Journal of Advances in Management, Engineering and Science (JAMES)
Volume 01, Issue 01, March 2023

MULTILEVEL SECURITY MODEL IN INTRUSION DETECTION AND PREVENTION SYSTEMS WITH DIFFERENT CRYPTOGRAPHY ALGORITHMS

P. Srinivasan
Professor, Department of Artificial Intelligence and Data Science
Muthayammal Engineering College (Autonomous), Tamilnadu
Email: salemsrini4u@gmail.com

K.Vengatesan
Professor, Department of Computer Science and Engineering
Sanjivani College of Engineering, Maharastra
Email: kvengatesancomp@sanjivanil.org.in

Abstract: A variety of commercial operations including banking, e-commerce, defence-related affairs, and social networking have emerged as a result of the growth of the Internet and are increasingly heavily reliant on network security. A crucial piece of technology for many different applications is system and network technology. To lessen the susceptibility of the computer to the network, security management, intrusion detection systems, authentication procedures, and firewalls may be utilised. In corporate organisations all throughout the globe, combinations of several of these instruments are employed. The most effective firewalls, antivirus software, intrusion detection systems, and other security solutions are expensive. These products are affordable for large businesses, but what about smaller ones? They do not have the same economic situation or need for the most advanced and costly computer security equipment. The well-known security technology known as an intrusion detection system (IDS) is used by businesses to stop data loss and damage. For businesses and organisations that do not have the same financial resources as bigger businesses and governmental entities, an open source intrusion detection system is a useful solution.

Keywords: *Diffie Helman, SHA algorithm, Digital signature, Discrete Logarithm.*

1. INTRODUCTION

Intrusion is the act of directly or indirectly accessing computer resources outside one's rights or without authorization. An attack is any successful effort to take advantage of configuration in system or application software, protocol, data, or services. It may cause the system to lag, applications to stop working, a denial of service attack, or a system crash. Attacks may come from both insiders (LAN users) and outsiders (from internet). Servers, networking hardware, hosts, and applications are all targets for attacks. Security may consequently be expanded to nearly any hardware or software engaged in communication up until the goal is reached, rather than being restricted to hosts or networks[1]. The destination system has the same security requirements. In addition to information encryption at several levels, intrusion detection systems may be employed to offer the necessary security at the host, network, or application level.

Security Hierarchy

The figure 1. shows that how the field of security protects the general assets like

Fig.1 Hierarchy of the security specializations

hardware, software and information resources and its various subfields. Three modes of security can be applied to any situation and three D's of security are:

- Defence
- Deterrence
- Detection

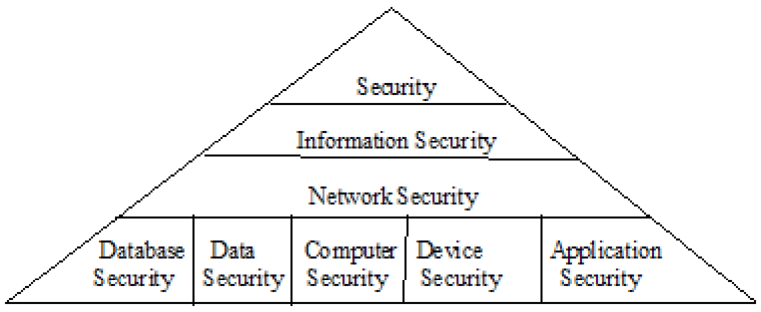

Figure.1 Hierarchy of the security specializations

Security against Networking Threats/Challenges of Security: Security of systems, information, resources, policies, configuration etc. are challenged by attacks. Various types of security attacks, vulnerability types are listed below

Security Attacks: Security attacks are further classified as

Passive Attacks: Passive attacks include release of message contents, traffic Analysis, emphasis on prevention rather than detection by encryption

Active Attacks: Active attacks include Masquerade, Replay, Denial of Service, Splicing Attacks, Timing Attacks, Spoofing and hijacking, Hacker/ Cracker Attacks, Session Hijacks, LoPht Crack, Network denial of service, TCP/IP spoofing, SYN flooding, Modification of message attacks. Ping Attacks, TCP sequence Guessing, IP/UDP Fragmentation, ICMP flooding (Smurf), DNS cache poisoning, Viruses and Worms, Evasion Attacks.

Attack Terminology

Historically, tactics including viruses, worms, buffer-overflow vulnerabilities, and denial of service assaults have been used to attack computers. On the other hand, network assaults often target machines that make use of a network in some manner[2]. A network might either be the mechanism of attack (such as a worm) or be utilised to transmit the assault (such as Distributed Denial of Service attack). Network attacks are often a subset of computer assaults. There are, however, a number of network assaults that target the network rather than the computers it is connected to. Instead than attacking a specific machine, flooding a network with packets slows down the whole network. Although the assault may be started on a computer, the target and the method of attack are both network-related. Numerous taxonomies and classifications of computer system attacks are available in the literature[3].

Attacks are categorised according to their effect on the computer system in Snort, the most popular open source network intrusion prevention and detection system. The assaults with the most dire consequences are given top consideration[4]. There are three different priority levels: high, medium, and low. Attacks with a high degree of priority include those that aim to obtain administrator privileges, use network "Trojans," or target online applications. Denial of service (DoS) assaults, an unusual protocol or event, possibly problematic traffic, attempts to log in using a suspicious user, etc. are examples of medium priority attacks. The ICMP event, a network scan, a generic protocol instruction, and other low-priority assaults[5].

Commonly Encountered Attacks

Following discussion gives an extensive view of the commonly encountered attacks.

Viruses are self-replicating programmes that spread via infected files. The majority of the time, they attach themselves to a file, causing them to execute when the file is opened[6].

The following list of common viral types includes:

File malware: By embedding themselves into a file, file infector viruses infect files on the victim's computer. The file is often an executable one, like a Windows.EXE or.COM. The virus also runs when the infected file is opened.

Infectors of the system and boot record: Up until the middle of the 1990s, system and boot record infectors were the most prevalent virus kind. The Master Boot Record (MBR) on hard drives and the DOS boot record on floppy discs, as well as other system components of a computer, are both often infected by these viruses. The virus may execute itself every time the machine boots up by inserting itself into the boot records[7].

Macro viruses: Malicious macros for well-known applications like Microsoft Word are known as macro viruses. They could, for instance, change the wording of a document or remove information from it. The virus often spreads via the contaminated files. If a person opens a document that is contaminated, the virus could be installed and infect any other papers it comes across. The macro virus is often attached as an innocent-looking file in an attempt to fool the user into being infected. Melissa is the most well-known macrovirus[8]. By sending a victim an email that seemed to be from a friend, the virus propagated. The virus would spread to the first 50 contacts in the victim's address book after opening the Microsoft Word attachment if the victim used the Microsoft Outlook 97 or 98 email client. Melissa severely destroyed email networks as a result of the torrent of emails that the virus spread.

2. LITERATURE REVIEW

Key distribution between two parties and key management are highly essential areas of consideration for the work to be done, especially in light of the current security needs of the expanding sectors of banking, finance, and e-commerce operations. The cypher creation time would be excessively long if the key generation, authentication, and encryption techniques were employed individually with the provided message as opposed to doing everything at once. In order to decrease the amount of cypher creation required for transmission, it was suggested to integrate the Key generation process with the authentication protocol [9].

Open source intrusion detection systems like Snort, which have built-in flexibility in terms of their operating modes and rule modification, nevertheless need further effort. To improve the security function of this lightweight open source IDS system, we might attempt to cooperate with Snort and investigate its rules. It might be set up to operate in a mode that configures network traffic more quickly while using less memory and processing power. The pattern matching algorithm of Snort may be improved by working with additional automata [10].

In order to increase network security, XinYuZhang et al. suggested incorporating a variety of security measures. While IDSs and firewalls may successfully enforce network security, IPS is a solution that appropriately combines IDS and firewall technology. The authors address two NIPS aspects, namely inline and traffic isolation, and provide a distributed IPS Design based on SNMP[11]. In order to manage associated rules in business networking systems, this study aims to offer a framework. For security reasons, authors suggested using a framework called ARM (attack response matrix) to combine intrusion analysis with traffic enforcement. The mapping from incursion categories to traffic enforcement measures is described in ARM[12].

In an effort to solve the issue of intrusion prevention by fusing IDS with a stateful load balancer, Joe Stevens et al. developed the concept and implementation of secure direct. Secure Direct is a real-time load balancer that can tell the difference between traffic from "legitimate" clients and traffic from malicious users. On the basis of this, traffic coming from a known attacker is sent to a different server, where harm might be reduced. The benefit of this system is that it prevents intrusions in a way that is transparent to the attacker and enables the administrator to respond appropriately by allowing for the monitoring and examination of assaults [13].

To help the information assurance community [14] have offered a novel strategy that uses an open source testing environment and methodology. The idea is that using actual traffic from the site where the algorithm is to be implemented, either live or recorded, is the best approach to assess any intrusion detection system.

The system can scan a far greater volume of data than software-based techniques since the authors have taken use of the parallelism provided by networks. By hashing several windows of data simultaneously to on-chip memory, throughput was increased. These memories may all be refreshed simultaneously. The hash followed by a counter update in software will need numerous instructions to be performed in order. Low false positive rates are guaranteed by the connection between timeout and threshold, together with the SRAM analyzer. To find irregularities in network traffic, current network monitoring solutions depend on the system administrator's gut instinct. An essential aspect of system security is preventing the execution of illegal software on a specific machine. Although a programme starts off with MAC verification at the beginning of execution, this is the main issue. It divides up a binary programme into blocks of instructions. A keyed MAC is appended as the block's footer and used to sign each block.

SGS (Security Gateway System) and CPCS make up the secure networking architecture (Central Policy Control Server). SGS serves as an IDS at the network's edge while CPCS serves as a higher-level server. SGS creates an IDMEF compliant with alert data and transmits it to CPCS. It creates an alert object for use in correlation analysis after parsing IDMEF alert data. While CPCS can see a large portion of the network, SGS can only observe its immediate region.

Active Mapping, a simple method developed by Umesh Shankar and Varun Panson, removes TCP/IP-based ambiguity in an NIDS analysis with a negligible runtime cost. The semantic and performance issues associated with traffic normalizations, which change traffic streams to eliminate ambiguities, are avoided by active mapping. In order to conduct experiments using the Active Mapping produced profile database, the authors updated an NIDS and created a prototype implementation of the technique [15]. How to choose the optimum model for an intrusion detection system (IDS). The IDS models are evaluated using the relative entropy density divergence. The comparison to the various IDS models based on the original audit data is carried out via the probability distribution dependency analysis of the data.

3. INTEGRATED DIFFIE-HELLMAN DIGITAL SIGNATURE (IDHDS) SECURITYALGORITHM

The algorithm's goal is to make it possible for two users to safely exchange a key that will later be used to encrypt communications. The algorithm is only capable of exchanging secret values. The success of the Diffie-Hellman method relies on how hard it is to compute discrete logarithms.

Discrete Logrithm Problem

In a nutshell, the discrete logarithm may be defined as follows. A prime number modulo p that generates all the integers from 1 to p-1 is said to have a primitive root of p. The numbers a mod p, aP2P mod p, aP(p-1)P mod p, etc. are unique and include the integers from 1 through p-1 in some permutation if an is a primitive root of the prime number p. We may discover a single exponent I such that, given an integer b and a primitive root of a prime number p, b aPiP(mod p), where 0= i= (p-1). For the base a mod p, the exponent I is known as the discrete logarithm of b. This value is written as dlogR a, pR(b)[5]. For this technique, there are two publicly available numbers: an integer that is a primitive root of q and the prime number q. Let's say that users A and B want to trade keys. User A chooses an arbitrary number, XRAR, and calculates YRAR as aPXAPmod q. A similar process is used by user B, who independently chooses a random number XRBR and calculates YRBR = aPX BPmod q. The Y value is made publicly visible to the opposite side while each side retains the X value secret. The key is calculated by users A and B as K = (YRBR)PXAPR Rmod q and K = (YRAR)PXBPmod q, respectively. These two computations yield the same outcomes:

K =(YRBR)PXAP mod q
= (aPXBPmod q)P XAP mod q
= (a PXBP) PXAP mod q
= (a)PXBXAP mod q
=(a PXAP)P XBP mod q
=(a PXAP mod q)P XBP mod q
= (YRAR)PXBPmod q

by the rules of modular arithmetic The result is that the two sides have exchanged a secret value. Furthermore, because XRAR and XRBR are private, an adversary only has the following ingredients to work with: q, a, YRAR, and YRBR . Thus, the adversary is forced to take a discrete logarithm to determine the key. For example, to determine the private key of user B, an adversary must compute XRBR = dlogRa,qR(YRBR) The adversary can then calculate the key K in the same manner as user B calculates it. Global Public Elements

q prime number

α α < q and α a primitive root of q

User A Key Generation

XRAR < q

Select private XRA

Calculate public YRA

YRAR = α PXAP mod q

Man-in-the-Middle Attack:

Suppose Alice and Bob wish to exchange keys, and Darth is the adversary. The attack proceeds as follows.

1. Darth prepares for the attack by generating two random private keys XRD1R and XRD2R and then computing the corresponding public keys YRD1R and YRD2R .
2. Alice transmits YRAR to Bob.
3. Darth intercepts YRAR and transmits YRD1R to Bob. Darth also calculates K2=(YRAR)PXD2P mod q .
4. Bob receives YRD1R and calculates K1 = (YRD1R)PXBPmod q .
5. Bob transmits YRBR to Alice.
6. Darth intercepts YRBR and transmits YRD2R to Alice. Darth calculates K1=(YRBR)PXD1P mod q
7. Alice receives YRD2R and calculates K2 = (YRD2R)PXAPmod q .

Bob and Alice mistakenly believe that they share a secret key at this moment, but in reality, Bob and Darth share secret key K1 and Alice and Darth share secret key K2. The following compromises are made to Bob and Alice's ability to communicate in the future.

1. Alice transmits the following ciphertext M: E(K2, M).
2. Darth decrypts the communication after intercepting it in order to get M.
3. When M' is any message, Darth sends Bob E(K1,M) or E(K1,M'). In the first instance, Darth only wants to listen in on the conversation without interfering with it. In the second instance, Darth seeks to alter the communication with Bob. Due to the fact that it does not

authenticate the participants, the key exchange protocol is susceptible to such an attack. Public-key certificates and digital signatures may be used to address this problem.

Digital Signature Algorithm (DSA):

DSA requires the following three Global Public Key Components:
When L is a multiple of 64, such as between 512 and 1024 bits in increments of 64 bits, then p is a prime integer where 2PL-1P, p, and L are all multiples of 64.
Q is the prime divisor of (p-1), where 2P159P q 2P260P, or 160 bits in length.
Where h is any integer with 1 h (p-1) such that hP, g = hP(p-1)/qP modp (p-1)
P modp > 1
Private Key for the User x = Random or pseudo-random iteger with 0 x q
Public Key for the User y = gPx Pmod p
Secret Number for each user's message
With 0 k q, k is a pseudo-random or random integer.
A user must compute two values, "r" and "s," that are functions of the public key parts (p,q,g), the user's private key (x), the message's hash code H(M), and an extra integer K that must be created randomly or pseudorandomly and be distinct for each signing.

3.2 Secure Hashing Algorithms

An authenticator, signature, or message authentication code (MAC) is delivered with the message and serves as the electronic equivalent of a signature on the message. The goals of message authentication are to safeguard a communication's integrity, confirm the sender's identity, and prevent origin from being denied (dispute resolution). The message may be any size, but the MAC is usually a predetermined value, therefore using a hash function to compress the message to the appropriate size is necessary. The message and a (public or private) key that is only known to the sender and recipient are used in some process to construct the MAC. In the event that the authentication method is unable to do this, replay problems with messages and MAC must be taken into account. This requires a message sequence number, timestamp, or predetermined random values.
# Private-key authentication Ciphers: Since only the sender or the receiver may have produced the session key being used to encrypt the message, the communication may also be authenticated. Any interference would taint the message (if it has enough redundancy to

detect change), but because it is impossible to establish who sent the message, non-repudiation is not ensured. The typical ways of a block cypher may also be used for message authentication. If you sometimes don't want to transmit encrypted communications, you may send the last block using the CBC or CFB modes since it depends on all the other bits of the message. Since this approach allows input of any length and creates a fixed output, often using a constant known IV, no hash function is necessary. The method used in Australian EFT standards AS8205 is this one. The main drawback is the short size of the generated MAC since 64 bits is possibly too few.
# Hashing Functions: Hashing functions are used to reduce an arbitrary-length message to a predetermined size, generally in preparation for a digital signature algorithm's later signature. The following characteristics a good cryptographic hash function h should have are:

SHA (Secure Hash Algorithms) is a 160-bit hashing algorithm created by NIST. The message is padded to a length that is a multiple of 512 bits, and the 5-word (160-bit) buffer (A,B,C,D,E) is initialised to (67452301, efcdab89,98badcfe,10325476,c3d2e1f0). The message is processed in 16-word (512-bit) chunks, with each chunk and buffer undergoing four rounds of 20-bit operations. The output hash value is the value of the final buffer. Although SHA and MD5 have many similarities in their designs, they also have some variances.

SHA-256 SHA-1 generates a hash value of 160 bits. SHA-384 algorithms. Three additional variants of SHA, with hash value lengths of 256, 384, and 512 bits, were established in 2002 by NIST in a revised version of the standard called FIPS 180-2. These versions are referred to as SHA-256, SHA-384, and SHA-512, respectively. These hashing methods are referred to as SHA-2 as a whole.

SHA-512 Logic: The technique generates a 512-bit message digest from an input message with a maximum length of less than 2128 bits. The input is handled in chunks of 1024 bits. Message Digest Algorithms, number On 32-bit computers, the MD5 algorithm is designed to be rather quick. The MD5 technique may also be programmed relatively compactly since it doesn't need any substantial substitution tables. The MD4 message-digest method is expanded upon by the MD5 algorithm. Although MD5 is a little slower than MD4, its architecture is more "conservative."

Table 3.1 Computed key & 'r' component of the D S with IDHDS

Prime Number	h value	'r' Component	Computed key
29	23	9	4
37	23	11	33
43	37	7	27
53	47	1	43
103	91	18	66
293	219	132	61
337	331	79	273
557	551	113	471
997	991	248	963
65537	65531	21949	12365

It indicates that when the prime number is raised, the h value rises while the r component sometimes exhibits a value decline and the calculated key rises in line with this. The graph 5.1 demonstrates that the calculated key increases from 50% to 67% when the prime number increases in the range of 37% to 43%.

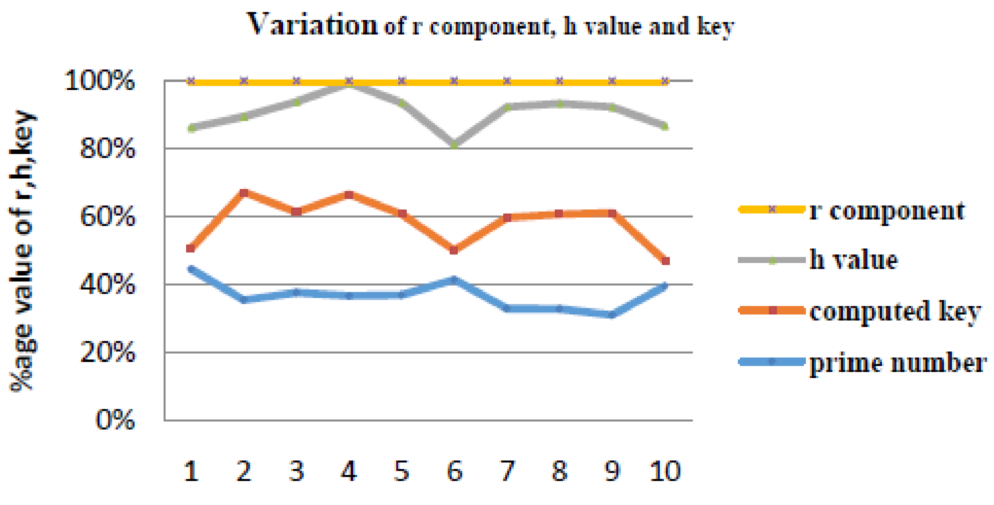

Figure 2. Computed keys

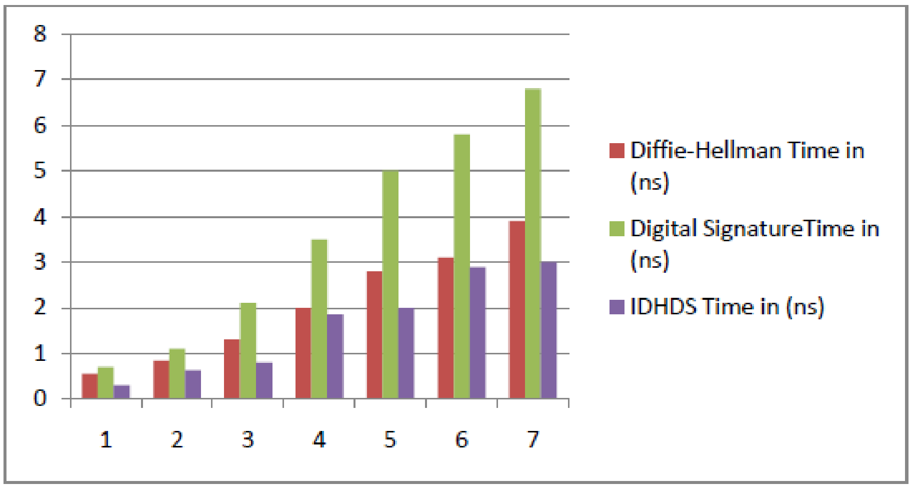

Figure 3. DH<DS<IDHS comparison

The comparative execution times for 30 rounds of the proposed IDHDS algorithm, D-H, and DS, are shown in Graph 5.4. The suggested approach takes substantially less time to execute than the other two techniques.

It is suggested that the steps for authenticity be transmission, compression, digital signature, and encryption. Various hash computing techniques, including MD5 and SHA, are used to compute the s component of digital signatures. The results demonstrate different r and s values for digital signatures created using hashing methods. IDHDS executes faster than Diffie Hellman and digital signatures. The suggested security paradigm incorporates encryption with smart snort for quick pattern matching and IDHDS for digital signature creation and verification.

4. CONCLUSION

Working with tools and approaches to accomplish the security goals of authentication, non-repudiation, access control, confidentiality, and availability were the study's aims. Diffie-Hellman cryptographic key exchange algorithm and digital signature to verify key exchange. The Integrated Diffie-Hellman Digital Signature algorithm (IDHDS), which combines the Diffie-Hellman and Digital Signature techniques, will be developed. Its security feature will be evaluated by computing the complexity of the two methods. Secure hashing algorithms (such as MD5 and SHA) will be used to guarantee the security of the data. To authenticate the key exchange, the Diffie-Hellman cryptographic key exchange mechanism is used with a digital signature. We developed the Integrated Diffie-Hellman-Digital Signature Algorithm (IDHDSA), which combines the Diffie-Hellman and digital signature algorithms, and evaluated their complexity to determine which the best method for assessing the security feature was. Used secure hashing algorithms like MD5 and SHA to assure the security of the data.

REFERENCES

1. XinYuZhang, Chengzhang Li and Wenbin Zheng, ”Intrusion Prevention System Design” IEEE International Conference on Computer And Information Technology, 2004, pp. 386-390 (2004).
2. Joe Stevens and Shadan Saniepour, “Secure Direct:Proactive Security through Content-Based Traffic Control” IEEE International Conference on Advanced Information Networking and applications (AINA’ 03) (2003).
3. Nicholas Athanasiades, Randel Abler, John Levine, Henry Owen and George Riely, “Intrusion Detection Testing and Benchmarking Methodologies” IEEE International Information Assurance Workshop, pp. 1-10 (2003).
4. Bharath Madusudan & JohnLockwood “Design of a System for Real-Time Worm Detection”, IEEE Symposium on High Performance Interconnects, pp. 77-83 (2004)
5. Milenko Drini and Darko Kirovski , “A Hardware-Software Platform for Intrusion Prevention”, IEEE International Symposium on Microarchitecture (MICRO- 37’04), pp. 238-242 (2004).
6. Sang-Kil Park,Ki-Young Kim, Jong-Soo Jang and Bong-Nam Noh “Supporting Interoperability to Heterogeneous IDS in Secure Networking Framework” ICANN, Vol. 2, pp. 844-848 (2003).
7. Umesh Shankar, Varun Panson, “Active Mapping : Resisting NIDS Evasion without alerting Traffic”, IEEE Symposium on Security and Privacy, pp. 44 (2003).
8. C.F.Jia, D.Q.Chen, K.Lin. “The application of the relative entropy density divergence in intrusion detection models”, International Conference on Computer Science and Software Engineering. Wuhan, pp. 951-954 (2008) .

9. Faeiz Alserhani, Monis Akhlaq, Irfan U. Awan, John Mellor, Andrea J. Cullen, Pravin Mirchandani, “Evaluating Intrusion Detection Systems in High Speed Networks”, Fifth International Conference on Information Assurance and Security, Vol. 2, pp. 454-459 (2009) .

10. Liang Guangmin, “Modelling Unknown Web Attacks in Network Anomaly Detection”, IEEE 3PrdP International Conference on Convergence And Hybid IT, pp. 101-109. (2008).

11. Zhou Zhimin, “The Study on Network Intrusion Detection System of Snort”, International Conference on Networking and Digital Society, Vol. 1, pp. 194-196 (2010)

12. Chintan C. Kacha, ”Improved Snort intrusion detection system Using Modified Pattern Matching Technique” IJTEAE, Volume 3, Issue 7 (2013).

13. Tongaonkar A, ”Fast Packet Classification for Snort by Native Compilation of Rules” Large Installation System Administration Conference, pp.159-165 (2008) .

14. Jiqiang Zhai,” Network Intrusion Prevention System Based on Snort” IEEE 6PthP International Forum on Strategic Technology Research (2011).

15. Rewagad, Parshant, Yogita “Use of Digital Signature with Diffie-Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing” IEEE International Conference on Communication Systems and Network Technologies, pp. 437-441 (2013).