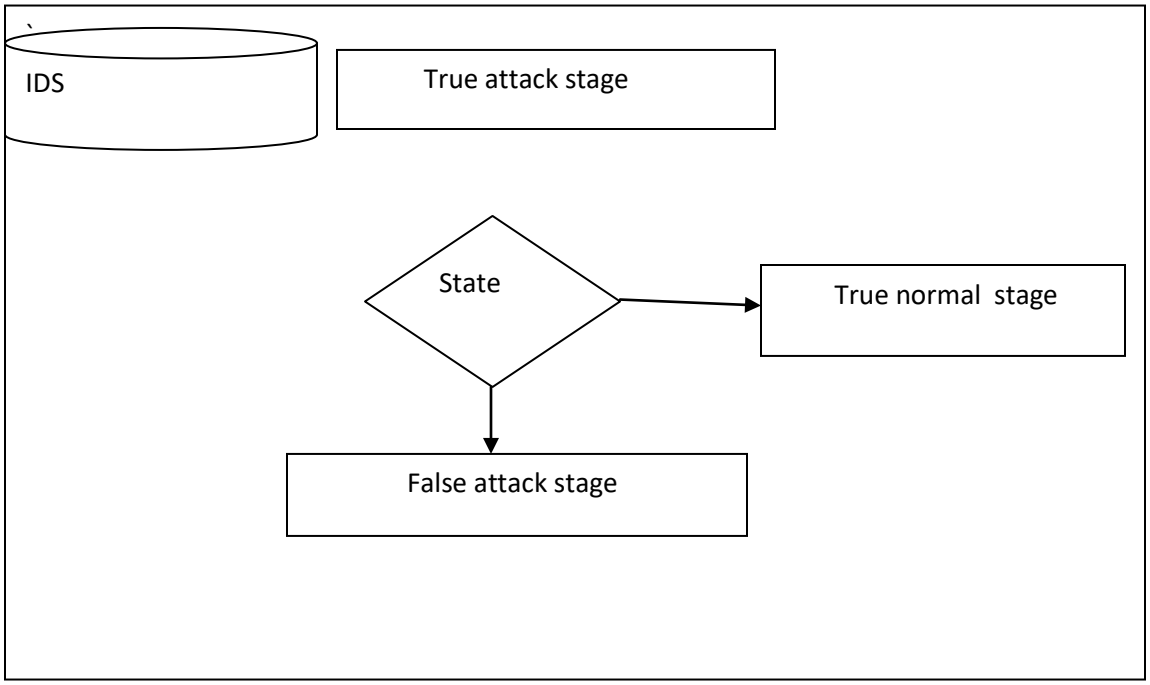


Figure 2.1 Illustrate misuse detection techniques model (Patel, 2008)

In the figure 2.1 given above illustrates the misuse detection model .The model use pre-defined signature pattern defined in its database to detect anomalous packet using pattern matching algorithm. The model above compares normal signature pattern with abnormal define in the IDS dabatase, If a match is found then a response is generated inform of alert or log the incident in the log files.The above model addressed the problems of false negative by updating the model profile (Patel, 2008).

Anomaly based detection techniques

Basically, in anomaly based detection network behavior is learned and the learned network behavior is compared with the incoming network traffic to detect an intrusion (Sandhu *et.al* 2011). In this technique many possibilities are used to detect anomalous behaviors. Data such as Kernel information, system log records .Software Information running information, normal

packet characteristics were collected stored and network throughput change. Therefore, any deviation to this gathered information is recorded as anomalous.The techniques used Metric to measure and detect the deviation of the system behavior , generate and send an alert to the alert modules.

The model development comprises of three different stages namely parameterization, training and detection stage (Ning and Jajodia, 2001).However, the working stile of this technique in contrast to the signature based detection techniques, it defines on some network protocols, it presents a problems in rule setting and produce higher rate of false alarm. The strength of this techniques over signature based engines is, its ability to be able to deter any novel attack whose pattern has deviated from normal traffic pattern (Ning, and Jajodia, 2001).

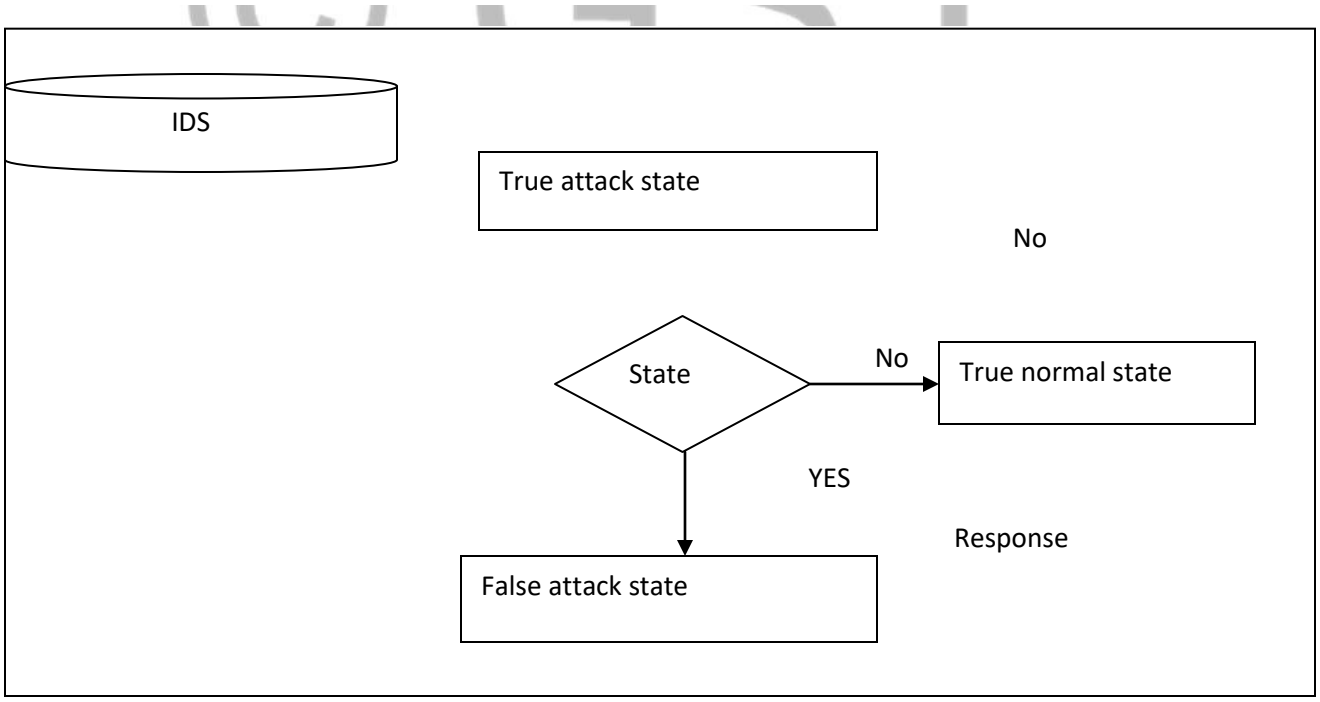


Figure 2.1 Anomalous Detection Techniques Model (Patel,2008)

In the Figure 2.2 given above illustrates the anomalous detection model. The detection is based on the deviation of the model system behavior as described in the section 2.3.5, the incoming data is analyzed and a significant deviations or correlation or similarities are observed and then responses are generated and either log or send to the alerting modules for the system administrator. In addition for the false alarm generated mostly by this model is addressed through profile upgrade and changes made within the system or network traffic behavior observed (Patel, 2008).

Historical background of SNORT

Snort intrusion detection is an open source intrusion detection which is a signature based. Snort tool was originally design as a packet sniffer in 1998 by Marty Roesch which was named APE (Linda Geddes, 2009). Despite the function perform by the APE, Marty Roesch wanted to have a sniffer that can have additional features or that can perform many functions such as ability to function in many different Operating systems platforms. Windows Linux and UNIX are few among the interested operating system platform Marty Roesch wanted to have snort tool working on. Another desire by the Marty Roesch is the ability for the sniffer (APE) to display multiple difference network packets in the unique form. Much advancement on sniffer features come to being including the sniffer ability to not only capture packet but to filter it also. This application is named libpcap. However, In December, 1998, snort become packet storm which has only one thousands and six hundred (1600) lines of code that are compiled in only two files. Marty's uses snort to perform many work such as monitoring his cable modern and debugging his network applications at around January 1999 snort become a fully features signature based detection system. In addition, on December 1999 a new version of snort 1.5 was released, which was used

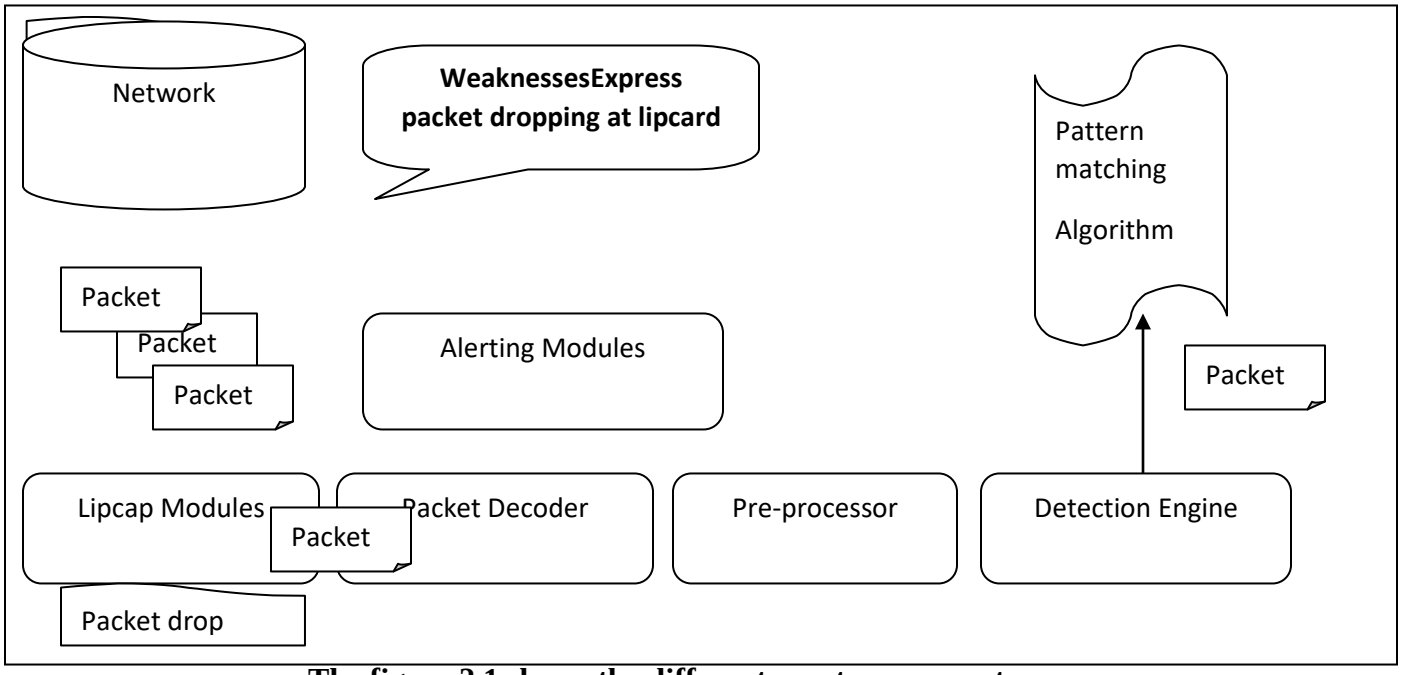
as a light weight intrusion detection system at that time snort used many different plug-in that are being used now. The latest version of snort tool came to being in 2003 snort 2.x.x.x has 75,000 line of code.

Snort Intrusion Detection System

According to Rani and Singh (2010) defined snort as a single threaded network security mechanism that work based on four difference configuration mode, Packet sniffer mode, packet logger mode, detection mode and prevention mode or inline mode. Snort is an open source network intrusion detection system which is configured in a network PC as a network IDS. Also, snort is incorporated in third party solutions, snort tool get wide acceptance by many industries all over the world with millions of a downloads. The detection of malicious traffic by snort is done by using transmission control protocol stack. A deep packet payload inspection is carried out by matching the observed packets and pre-defined snort signature. In a network, snort can be implemented in many different platforms such as Linux, FreeBSD, windows but snort has higher performance when deployed in a Linux platform because of its higher supportability, stability, security and reconfigurable network subsystems.Also, snort performance is optimized by using Berkeley filter (BPF) using BPF only interested network packet are allowed to pass for analyzes by the snort components (Terrence *et. al*, 2010).

Similarly, Snort packet logging performance and its fault alarm produces during packet inspections depends on the accuracy of its configurable components. i.e. snort packet grabbing mechanism (lipcap library). In addition the accuracy of how detection engines algorithms inspect the packet payload to detect malicious pattern gives rise to the snort better performance. Also, these components are separated into different dependent components; the first component is the packet capturing mechanism or modules through which the transmitted network is tossed in to the snort. After the lib-cap accomplished its functions the captured packet in raw form is passed

to the decoder components after the raw packet is being decoded or prepared for pre-processing its send in to the snort next components for pre-processing which is the pre-processor component. The pre-processor perform many task on the received packets before sending it to the detection engine, among the functions are packet examinations, manipulations, and defragmentation and packet classifications. The detection engine which is the main snort components perform a test to check whether the packet is malicious to detect intrusion and the result obtained is send to the final plug-in which is the snort output.



The figure 2.1 shows the different snort components

Pattern Matching Algorithms

Pattern matching algorithm according to (Yaron,2011) is a core component of intrusion detection which is composed of step by step instructions made to match the incoming packet string against the pre-defined signature with the aims of finding the malicious signature. Pattern

matching algorithm is a snort core component that plays a vital role in packet inspection for a malicious identification. The pattern matching algorithms according to (AbuHmed *et. al*, 2012) are classified according nature of the platform where they are deployed, that is software based algorithm or hardware based algorithm and in some cases pattern matching algorithms are usually found to have both characteristic software and hardware based as shown in figure 2.4 below.(Hong,2012) stated in his literature that in some case algorithm categories based on theie mode of operations and this categoration had leaded to the widely known pattern matching techniques types algorithms, single pattern matching and multiple pattern matching algorithms. In single pattern matching techniques, the algorithms scans the entire text and match one pattern at a time while in the multiple pattern matching many patterns are match at a scan. Indeed, the most commonly known types of pattern matching algorithms are either Ah-corasick, boyer-more or Wu-Manber algorithm (Weinsberg *et. al*,2011).Basically, as a result of many drawbacks or limitation of these algorithms researchers aimed at developing new algorithms based on the traditional normal behaviors of the existing algorithm to come up or solve the challenges of their predecessor in order to have good and efficient performance in packet inspection

.

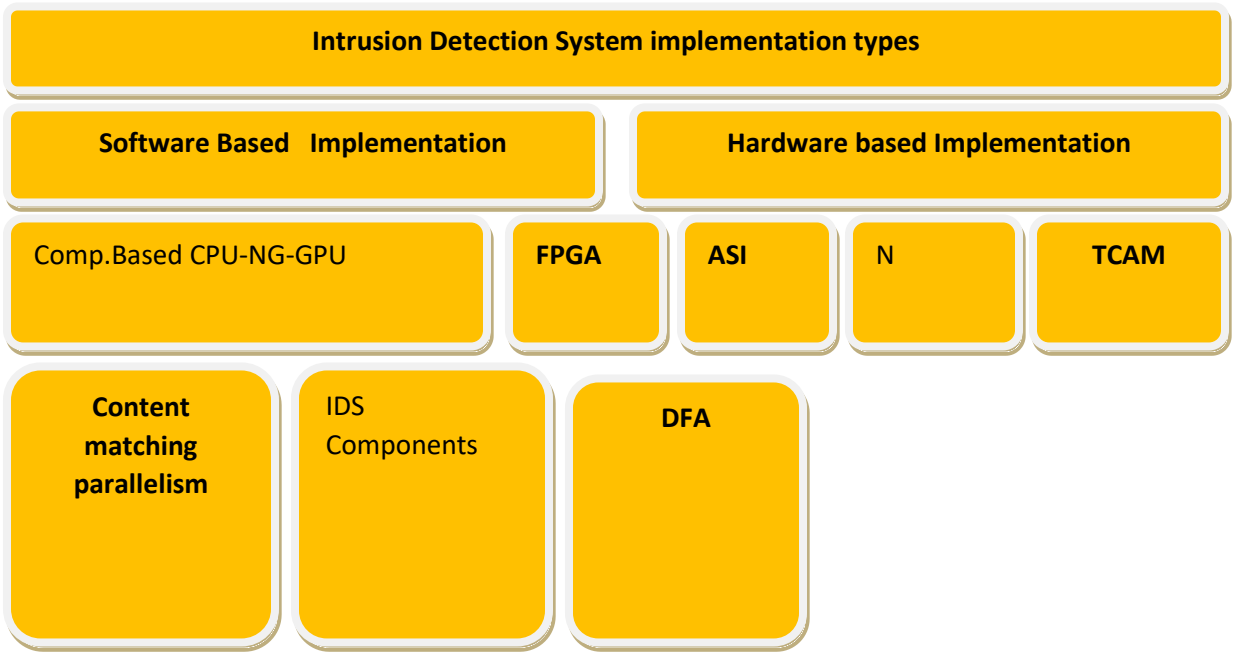


Figure 2.1 hardware and software based IDS implementation

In addition, among the modified or newly build algorithms includes quick search algorithm, hoorspol, dual, hybrid algorithms and many more among which are going to be described in the later section of this paper.

Despite the current enhancement or improvement often did to the snort pattern matching algorithm they are yet to meet the performance requirement needed for snort to function well in a higher traffic network. Many challenges are yet to be solved by the current traditional and modified pattern matching algorithm. Basically, pattern matching algorithms uses four different methods in performing pattern matching which includes left to right comparison, right to left comparison, specific order and un-relevant order.

Aho Corasick Algorithm

Aho-Corasick is a multi patterns matching algorithm, which is usually used for packet string matching in intrusion detection system. This is build often Automata technology it requires more or much time for performing pattern analysis and its considered effective in performing pattern search independent of pattern size. This algorithm contains difference time complication of search, this time is expresses as either 0 (n) or 0 (n x log @) depending on either the automaton is in direct 71 access table. Indeed, for the algorithm pre-processor components to perform its function it requires more time and this can be done up line depending on the application it dealt with. In the traditional Aho carosick algorithm building deterministic finite automaton can be achieved by using a single unit of character. Many problems hinders the performance of Aho Corasick pattern matching algorithm, (Norton,2004). Also based on the algorithm implementation a measurable improvement has been presented in terms of memory requirement reductions as highlighted in (Jonhson and Yao,2002), which had proposed a best possible

memory representation of the sparse state table, but this had contributed to the lost of performance compared to the full matrix presentation. Similarly, this algorithm is considered to be out-dated from being used as a snort algorithm quite long of time. Since addressing the memory requirement and the matching efficiency had became difficult issue to be addressed completely, the algorithm deals with forming of a finite state machine and the input text usually processed using pattern matching machine in a single pass. The example of such is given in:

Figure 2.4 below shows the construction of finite state machine using the four different key words

The state machine is constructed based on the go to function that allowed moving from one state to another.

Wu-Moore algorithm

Wu-Moore pattern matching algorithm is based on the Boyer Moore algorithm which has the capability to perform multiple searches in a single step. The Wu Moore because of its efficiency it allow more than one character to be matched at once, therefore, its pattern shift is made to be small this is because during text scan there is possibility for the algorithm to make match between two adjacent last characters of the patterns and text. Also, the WM algorithm make used of two distinct mechanisms while searching a text for a specific pattern, the two mechanisms includes, filter mechanism that is form based on the hash technology and other mechanism utilized by this algorithms is block character shift mechanism. This mechanism is a borrowed idea from the Boyer Moore bad character shift function the matching process in this algorithm is done by calculating the hash values of the suffix block character which is compared with that of the pattern value.Similarly, the main function of the bad character shift is to ensure the shifting of the windows when match occurs. Operationally, Wu-Moore algorithm (WM) has two main processing stages as done by other traditional pattern matching algorithms. The main stages

include pre-processing stage and pattern searching stage. In this operations are perform by the algorithm which includes setting the size of the matching windows as well as establishment of the three important functions i.e. Shift table, hash table and prefix table. Many matching information are stored in the functions such as shift distance, entry of the link list used for linking the pattern that are of the same or similar mach window in the text also another entry information of link list are stored that specified the group of pattern having similar prefix within the match window. For examples assuming a text donated by the letter *T= t1,…t2…..,tn''*, where letter n indicate the overall length of the text characters and the patter represented by letter P, the pattern set is given by *''P={p1….p2….,Pi}* , and the match window size is given by the letter ''m'' and the block character size is given by an expression *''B''=2''*

Knuth Marith Algorithm

The Knuth-Morris-Pratt (KMP) algorithm is an improved idea of Brute Force algorithm, which uses a shift function based on the notion of the prefixes of the pattern and it is measured as the first linear string matching algorithm. The algorithm is used by core component of intrusion detection system to detect signature pattern match in the incoming network packet with the aims of detecting an intrusions. The Algorithm is a single string matching algorithm unlike Boyer Moore algorithm, this algorithm work in a different way. The main concern of the KMP algorithm is to find occurrences of pattern "P" in string of text "T". However, in a operation by this algorithm two distinct operations or stages are carried out, firstly, each character of the pattern "P" is compared with the pre-defined set of string, if matched is found the operation is continuing, otherwise, the pattern is shift by a position toward right and the above mentioned procedure is repeated again. Indeed, the running time of the Knuth Marith algorithm (KMP) is express mathematically as o (m+n) where n and m represent the length of the pattern and string text respectively (Kumar,

2011).However, Knuth Marith algorithm in dealing with set of string data it adopts two phases, pre-processing phase and searching phase. The Knuth pattern matching algorithm is based on feature matching that is widely adopted in the modern intrusion detection, less time of o(n) is set to be achieved by ignoring the set of characters previously been compared and one important characteristic of Knuth Marith algorithm (KMP) backtracking never occurs. Similarly, Knuth uses prefix function to avoid backtracking (Unused shift of the pattern) and the Knuth Marith (KMP) matcher that returns the number of shifts of the pattern when a successful matched is done. However, work by Baker (2005), Good feature of KMP pattern matching algorithms has made it possible for most of the intrusion detection system to use it for packet pattern matching, the algorithms uses two difference mechanism single comparator and pre-computed transition table which enable reduction of repeated comparison. In addition, in pattern matching, computational efficiency is needed as such; the Knuth Marith algorithm (KMP) is made to be used as a hardware based pattern matching algorithm. The implementation guarantees a higher throughput in parallel hardware architecture.However, the snort performance degradation in intrusion detection is course as result of inefficiency of the detection engine to performs intrusion detection at the rate of the network speed (Soumya,2010) and the efficiency of snort detection engine depends on how accurate and fast its pattern matching algorithm could perform deep packet inspections,(Ibrahim,2011) comparing pre-defined signature pattern and that of the incoming network packet. Therefore, in this review many literatures are studied that have contributed in enhancing snort detection performance based on the platform where the tool (Ids) is configure and the pattern matching algorithm used.

Refference

Abuhmed, T., Mohaisen, A., & Nyang, D. (n.d.). Deep Packet Inspection for IntrusionDetection Systems *A Survey. Information Security*

Alserhani, F., Akhlaq, M., Awan, I. U., Cullen, A. J., & Mellor, J. (n.d.). Snort Performance Evaluation 1. *Performance Evaluation international journal*

Alia, M. A., Hnaif, A. A., Al-anie, H. K., Abu, K., Manasrah, A. M., & Sarwar, M. I. (2011). An ovel header matching a lgorithm for, *3*(4).

Antonatos, S., Anagnostakis, K. G. Ý., Markatos, E. P., Polychronakis, M., & Street, S. (n.d.). Performance Analysis of Content Matching Intrusion Detection Systems

Baker, Z. K., Member, S., & Prasanna, V. K. (2005). Flexible Intrusion Detection *October, 13(10), 1179-1189.*

Bansal, K. (2008). The Knuth-Morris-Pratt algorithm.

Boob, S., & Jadhav, P. (2010). Wireless Intrusion Detection System *International Journal, 5(8), 9-13.*

Brown, D. J., Suckow, B., & Wang, T. (n.d.-a). A Survey of Intrusion Detection Systems Information Sources Analysis Techniques.

Brown, D. J., Suckow, B., & Wang, T. (n.d.-b). A Survey Information Sources Analysis Techniques *international conference*

Dhanalakshmi, Y. (2008). Intrusion Detection Using Data Mining Along Fuzzy Logic and Genetic Algorithms. *Journal of Computer Science, 8(2), 27-32*

Dı, J. (2009). Anomaly-based network intrusion detection : Techniques , systems and *Challenges, 28, 18-28. doi:10.1016/j.cose.2008.08.003.*

Hai-sheng, Q. I. N. (2011). Algorithm Based on Instrusion Detection System, 0-3. *International journal of computer science.*

Idika, N. (2007). A Survey of Malware Detection Techniques. Purdue University, 48. *Citeseer. Retrieved from*

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.4594&rep=rep1&type=pdf

Jyothsna, V., Prasad, V. V. R., & Prasad, K. M. (2011). A Review of Anomaly based.

Dı, J. (2009). Anomaly-based network intrusion detection : Techniques , systems and *challenges, 28, 18-28. doi:10.1016/j.cose.2008.08.003*

Hai-sheng, Q. I. N. (2011). Algorithm Based on Instrusion Detection System, 0-3.

Idika, N. (2007). A Survey of Malware Detection Techniques. Purdue University, 48. Citeseer.Retrieved

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.4594&rep=rep1&type=pdf

Jyothsna, V., Prasad, V. V. R., & Prasad, K. M. (2011). A Review of Anomaly based Intrusion Detection Systems. *International Journal, 28(7), 26-35.*

Kumar, S. (2011). Design and Implementation of IDS Using Snort , Entropy and *Alert Ranking System. Source, (Icsccn), 264-268.*

Mandumula, K. K. (2011). nu t or ris r at t nu t. History.

Nazer, G. M. (2011). Current Intrusion Detection Techniques in Information Technology - A Detailed Analysis. European Journal of Scientific *Research, 65(4), 611-624.*

Norton, M. (2002). Optimizing Pattern Matching for Intrusion Detection. System, 11.

Papadogiannakis, A., Polychronakis, M., & Markatos, E. P. (n.d.). Improving the Accuracy of Network Intrusion Detection Systems Under Load Using *Selective Packet s Discarding*.

Rajasekhar, K., Babu, B. S., Prasanna, P. L., Lavanya, D. R., & Krishna, T. V. Raju, (2011). An Overview of Intrusion Detection System Strategies and Issues. *Network, 8491, 127-131*.

B., & Srinivas, B. (2012a). Network Intrusion Detection System Using KMP Pattern MatchingAlgorithm. *Computer Science and Telecommunications,3(1),14.*

Raju, B., & Srinivas, B. (2012b). Network Intrusion Detection System Using KMP PatternMatchingAlgorithm.*ComputerScience and Telecommunications,*

3(1), 1-4.

Re, K.-morris-pratt. (n.d.). Pattern Matching.

Roozbahani, A. R. (2009). Service Oriented Approach to Improve the Power of *Snorts. doi:10.1109/ICCEE.2009.270.*

Salah, K. Ã., & Kahtani, A. (2010). Journal of Network and Computer Applications Performance evaluation comparison of Snort NIDS under Linux and Windows Server. Journal of Network and Computer Applications, 33(1), 6-15 *Elsevier doi:10.1016/j.jnca.2009.07.005.*

Sandhu, U. A., Haider, S., Naseer, S., & Ateeb, O. U. (2011a). A Survey of Intrusion Detection & Prevention Techniques. *Management, 16, 66-71*.

Security, C., & Monitoring, T. (2011). Importance of Intrusion Detection System *(IDS). International Journal, 2(1), 1-4.*

Sedjelmaci, H., & Feham, M. (2011). novel h ybrid i ntrusion d *etection system. Network Security, 3(4), 1-14.*

Sheik, S. S., Aggarwal, S. K., Poddar, A., Balakrishnan, N., & Sekar, K. (2004). A FAST Pattern Matching Algorithm, *1251-1256*

Singhrova, A. (2011). A Host Based Intrusion Detection System for DDoS Attack in. *wlan. Engineering, 433-438.*

Singla, N., & Garg, D. (2012). String Matching Algorithms and their Applicability in. various Applications. *Soft Computing, (6), 218-222*

Snort, D. (n.d.). Dissecting Snort. Network. Tekniska, K. (n.d.). Intrusion Detection *Systems.*

Verwoerd, T., & Hunt, R. (2002). Intrusion detection techniques and approaches. *Computer Communications, 25, 1356-1365*

Weinsberg, Y., & Dolev, D. (n.d.). High Performance String Matching Algorithm for a Network Intrusion Prevention System (NIPS)s. *P And T.*

Wu, S., & Manber, U. (1994). A fast algorithm for multi-pattern searching. Zeng, B., Yao, L., & Chen, Z. (2010). A Network Intrusion Detection System with the *Snooping Agents. Source, (Iccasm), 232-236.*

Wu, S., & Manber, U. (1994). A fast algorithm for multi-pattern searching, 1-11.

Xian-feng, H., & Yu-bao, Y. (2010). ((ri g ht (t [J, 310-313.