utilizing information mining calculations, for example, Supervised Algorithms and Unsupervised Algorithms are fluffy rationale demonstrate, and hereditary calculation have been broadly utilized by analysts to enhance the execution of Sequence Discovery System [3]-[8]. Be that as it may, the present economically accessible Sequence Discovery Systems are mark based. Mark based Sequence Discovery System performs design coordinating strategies to coordinate an assault design comparing to known assault designs in the database and results "low False Positives (FP)", yet it requires update the rules for the system to fit for recognizing obscure assaults. Then again, irregularity based Sequence Discovery System constructs models of ordinary conduct and naturally identifies atypical practices. Peculiarity identification procedures recognize new kinds of interruptions as deviations from ordinary utilization [9], yet the downside of these strategies is the rate of False Positives (FP). The utilization of information digging calculations for irregularity based Sequence Discovery System are to incorporate an insightful operator in the framework that can recognize the known and obscure assaults or interruptions.

Interruption recognition frameworks (Sequence Discovery System) assemble and break down data from an assortment of frameworks and system hotspots for indications of interruptions. Sequences Discovery System can be have based or organize based frameworks. Host-based Sequence Discovery System situated in servers to look at the inward interfaces and system based Sequence Discovery System screen the system traffics for recognizing interruptions . System based Sequence Discovery System perform bundle logging, continuous activity investigation of IP system, and tries to find if an interloper is endeavoring to break into the system. The real capacities performed by Sequence Discovery System are: (1) observing clients and frameworks movement, (2) evaluating framework design, (3) surveying the information records, (4) perceiving known assaults, (5) distinguishing unusual exercises, (6) overseeing review information, (7) featuring typical exercises, and (8) rectifying framework setup blunders. An assortment of Sequence Discovery System have been utilized for ensuring PCs and systems in a decades ago, yet at the same time there a few issues that ought to be consider in the present Sequence Discovery System like low location exactness, unequal identification rates for various sorts of assaults, and high False Positives. In this paper, another choice tree proposed for learning calculation to detect system assaults, which enhances the recognition rates and diminishes False Positives (FP) utilizing KDD-99 bench-mark arrange interruption location data-set in correlation with other existing techniques.

The remnants of the paper are sorted out as takes after. Segment 2 shows the different methodologies for irregularity based interruption discovery frameworks. Our proposed calculation for abnormality based system interruption recognition framework is presented in Section 3. In Section 4, the exploratory outcomes are communicated. At long last, our decisions and future works are specified in Section 5.

2. SEQUENCE DISCOVERY SYSTEM FOR ANOMALY DETECTION

In 1980, the idea of Sequence Discovery System started in [1]; a danger order demonstrate that builds up a security checking observation framework in light of distinguishing peculiarities in client conduct. In [1] model dangers are named outside infiltrations, inner entrances, and misfeasance. Outer infiltrations are interruptions in PC framework by outside interlopers, who don't have any approved access to the framework that they assault. Misfeasance is characterized as the abuse of approved access of both to the framework and to its information. In 1986, [2] specified a few models for business Sequence Discovery System improvement in view of insights, Markov chains, time-arrangement. In the mid 1980's, a research institute built up a Sequence Discovery Expert System that consistently observed client conduct and distinguished suspicious occasions [10].

Later SRI built up an enhanced adaptation of called the Next-Generation Sequence Discovery System [11], [12] that could work progressively for consistent observing of client action or could keep running in a cluster mode for occasional investigation of the review information, a review information is a record of exercises created by the working framework that are logged to a document in sequentially arranged request. The Sequence Discovery Expert System empower the framework to look at the present exercises of the client/framework/connect with the evaluated interruption discovery factors put away in the profile and afterward raise a caution if the present action is adequately a long way from the put away reviewed movement. In 1988, a measurable inconsistency based Sequence Discovery System was contributed by [13], which utilized both client and gathering based peculiarity recognition techniques. In this framework, a scope of qualities were viewed as typical for each characteristic and amid a session if a property fell outside the ordinary range then an alert raised. It was intended to recognize six kinds of interruptions: endeavored break-ins by unapproved clients, disguise assaults, infiltration of the security control framework, spillage, foreswearing of administration, and noxious utilize . (SPADE) [14] is a measurable oddity interruption recognition framework that is accessible as a module for SNORT that an open source organize interruption identification and counteractive action framework (NIDPS) created by Source fire [15], [16] .

An author proposed a similarity between the human safe framework and interruption identification that included breaking down a program's framework call successions to manufacture an ordinary profile [17], which broke down a few UNIX systems, based projects like send mail, ip, and so on. On the off chance that the arrangements veered off from the ordinary succession profile then it considered as an assault. The framework they created was just utilized disconnected utilizing beforehand gathered information and utilized a very basic table-query calculation to take in the profiles of projects. In 2000, [18] built up an irregularity based interruption discovery framework that utilized credulous Bayesian system to perform interruption recognizing on movement blasts. In 2003 [19] proposed a multisensory combination approach utilizing Bayesian classifier for characterization and concealment of false cautions that the yields of various Sequence Discovery System sensors were accumulated to create single alert . Around the same time, [20] proposed an oddity based interruption location conspire utilizing vital parts investigation (PCA), where PCA was connected to decrease the dimensionality of the review information and land at a classifier that is an element of the vital segments. In another paper , [21] proposed an irregularity based interruption identification utilizing concealed Markov models that registers the example probability of a watched arrangement utilizing the forward or in reverse calculation for recognizing odd conduct from ordinary practices. [22] Proposed characterization based peculiarity discovery utilizing inductive guidelines to portray arrangements happening in ordinary information. In 2000, [23] built up the Fuzzy Intrusion Recognition Engine (FIRE) utilizing fluffy rationale that procedure the system input information and produce fluffy sets for each watched highlight and afterward the fluffy sets are utilized to characterize fluffy guidelines to distinguish singular assaults . FIRE makes and applies fluffy standards to the review information to group it as ordinary or strange. In another paper, [24] displayed the odd system activity discovery with self-arranging maps utilizing DNS and HTTP administrations for organize based Sequence Discovery System that the neurons are prepared with ordinary system movement then constant system information is nourished to the prepared neurons, if the separation of the approaching system movement is more than a preset edge then it rises a caution. Another system have been proposed based irregularity recognition utilizing information mining methods created by Minnesota Sequence Discovery System in 2004 [25].

Journal of AL-Qadisiyah for computer science and mathematics Vol.10 No.2 Year 2018
ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

3. LEARNING ALGORITHM

A. The "Decision Tree (DT)" is intense and prominent information digging calculation for basic leadership and grouping issues. It has been utilizing as a part of numerous genuine applications like restorative analysis, radar flag arrangement, climate forecast, credit endorsement, and misrepresentation recognition and so on. DT can be developed from substantial volume of data-set with many characteristics, on the grounds that the tree measure is free of the data-set estimate.

A "Decision tree has three principle parts: hubs, leaves, and edges". Every hub is named with a characteristic by which the information is to be parceled. Every hub has various edges, which are marked by conceivable estimations of the characteristic. An edge interfaces either two hubs or a hub and a leaf .

B. Leaves are marked with a choice incentive for arrangement of the information. To settle on a choice utilizing a choice Tree, begin at the root hub and take after the tree down the branches until the point that a leaf hub speaking to the class is come to . Every Decision Tree speaks to an administerator set, which arranges information as per the properties of data-set. The DT building calculations may at first form the tree and after that prune it for more successful characterization. With pruning strategy, bits of the tree might be expelled or consolidated to diminish the general size of the tree. The time and space many-sided quality of developing a choice tree relies upon the extent of the informational index , the quantity of properties in the informational collection, and the state of the subsequent tree. Choice trees are utilized to arrange information with normal qualities. The ID3 calculation constructs choice tree utilizing data hypothesis, which pick part characteristics from an informational collection with the most astounding data pick up [26]. The entropy figuring is appeared in condition 1. Given probabilities p1, p2,..,pn for various classes in the informational index

"Entropy: H(p1,p2,…pn) = $\sum_{i=1}^{\$}(pilog(1/pi))$ " (1)

Given an index values, *D*, *H(D)* finds the measure of entropy in class based subsets of the informational collection. At the point when that subset is part into s new subsets *R = {D1, D2,… , Dn}* utilizing some characteristic, we can again take a gander at the entropy of those subsets. A subset of informational collection is totally requested and does not require any additionally split if all cases in it have a place with a similar class. The ID3 calculation ascertains the data pick up of a split by utilizing condition 2 and picks that split which gives most extreme data pick up.

"*Gain*(D,S)= H(D)- $\sum_{i=1}^{\$} p(Di)H(Di)$ " (2)

"Regression Trees" is a procedure of creating a twofold tree for basic leadership [28]. Truck handles missing information and contains a pruning system. The SPRINT (Scalable Parallelizable Induction of Decision Trees) calculation utilizes a contamination work called GINI list to locate the best split [29].

"*GINI* (D) = 1-∑ pj2 " (3)

Where, pj is the likelihood of class Cj in informational index D. The integrity of a split of D into subsets D1 and D2 is characterized by

"*GINISPLIT*(D) = n1/n(GENO(D1))+ n2/n(GENO(D2))" (4)

A. Improved Learning Algorithm

In a particular data-set, the first step the algorithm sets the weight's values for each example of data-set; W_i equal to 1/*n*, where *n* is the number of all examples in data-set . After that the algorithm estimates the prior probability $P(C_j)$ for each class by summing the weights that how often each class occurs in the data-set. Also for each attribute, A_i, the number of occurrences of each attribute value A_{ij} can be counted by summing the weights to determine $P(A_{ij})$. In An algorithm C4.5 [27], which is an enhanced of ID3 version algorithm uses highest "*Gain Ratio"* in equation 3 for splitting issue that ensures a larger than average information that have been gained .

"*GainRatio(D,S)*= $\frac{Gain(D,S)}{(\frac{D1}{D}\ldots\frac{D\$}{D})}$ "

(5)

The C5.0 computation enhances the execution of building trees utilizing boosting, however is a way to deal with consolidating diverse classifiers. Be that as it may, boosting does not generally help when the preparation information contains a great deal of clamor. At the point when C5.0 plays out a grouping

another hand , the conditional probabilities $P(A_{ij} \mid C_j)$ are estimated for all values of attributes by summing the weights how often each attribute value occurs in the class C_j. Next, the algorithm uses these probabilities to update the weight's values for each example in the data-set. It's executed by multiplying the probabilities of the different features values from the examples. Suppose the example e_i has independent attribute values $\{A_{i1}, A_{i2}, \ldots, A_{ip}\}$. We already know $P(A_{ik} \mid C_j)$, for each class C_j and attribute A_{ik}. We then estimate $P(e_i \mid C_j)$ by

$$\text{“}P(e_i \mid C_j) = P(C_j) \prod_{k=1 \to p} P(A_{ij} \mid C_j)\text{”} \quad (6)$$

To update the weight, the estimation of likelihood for e_i in each class C_j. The probability that e_i is in a class is the product of the conditional probabilities. The posterior probability $P(C_j \mid e_i)$ is then found for each class for each attribute value .

The last step, the algorithm will calculates the information gain using updated weights and builds a tree for decision. The main procedure for the algorithm is described below :

Steps: Tree-Creation
Input:datat
a-set *D*
Output:
decision
tree *T*
Procedure:

1. Which are the weights in D, Wi=1/n, where n is the total number of the examples.
2. Figure out the prior probabilities P(Ci) for each class Ci
3. Compute the subjunctive probabilities P(Aij | Cj) for each feature values in D. “P(Aij | Cj) = P(Aij)/Ci
4. Figure out the posterior probabilities for each example in *D*. “$P(e_i \mid C_j) = P(C_j) \prod P(A_{ij} \mid C_j)$”
5. Update the weights by; “Wi= PML(Cj|ei)”
6. Discover the splitting attribute with the highest info. That gaining using the weights updated, W_i in *D*.T = construct the root node and label with splitting attribute.
7. For *T*, *D* = database created by applying splitting predicate to *D*, and continue steps 1 to 7 until each final subset belong to the same class or leaf node created.
8. After the decision tree construction is completed the algorithm terminates.

4. Experimental Results Analysis

A. Sequence Discovery Data-set

The KDD-99 data-set has been used as a part of the third International Knowledge Discovery and Tools for Data Mining Competition used for building a system interruption identifier, a prescient model fit for recognizing interruptions and typical system associations [30]. However, being impacted with numerous interruptions assaults and got much consideration in the examination group of versatile interruption discovery. The KDD-99 data-set challenge utilizes an adaptation of DARPA-98 data-set. In KDD-99 data-set, every case speaks to quality estimations of a class in the system information stream, and each class is named either ordinary or assault. The attack types in KDD-99 data-set sorted into five primary types as shown in Table 2.

1- Remote to User (R-2-L) is a strike that a remote customer acquires passageway of an area (customer/account) by targeting over a framework correspondence, which join send-letters. Client to Root (U-2-R) is an attack that an intruder begins with the passageway of a common customer record and a short time later transforms into a root-customer by abusing diverse gaps of the system. Most basic adventures of U-2-R assaults are customary cradle floods and stack module.

2- "Denial of Service (DoS)": the source of the handling power or memory of a setback target unreasonably possessed or too full caused by DoS attacks, making it difficult to manage genuine requests. DoS attacks are requested in perspective of the organizations that an attacker renders unavailable to honest to goodness customers like Apache 2, arrive, mail bomb, back.

3- Probing: A gatecrasher with a guide of targeted and administrations that are accessible on a system can utilize the data to search for misuses. And is an assault that outputs a system to assemble data or find known system vulnerabilities. In KDD-99 data-set these four attack classes (DoS, U2R, R2L, and probe) are divided into 20 different attack classes that illustrated in Table 1.

TABLE 1
The Types of Attacks in KDD-99 Data-set

Four Attack Classes	20 Attacks Classes
"Denial of Service (DoS)"	"land, back, neptune, pod, smurt"
Remote to User	"imap, ftp_write, guess_passwd, , warezmaster multihop, spy, warezclient"
User to Root	"perl, loadmodule, buffer_overflow, rootkit"
"Probing"	"ipsweep, nmap, satan , portsweep"

There are 41 input qualities in KDD-99 data-set for each system association that have either discrete or nonstop esteems and partitioned into three gatherings . The primary gathering of characteristics is the fundamental highlights of system association, which incorporate the "span", "model", "benefit", number of bytes from source IP addresses or from goal IP locations, and a few banners in Transfer Protocol. The second gathering of qualities in KDD-99 is made out of the substance highlights of system associations and the third gathering is made out of the factual highlights that are figured either by a period window or a window of certain sort of associations.

Table 2 demonstrates the quantity of cases of 10% preparing illustrations and 10% testing cases in KDD-99 data-set. There are some new assault cases in testing information, which is no present in the preparation information .

TABLE 2
Attack Types' in KDD-99 Data

Attacks Type	Training Cases	Testing Cases
"legitimate"	97233	60423
"Denial of Service"	391445	237593
"Remote to User"	1326	8516
"User to Root"	62	80
"Probing"	4304	4276
Examples		310868

A. Analysis of Experiment

The evaluation of the performance for the enhanced algorithm for sequence discovery, 5-class have been executed to classy the anomalies using KDD-99 data-set. All the experiments were executed by using (Intel Core i7-2640M Processor 2.80 GHz processor with 8 GB of RAM). The results of the comparison of the enhanced algorithm along with ID3 and C4.5 as shown in Table 3 .

TABLE 3
41 Attributes for Comparison

Method	legitimate	Probe	DoS	U-2-R	R-2-L
New Algorithm (Data Rate %)	98.67	98.51	98.54	98.52	97.26
New Algorithm (False Positive %)	0.09	0.53	0.07	0.17	7.89
ID3 (Data Rate %)	97.67	96.45	97.51	43.23	92.67
ID3 (False Positive %)	0.11	0.66	0.05	0.18	10.12
C4.5 (Data Rate %)	98.45	97.90	97.56	49.23	94.70
C4.5 (False Positive %)	0.14	0.57	0.09	0.17	11.09

5. CONCLUSION

This paper presents another learning approach for peculiarity based framework intrusion revelation using decision tree, which modifies the weights of data-set in light of probabilities and split the data-set into sub-data-set until all the sub-data-set has a place with a comparable class. In this paper, the Sequence Discovery System using decision tree has been developed. The trial comes to fruition on KDD-99 benchmark data-set demonstrate that proposed estimation achieved high area rate on different sorts of framework strikes. The future research issues will be to test it extensively to get higher detection rate .

ACKNOWLEDGMENT

University of Wasit, Computer Science and Information Technology collage have supported this research.

References

[1] Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16-24.

[2] Dorothy E. Denning, "An intrusion detection model," IEEE Transaction on Software Engineering, SE-13(2), 1987, pp. 222-232.

[3] Barbara, Daniel, Couto, Julia, Jajodia, Sushil, Popyack, Leonard, Wu, and Ningning, "ADAM: Detecting intrusion by data mining," IEEE Workshop on Information Assurance and Security, West Point, New York, June 5-6, 2001.

[4] Sindhu, S. S. S., Geetha, S., & Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach. Expert Systems with applications, 39(1), 129-141..

[5] Mukkamala S., Janoski G., and Sung A.H., "Intrusion detection using neural networks and support vector machines," In Proc. of the IEEE International Joint Conference on Neural Networks, 2002, pp.1702- 1707.

[6] J. Luo, and S.M. Bridges, "Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection," International Journal of Intelligent Systems, John Wiley & Sons, vol. 15, no. 8, 2000, pp. 687- 703.

[7] Jonnalagadda, S. K., & Reddy, R. P. (2013). A literature survey and comprehensive study of intrusion detection. International Journal of Computer Applications, 81(16), 40-47.

[8] Shon, T., Seo, J., & Moon, J. (2005, October). SVM approach with a genetic algorithm for network intrusion detection. In International Symposium on Computer and Information Sciences (pp. 224-233). Springer, Berlin, Heidelberg.

[9] Lazarevic, A., Ertoz, L., Kumar, V., Ozgur,. A., Srivastava, and J., "A comparative study of anomaly detection schemes in network intrusion detection," In Proc. of the SIAM Conference on Data Mining, 2003.

[10] Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer networks, 51(12), 3448-3470.

[11] D. Anderson, T. Frivold, A. Tamaru, and A. Valdes, "Next generation intrusion detection expert system (NIDES)," Software Users Manual, Beta-Update Release, Computer Science Laboratory, SRI International, Menlo Park, CA, USA, Technical Report SRI-CSL-95-0, May 1994.

[12] D. Anderson, T.F. Lunt, H. Javitz, A. Tamaru, and A. Valdes, "Detecting unusual program behavior using the statistical component of the next generation intrusion detection expert system (NIDES)," Computer Science Laboratory, SRI International, Menlo Park, CA, USA, Technical Report SRI-CSL-95-06, May 1995.

[13] S.E. Smaha, and Haystack, "An intrusion detection system," in Proc. of the IEEE Fourth Aerospace Computer Security Applications Conference, Orlando, FL, 1988, pp. 37-44.

[14] N. Ye, S.M. Emran, Q. Chen, and S. Vilbert, "Multivariate statistical analysis of audit trails for host-based intrusion detection," IEEE Transactions on Computers 51, 2002, pp. 810-820.

[15] Martin Roesch, "SNORT: The open source network intrusion system," Official web page of Snort at http://www.snort.org/

[16] L. C. Wuu, C. H. Hung, and S. F. Chen, "Building intrusion pattern miner for sonrt network intrusion detection system," Journal of Systems and Software, vol. 80, Issue 10, 2007, pp. 1699-1715.

[17] S. Forrest, S.A. Hofmeyr, A. Somayaji, T.A. Longstaff, "A sense of self for Unix processes," in Proc. of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, USA, 1996, pp. 120-128.

[18] A. Valdes, K. Skinner, "Adaptive model-based monitoring for cyber attack detection," in Recent Advances in Intrusion Detection Toulouse, France, 2000, pp. 80-92.

[19] C. Kruegel, D. Mutz, W. Robertson, F. Valeur, "Bayesian event classification for intrusion detection," in Proc. of the 19th Annual Computer Security Applications Conference, Las Vegas, NV, 2003.

[20] M.L. Shyu, S.C. Chen, K. Sarinnapakorn, L. Chang, "A novel anomaly detection scheme based on principal component classifier," in Proc. of the IEEE Foundations and New Directions of Data Mining Workshop, Melbourne, FL, USA, 2003, pp. 172-179.

[21] D.Y. Yeung, Y. Ding, "Host-based intrusion detection using dynamic and static behavioral models," Pattern Recognition 36, 2003, pp. 229- 243.

[22] Lee, W., Stolfo, S. J., & Mok, K. W. (1999). A data mining framework for building intrusion detection models. In Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on (pp. 120-132). IEEE.

[23] J.E. Dickerson, J.A. Dickerson, "Fuzzy network profiling for intrusion detection," In Proc. of the 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS), Atlanta, GA, 2000, pp. 301-306.

[24] M. Ramadas, S.O.B. Tjaden, "Detecting anomalous network traffic with self-organizing maps," In Proc. of the 6th International Symposium on Recent Advances in Intrusion Detection, Pittsburgh, PA, USA, 2003, pp. 36-54.

[25] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 15.

[26] J. R. Quinlan, "Induction of Decision Tree," Machine Learning Vol. 1, pp. 81-106, 1986.

[27] J. R. Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[28] L. Breiman, J. H. Friedman, R. A. Olshen and C.J. Stone, "Classification and Regression Trees," Statistics probability series, Wadsworth, Belmont, 1984.

[29] John Shafer, Rakesh Agarwal, and Manish Mehta, "SPRINT: A Scalable Parallel Classifier for Data Maining," in Proceedings of the VLDB Conference, Bombay, India, September 1996.

[30] The KDD Archive. KDD-99 cup data-set, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Journal of AL-Qadisiyah for computer science and mathematics Vol.10 No.2 Year 2018
ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

نظام كشف التسلل التكيفي باستخدام شجرة القرارات

أسامة عادل رحيم اسراء صالح حسون
كلية علوم الحاسوب وتكنولوجيا المعلومات / قسم الحاسوب / جامعة واسط

المستخلص :

في العقود الأخيرة ، تحول أمن البيانات إلى رؤية جديدة في ابتكار البيانات حيث يتم تقديم كمية من فواصل حماية الكمبيوتر الشخصي إلى انفجار في عدد الحوادث الأمنية. تم استخدام مجموعة متنوعة من نظام اكتشاف التسلسل من أجل حماية أجهزة الكمبيوتر والأنظمة من الاعتداءات الخبيثة القائمة على النظام أو التي تعتمد على النظام من خلال استخدام الاستراتيجيات التقليدية القابلة للقياس إلى أساليب التعدين الجديدة للمعلومات في السنوات الماضية. وعلى أي حال ، فإن الأطر القائمة الحالية لتحديد الهوية التي يمكن الوصول إليها بصورة نقدية تستند إلى علامة غير مجهزة للتعرف على الاعتداءات الغامضة. في هذه الورقة ، نقدم حسابًا تعليميًا آخر لإطار تعريف عدم انتظام النظام القائم على الشذوذ باستخدام حساب شجرة الاختيار الذي يعترف بالاعتداءات من الممارسات العادية ويميز الأنواع المتنوعة من الانقطاعات. يظهر الاختبار على KDD-99 تنظيم قاعدة البيانات اكتشاف تنظيم مجموعة معارض أن الخوارزمية المقترحة أسفرت عن 98.5٪ معدل اكتشاف مقارنة مع تقنيات التنفيذ الأخرى.