A Stable worm - apprehension method for Manet

Prof. Shruti Tiwari[1], Prof. Shreyanshi Patel[2]
[1,2]*Assistant Professor, Department of CT, Priyadarshini College of Engineering, Nagpur*

Abstract **- In few years, mobiles uses like basic need of people in the world. Everyone wants to put up linked with one another. So, the new trend of migration of wired to wireless network application is borned. The wireless network gives mobility & scalability. A mobile ad hoc network (MANET) is an automating framework less network of mobile devices associated by wireless. Ad hoc is Latin and means "for this purpose". Mobile Ad hoc network (MANET) is mostly used in wireless network. It is wireless unfix network infrastructure. In the MANET each one node is sender & receiver to communicate within range. MANET is open & wide infrastructure of network. This is become attraction of attackers. To prevent these attacks intrusion detection mechanism introduced. For using, this mechanism, we have strong faith on its address has potential security issues. In this paper, we propose and implement a new intrusion-detection system designed for MANET named as Enhanced Adaptive Acknowledgment (EAACK). Correlate to extent approaches, EAACK shows higher hostile- behavior-detection rates in specific occurrence while does not greatly affect the network performances.**

Index Terms **- Genetic algorithm (GA), Digital signature, digital signature algorithm (DSA), Enhanced Adaptive Acknowledgment (AACK) (EAACK), Mobile Ad hoc Network (MANET).**

1.INTRODUCTION

When you operate a network you may want to know the status of each, you may hosts on your network, the traffic rate flowing from and into each hosts, and so on. In abbreviated, you may want is going on the network so you can make determination on who may want to do. TCP/IP suites provide a protocol called SNMP (Simple Network Management Protocol) to serve with this purpose.

You can gather network information set host parameters remotely using SNMP. The SNMP works basically as follows: a host acting as the SNMP Manager sends a command to another host acting as an SNMP Agent. The Agent then returns the appropriate responds to the Manager.

With SNMP, you can get data such as:

1. a hosts' uptime
2. a hosts' system description
3. a hosts' available memory
4. the amount traffic flowing in and out of a network interface
5. a host process; i.e. you know whether the web server is up and running

This kind of data helps you to manage your network.

EAACK is Intruder Detection in MANET. It is a *"Network monitoring In Manet"*. It refers to the practice of overseeing the operation of a computer network using specialized management software tools. Network monitoring in MANET are used to ensure availability and overall performance of computers (hosts) and network services. These systems are typically employed on larger scale corporate and university IT networks.

It is capable of detecting and reporting failures of devices or connections. It normally measures the processor (CPU) utilization of hosts, the network bandwidth utilization of links, and other aspects of operation. It will often send messages (sometimes called *watchdog* messages) over the network to each host to verify it is responsive to requests. When failures, unacceptably slow response, or other unexpected behavior is detected, these systems send additional messages called *alerts* to designated locations (such as a management server, an email address, or a phone number) to notify system administrators.

It is the latest in employee network monitoring software. This program allows you to monitor and control all user activity on your network in real time from your own workstation this program makes it easy to see what users are doing and whether or not they are wasting time. Are your employees wasting time online or leaking sensitive data? Do your employees visit inappropriate websites?

Employers seeking to regain discipline in their employees' Internet activity commonly use Intruder

Detection Server. Many companies are faced with employees who waste time online while a supervisor has their back turned. Finding out the TRUTH about what they are doing and correcting an employee's bad behavior couldn't be easier!

View the screen in real time, monitor activity, block web sites or applications and do many other useful tasks. Works with any Windows based network, including wireless networks!

Each device in a MANET is free to move independently in any direction and will hence change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger Internet. MANETs are a kind of Wireless ad hoc network that usually has a routable networking environment on top of a Link Layer ad hoc network.

The growth of laptops and 802.11/Wi-Fi wireless networking have made MANETs a popular research topic since the mid-1990s. Many academic papers evaluate protocols and their abilities, assuming varying degrees of mobility within a bounded space, usually with all nodes within a few hops of each other. Different protocols are then evaluated based on measures such as the packet drop rate, the overhead introduced by the routing protocol, end-to-end packet delays, network throughput etc.

MANET into two types of networks, namely, single-hop and multi-hop. In a single-hop network, all nodes within the same radio range communicate directly with each other. On the other hand, in a multi-hop network, nodes rely on other intermediate nodes to transmit if the destination node is out of their radio range.

MANETS can be used for facilitating the collection of sensor data for data mining for a variety of applications such as air pollution monitoring and different types of architectures can be used for such applications. It should be noted that a key characteristic of such applications is that nearby sensor nodes monitoring an environmental feature typically register similar values. This kind of data redundancy due to the spatial correlation between sensor observations inspires the techniques for in-network data aggregation and mining. By measuring the spatial correlation between data sampled by different sensors, a wide class of specialized algorithms can be developed to develop more efficient spatial data mining algorithms as well as more efficient routing strategies. Also, researchers have developed performance models for MANET by applying Queuing Theory.

2.WHAT IS INTRUSION DETECTION SYSTEM?

An intrusion detection system (IDS) monitors network traffic and monitors for suspicious activity and alerts the system or network administrator. In some cases the IDS may also respond to anomalous or malicious traffic by taking action such as blocking the user or source IP address from accessing the network.

IDS come in a variety of “flavors” and approach the goal of detecting suspicious traffic in different ways. There is network based (NIDS) and host based (HIDS) intrusion detection systems. There are IDS that detect based on looking for specific signatures of known threats- similar to the way anti-virus software typically detects and protects against malware- and there are IDS that detect based on comparing traffic patterns against a baseline and looking for anomalies. There are IDS that simply monitor and alert and there are IDS that perform an action or actions in response to a detected threat. We’ll cover each of these briefly.

NIDS

Network Intrusion Detection Systems are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. Ideally you would scan all inbound and outbound traffic, however doing so might create a bottleneck that would impair the overall speed of the network.

HIDS

Host Intrusion Detection Systems are run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator of suspicious activity is detected.

Signature Based

A signature based IDS will monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. This is similar to the way most anti-virus software

detects malware. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to your IDS. During that lag time your IDS would be unable to detect the new threat.

Anomaly Based

An IDS which is anomaly based will monitor network traffic and compare it against an established baseline. The baseline will identify what is "normal" for that network- what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect to each other- and alert the administrator or user when traffic is detected which is anomalous, or significantly different, than the baseline.

Passive IDS

A passive IDS simply detects and alerts. When suspicious or malicious traffic is detected an alert is generated and sent to the administrator or user and it is up to them to take action to block the activity or respond in some way.

Reactive IDS

A reactive IDS will not only detect suspicious or malicious traffic and alert the administrator, but will take pre-defined proactive actions to respond to the threat. Typically this means blocking any further network traffic from the source IP address or user.

One of the most well known and widely used intrusion detection systems is the open source, freely available Snort. It is available for a number of platforms and operating systems including both Linux and Windows. Snort has a large and loyal following and there are many resources available on the Internet where you can acquire signatures to implement to detect the latest threats. For other freeware intrusion detection applications you can visit Free Intrusion Detection Software.

There is a fine line between a firewall and an IDS. There is also technology called IPS – Intrusion Prevention System. An IPS is essentially a firewall which combines network-level and application-level filtering with a reactive IDS to pro-actively protect the network. It seems that as time goes on firewalls, IDS and IPS take on more attributes from each other and blur the line even more.

Essentially, your firewall is your first line of perimeter defense. Best practices recommend that your firewall be explicitly configured to DENY all incoming traffic and then you open up holes where necessary. You may need to open up port 80 to host web sites or port 21 to host an FTP file server. Each of these holes may be necessary from one standpoint, but they also represent possible vectors for malicious traffic to enter your network rather than being blocked by the firewall.

That is where your IDS would come in. Whether you implement a NIDS across the entire network or a HIDS on your specific device, the IDS will monitor the inbound and outbound traffic and identify suspicious or malicious traffic which may have somehow bypassed your firewall or it could possibly be originating from inside your network as well.

An IDS can be a great tool for proactively monitoring and protecting your network from malicious activity, however they are also prone to false alarms. With just about any IDS solution you implement you will need to "tune it" once it is first installed. You need the IDS to be properly configured to recognize what is normal traffic on your network vs. what might be malicious traffic and you, or the administrators responsible for responding to IDS alerts, need to understand what the alerts mean and how to effectively respond.

3.LITERATURE REVIEW

The aim of is to address the issues of information security and describes the security needs of an organization to protect their critical information from attacks. A well trained staff and analysts are required to continuously monitor the system. But still a huge effort is required to construct new security strategies in this system which is discussed. Provides a multilayer approach in IDPS to monitor a single host. Multilayer approach consists of three layers. File Analyzer, System Resource Analyzer and Connection Analyzer. The advantage of this technique is that it provides both signatures based and anomaly based detection and prevention. The drawback in Multilayer approach is that the IDPS require a large amount of memory to store the data of the system and network traffic.

Prevention desktop is software based solution which detects and protects the system from network layer up to application layer by known and unknown attacks. This software has great flexibility to set different type

of filtering rules. The major drawback of HIPS is its high rate of false-positives. A lot of time and trained staff is required to monitor the IDPS.

Marti et al. proposed a scheme named Watchdog that aims to enhance the throughput of network with the presence of malicious nodes. In reality, the Watchdog scheme consisted of two different parts, namely, Watchdog and Research Article July 2013 Anusha et al. Page 14 International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-2, Issue-7) Pathrater. Watchdog serves as an ID for MANETs and it is responsible for detecting the malicious node misbehavior's in the network. Watchdog detects the malicious misbehavior's by prominently listening to its next hop's broadcast. If a Watchdog node overhears that its next node fails to forward the packet within a definite period of time, it increases its failure counter. Whenever a node's failure counter exceeds a predefined threshold, the Watchdog node informs it as misbehaving node. In this case, the Path-rater cooperates with the routing protocols to avoid the reported nodes in future transmission. Many research studies and implementations have proved that the Watchdog scheme is efficient. Besides, compared to some other schemes, Watchdog is competent of detecting malicious nodes rather than links in the network. These advantages have made the Watchdog scheme a popular choice in the field. Nevertheless, as pointed out by Marti et al. , the Watchdog scheme fails to detect malicious misbehavior's with the presence of the following ie. Ambiguous collisions, receiver collisions, limited transmission power and false misbehavior report.

According to, Snort and source fire are best IPSs for a multinational company. Snort product provides high flexibility that allow to the user to self configure and modify its source code by using source fire. The major drawback of Snort is that it uses only signature based technique to detect the intrusion but if anomaly behavior occur then it will not be possible for SNORT to detect that anomaly attack.

This paper provides a technique of secure mobile agent in IDPS for the security of system. Secure mobile agent monitors the system, processes the logs, detects the attacks, and protects the host by automated real time response. Major disadvantage is that if the target of the attackers is mobile agent then it will be difficult to protect the system from being hacked. So it needs to adopt some security infrastructures for the protection of mobile agent.

David and Paolo in examined the technique which shows that how application interacts with the operating system and how (PH) IDS can be broken without detection, by using the technique of sequence matching, inserting malicious sequence and inserting no-op. This technique is unaware about that how much effort and knowledge is required to produce such an attack and also unaware about that how attackers can predict that how IDS actually works.

Harley defines the difference between host based and network based intrusion detection and prevention system. This paper describes two types of network intrusion detection system: Promiscuous-mode and Network-node. The main disadvantage observed is that this IDS only responds to the signature based detected attacks but not to the anomaly based detected attacks. So still there is a need of human interaction who took real time action to resolve issue .

Novel string matching technique is an optimization of other matching algorithms. Novel string matching algorithm breaks the string into small sets of state machines. Each state machine recognizes the subset of string. If any suspicious behavior occurs then the system broadcasts the information about intruder to every module (state machine) which holds the database in order to define rules and compares the signatures of intruder with predefined detected signatures. This algorithm is most efficient and ten times faster than the other existing systems and it consumes less resources. The major issue is its practical implementation and it requires a large amount of memory. This algorithm is not capable to detect the anomaly behavior of the intrusion.

4.PROPOSED SYSTEM

In fact, many of the existing IDSs in MANETs adopt an acknowledgment-based scheme, including TWO ACK and AACK. The functions of such detection schemes all largely depend on the acknowledgment packets. Hence, it is crucial to guarantee that the acknowledgment packets are valid and authentic. To address this concern, we adopt a digital signature in our proposed scheme named Enhanced AACK (EAACK).

5.DIGITAL SIGNATURE ALGORITHM

The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures. It was proposed by the National Institute of Standards and Technology (NIST) in August 1991 for use in their Digital Signature Standard (DSS) and adopted as FIPS 186 in 1993. Four revisions to the initial specification have been released: FIPS 186-1 in 1996, FIPS 186-2 in 2000, FIPS 186-3 in 2009, and FIPS 186-4 in 2013.

DSA is covered by U.S. Patent 5,231,668, filed July 26, 1991 and attributed to David W. Kravitz, a former NSA employee. This patent was given to "The United States of America as represented by the Secretary of Commerce, Washington, D.C.", and NIST has made this patent available worldwide royalty-free. Claus P. Schnorr claims that hisU.S. Patent 4,995,082 (expired) covered DSA; this claim is disputed. DSA is a variant of the ElGamal Signature Scheme

Key Generation

Key generation has two phases. The first phase is a choice of *algorithm parameters* which may be shared between different users of the system, while the second phase computes public and private keys for a single user.

Parameter generation

- Choose an approved cryptographic hash function *H*. In the original DSS, *H* was always SHA-1, but the stronger SHA-2 hash functions are approved for use in the current DSS. The hash output may be truncated to the size of a key pair.
- Decide on a key length *L* and *N*. This is the primary measure of the cryptographic strength of the key. The original DSS constrained *L* to be a multiple of 64 between 512 and 1024 (inclusive). NIST 800-57 recommends lengths of 2048 (or 3072) for keys with security lifetimes extending beyond 2010 (or 2030), using correspondingly longer *N*. FIPS 186-3 specifies *L* and *N* length pairs of (1024,160), (2048,224), (2048,256), and (3072,256).
- Choose an *N*-bit prime *q*. *N* must be less than or equal to the hash output length.
- Choose an *L*-bit prime modulus *p* such that *p*–1 is a multiple of *q*.
- Choose *g*, a number whose multiplicative order modulo *p* is *q*. This may be done by setting $g = h^{(p-1)/q} \bmod p$ for some arbitrary h ($1 < h < p-1$), and trying again with a different *h* if the result comes out as 1. Most choices of *h* will lead to a usable *g*; commonly $h=2$ is used.

The algorithm parameters (*p*, *q*, *g*) may be shared between different users of the system.

Per-user keys

Given a set of parameters, the second phase computes private and public keys for a single user:

- Choose *x* by some random method, where $0 < x < q$.
- Calculate $y = g^x \bmod p$.
- Public key is (*p*, *q*, *g*, *y*). Private key is *x*.

There exist efficient algorithms for computing the modular exponentiations $h^{(p-1)/q} \bmod p$ and $g^x \bmod p$, such as exponentiation by squaring.

- Signing

Let *H* be the hashing function and *m* the message:

- Generate a random per-message value *k* where $0 < k < q$
- Calculate $r = \left(g^k \bmod p\right) \bmod q$
- In the unlikely case that $r = 0$, start again with a different random k
- Calculate $s = k^{-1} \left(H\left(m\right) + xr\right) \bmod q$
- In the unlikely case that $s = 0$, start again with a different random k
- The signature is (r, s)

The first two steps amount to creating a new per-message key. The modular exponentiation here is the most computationally expensive part of the signing operation, and it may be computed before the message hash is known. The modular inverse $k^{-1} \bmod q$ is the second most expensive part, and it may also be computed before the message hash is known. It may be computed using the extended Euclidean algorithm or using Fermat's little theorem as $k^{q-2} \bmod q$.

- Verifying
- Reject the signature if $0 < r < q$ or $0 < s < q$ is not satisfied.
- Calculate $w = s^{-1} \bmod q$
- Calculate $u_1 = H\left(m\right) \cdot w \bmod q$
- Calculate $u_2 = r \cdot w \bmod q$
- Calculate $v = \left(\left(g^{u_1} y^{u_2}\right) \bmod p\right) \bmod q$

- The signature is valid if $v = r$

DSA is similar to the El Gamal signature scheme.

Correctness of the algorithm

The signature scheme is correct in the sense that the verifier will always accept genuine signatures. This can be shown as follows:

First, if $g = h^{(p-1)/q} \bmod p$ it follows that $g^q \equiv h^{p-1} \equiv 1 \pmod p$ by Fermat's little theorem. Since $g > 1$ and q is prime, g must have order q.

The signer computes

$$s = k^{-1}(H(m) + xr) \bmod q$$

Thus

$$\begin{aligned} k &\equiv H(m)s^{-1} + xrs^{-1} \\ &\equiv H(m)w + xrw \pmod q \end{aligned}$$

Since g has order q (mod p) we have

$$\begin{aligned} g^k &\equiv g^{H(m)w} g^{xrw} \\ &\equiv g^{H(m)w} y^{rw} \\ &\equiv g^{u1} y^{u2} \pmod p \end{aligned}$$

Finally, the correctness of DSA follows from

$$\begin{aligned} r &= (g^k \bmod p) \bmod q \\ &= (g^{u1} y^{u2} \bmod p) \bmod q \\ &= v \end{aligned}$$

Sensitivity

With DSA, the entropy, secrecy, and uniqueness of the random signature value *k* is critical. It is so critical that violating any one of those three requirements can reveal the entire private key to an attacker.[11] Using the same value twice (even while keeping *k* secret), using a predictable value, or leaking even a few bits of *k* in each of several signatures, is enough to break DSA.[12]

In December 2010, a group calling itself *fail0verflow* announced recovery of the ECDSA private key used by Sony to sign software for the PlayStation 3 game console. The attack was made possible because Sony failed to generate a new random *k* for each signature.[13]

This issue can be prevented by deriving *k* deterministically from the private key and the message hash, as described by RFC 6979. This ensures that *k* is different for each*H(m)* and unpredictable for attackers who do not know *x*.

Applying genetic algorithm to intrusion detection seems to be a promising area. We discuss the motivation and implementation details in this section.

6 GENETIC ALGORITHMS

6.1 Overview

Genetic algorithms can be used to evolve simple rules for network traffic (Sinclair, Pierce, and Matzner 1999). These rules are used to differentiate normal network connections from anomalous connections. These anomalous connections refer to events with probability of intrusions. The rules stored in the rule base are usually in the following form (Sinclair, Pierce, and Matzner 1999):

if { condition } then { act }

For the problems we presented above, the condition usually refers to a match between current network connection and the rules in IDS, such as source and destination IP addresses and port numbers (used in TCP/IP network protocols), duration of the connection, protocol used, etc., indicating the probability of an intrusion. The act field usually refers to an action defined by the security policies within an organization, such as reporting an alert to the system administrator, stopping the connection, logging a message into system audit files, or all of the above. For example, a rule can be defined as: if {the connection has following information: source IP address 124.12.5.18; destination IP address:
130.18.206.55; destination port number: 21; connection time: 10.1 seconds }
then {stop the connection}

This rule can be explained as follows: if there exists a network connection request with the source IP address 124.12.5.18, destination IP address 130.18.206.55, destination port number 21, and connection time 10.1 seconds, then stop this connection establishment. This is because the IP address 124.12.5.18 is recognized by the IDS as one of the blacklisted IP addresses; therefore, any service request initiated from it is rejected.

The final goal of applying GA is to generate rules that match only the anomalous connections. These rules are tested on historical connections and are used to filter new connections to find suspicious network traffic. In this implementation, the network traffic used for GA is a pre-classified data set that differentiates normal network connections from anomalous ones.

This data set is gathered using network sniffers (a program used to record network traffic without doing something harmful) such as Tcpdump (http://www.tcpdump.com) or Snort (http://www.snort.com). The data set is manually classified based on experts' knowledge. It is used for the fitness evaluation during the execution of GA. By starting GA with only a small set of randomly generated rules, we can generate a larger data set that contains rules for IDS. These rules are "good enough" solutions for GA and can be used for filtering new network traffic.

6.2 Data Representation

In order to fully exploit the suspicious level, we need to examine all fields related with a specific network connection. For simplicity, we only consider some obvious attributes for each connection. The definition of rules (for TCP/IP protocols) is shown in Table 1.

The corresponding rule for the "Example Value" attribute in Table 1 could be translated as:

if {the connection has following information: source IP address 209.11.??.??; destination IP address: 130.18.176+?.??; source port number: 42335; destination port number: 80; connection time: 482 seconds; the connection is stopped by the originator; the protocol used is TCP; the originator sent 7320 bytes of data; and the responder sent 38891 bytes of data } then {stop the connection}

Altogether there are fifty-seven genes in each chromosome. For simplicity, we use hexadecimal representations for the IP addresses. The rule can be explained as follows: if a network connection with source IP address 209.11.??.?? (209.11.0.0 ~ 209.11.255.255), destination IP address 130.18.176.?? (130.18.176.0 ~ 130.18.255.255), source port number 42335, destination port number 80, duration time 482 seconds, ends with state 11 (the connection terminated by the originator), uses protocol type 2 (TCP), and the originator sends 7320 bytes of data, the responders sends 38891 bytes of data, then this is a suspicious behavior and can be identified as a potential intrusion. The actual validity of this rule will be examined by matching the historical data set comprised of connections marked as either anomalous or normal. If the rule is able to find an anomalous behavior, a bonus will be given to the current chromosome. If the rule matches a normal connection, a penalty will be applied to the chromosome. Clearly no single rule can be used to separate all anomalous connections from normal connections. The population needs evolving to find the optimal rule set. In the example shown in Table 1, some wild cards (the '*' character and the '?' character) are used and the corresponding genes within the chromosome are shown as –1. These wild cards are used to represent an appropriate range of specific values (Crosbie and Spafford, 1995). It is useful when representing a network block (a range of IP addresses or port numbers) in a rule. Once the spatial information is included in the rules, the capability of the IDS can be greatly improved as an intrusion may initiate from many different locations. The inclusion of the duration time of a network connection in the chromosome ensures incorporation of temporal information for network connections. The maximum value of duration time is 99999999 seconds, which is more than a year. This is helpful for identifying intrusions because complex intrusions may span hours, days, or even months. The genetic algorithm starts with a population that has randomly selected rules. The population can evolve by using the crossover and mutations operators. Due to the effectiveness of the evaluation function, the succeeding populations are biased toward rules that match intrusive connections. Ultimately as the algorithm stops, rules are selected and added into the IDS rule base.

6.3 Parameters in Genetic Algorithm

There are many parameters to consider for the application of GA. Each of these parameters heavily influences the effectiveness of the genetic algorithm. We will discuss the methodology and related parameters in the following section.

Evaluation function

The evaluation function is one of the most important parameters in genetic algorithm. The proposed implementation differs from the scheme used by (Crosbie and Spafford, 1995) in that the definition on calculations of outcome and fitness is different. The following steps are used to calculate the evaluation function. First the overall outcome is calculated based on whether a field of the connection matches the pre-classified data set, and then multiply the weight of that field. The Matched value is set to either 1 or 0.

Outcome= Σ=

57

1

*

i

i Matched Weight

The order of weight values in the function is shown in Figure 4. These orders are categorized according to different fields in the connection record as reported by network sniffers. Therefore, all genes representing destination IP address field have the same weight. The actual values can be finely tuned at execution time. The

(d, 1, 0, b, -1, -1, -1, -1, 8, 2, 1, 2, b, -1, -1, -1, 4, 2, 3, 3,

5, 0, 0, 0, 8, 0, 0, 0, 0, 0, 0, 4, 8, 2, 1, 1, 2, 0, 0, 0,

0, 0, 0, 7, 3, 2, 0, 0, 0, 0, 0, 0, 3, 8, 8, 9, 1)

5

basic idea behind this order is the importance of different fields in TCP/IP packets. This scheme is straightforward and intuitive. Destination IP address is the target of an intrusion while the source IP address is the originator of the intrusion. These are the most important pieces of information needed to capture an intrusion. Destination port number indicates to applications that the target system is running (for example, FTP service usually runs on port 21). Some IP addresses are more probable targets for intrusions—for example, IP addresses for military domains. Domain-specific information is less important compared with the source IP addresses. Other parameters like duration, bytes sent by the originator, bytes sent by the receiver, and state are usually less important than the above fields but are still useful. The protocol and source port number fields are commonly dispensable and are used for identifying some specific intrusions.

The absolute difference between the outcome of the chromosome and the actual suspicious level is then computed using the following equation. The suspicious_level is a threshold that indicates the extent to which two network connections are considered a "match." The actual value of suspicious_level reflects observations from historical data.

Δ = | outcome – suspicious_level |

Once a mismatch happens, the penalty value is computed using the absolute difference. The ranking in the equation indicates whether or not an intrusion is easy to identify.

)

100

penalty (* ranking Δ

=

The fitness of a chromosome is computed using the above penalty:

fitness = 1- penalty

Obviously, the range of the fitness value is between 0 and 1. By defining evaluation, we have incorporated both temporal and spatial information needed for identification of network intrusions.

Destination IP address

Source IP address

Duration

Protocol

Destination Port Number

Source Port Number

State

High

Low

Bytes sent by the Originator

Bytes sent by the Receiver

6

Crossover and Mutation

Traditional genetic algorithms have been used to identify and converge populations of candidate hypotheses to a single global optimum. For this problem, a set of rules is needed as a basis for the IDS. As mentioned earlier, there is no way to clearly identity whether a network connection is normal or anomalous just using one rule. Multiple rules are needed to identify unrelated anomalies, which means that several good rules are more effective than a single best rule (Sinclair, Pierce, and Matzner 1999). Another reason for finding multiple rules is that because there are so many network connection possibilities, a small set of rules will be far from enough.

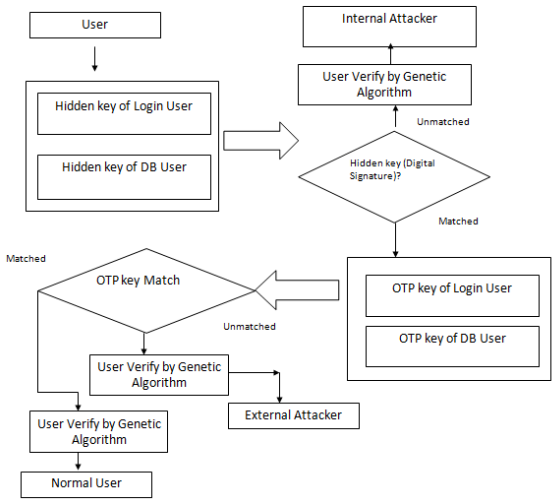
Using the genetic algorithm, we need to find local maxima (a set of "good-enough" solutions) as opposed to the global maximum (the best solution) (Sinclair, Pierce, and Matzner 1999). The niching techniques can be used to find multiple local maxima (Miller and Shaw, 1996; see also Sinclair, Pierce, and Matzner 1999). It is based on the analogy to nature in that within each environment, there are different subspaces (niches) that can support different types of life. In a similar manner, genetic algorithm can maintain the diversity of each population in a multimodal

domain, which refers to domains requiring the identification of multiple optima. Two basic methods, crowding and sharing can be used for niching. The crowding method uses the most similar member for

replacement to slow down the population to converge towards a single point in the following generations. The sharing method reduces the fitness of individuals that have highly similar members and forces individuals to evolve to other local maxima that may be less populated. The similarity metrics used in these techniques can be phenotype to genotype similarity such as Hamming distance between bit representations, or phenotype similarity such as the relation between two network connections in this problem. The latter one is more fitful for finding rules used in IDS. The disadvantage of this approach is that it requires more domain-specific knowledge (Miller and Shaw, 1996; see also Sinclair, Pierce, and Matzner 1999).

The mutation operation should be meaningful during evolution. For example, each segment of the IP address should not exceed 255 (decimal representation). Mutations should be done following the requirements specified in Table 1. These limitations can be enforced by defining proper mutation rules.

7. IMPLEMENTATION AND EXPERIMENT RESULT



On a Server-based network, special computers, known as Servers, process data for and facilitate communication between the other computers on the network, which are known as Clients. Clients usually take the form of desktop computers, known as workstations. A network that uses a Server to enable Clients to share data, data storage space, and devices, is known as a Client/Server network.

Let us see some of the key terms:

Client: A computer on the network that requests resources or services from another computer on a network. In some cases, a Client could also act as a Server. The term "Client" may also refer to the human user of a Client workstation.

Server: A computer on the network that manages shared resources. Servers usually have more processing power, memory, and hard disk space than Clients. They run network operating software that can manage not only data, but also users, groups, security, and applications on the network.

Workstation: A desktop computer, which may or may not be connected to a network. Most Clients are workstation computers.

Network interface card (NIC): The device that enables a workstation to connect to the network and communicate with other computers. Several companies (such as 3Com, IBM, Intel, SMC, and Xircom) manufacture NICs, which come with a variety of specifications that are tailored to the requirements of the workstation and the network.

Network operating system (NOS): The software that runs on a Server and enables the Server to manage data, users, groups, security, applications, and other networking functions. The most popular network Operating Systems are Microsoft Windows NT, Windows 2000, Novell NetWare, and UNIX.

Host: A Server that manages shared resources.

Node: A Client, Server, or other device that can communicate over a network and that is identified by a unique identifying number, known as its network address.

Topology: The physical layout of a computer network. Topologies vary according to the needs of the organization and available hardware and expertise. Networks are usually arranged in a ring, bus, or star formation; hybrid combinations of these patterns are also possible.

Protocol: The rules that the network uses to transfer data. Protocols ensure that data are transferred whole,

in sequence, and without error from one node on the network to another. To effectively maintain and manage a network, you must have a thorough understanding of network protocols.

Data packets: The distinct units of data that are transmitted from one computer on a network to another. Data packets are also known as datagram's, protocol data units (PDU), frames, or cells, depending on the context.

Addressing: It is a scheme for assigning a unique identifying number to every workstation and device on the network. The type of addressing used depends on the network's protocols and network Operating System. It is important that each computer on a network have a unique address so that data can be transmitted reliably to and from that computer.

Transmission media: The means through which data are transmitted and received. Transmission media may be physical, such as wire or cable, or atmospheric (wireless), such as radio waves.

Sockets: Socket is one of the most prevalent communication application program interfaces (API) to the communication protocols. The API is an interface available to a programmer. The availability of an API depends on the both the Operating System being used and the programming language.

Connection and Associations: The term connection is used to define the communication link between two processes. The term association is used for the 5-tuple that completely specifies the two processes that make up a connection:
{Protocol, local-address, local-process, foreign-address, foreign-process}
The local-address and foreign-address specify the network ID and the host ID of the local host and foreign host, in whatever format is specified by the protocol suite. The local-process and foreign-process are used to identify the specific processes on each system that are involved in the connection, again in whatever format are defined by the protocol suite. We also define a half association either as {protocol, local-address, local-process} or {protocol, foreign-address, foreign-process}, which specify each half of a connection. This half association is also called a socket.

7.CONCLUSION

The software development is very flexible and much functionality can be added to it, to enhance performance of this paper titled "EAACK- Intrusion Detection System .By using genetic algorithm, during run time the new set of rules will added in the dataset. A brief overview of Intrusion Detection System, Genetic algorithm, digital signature algorithm and related detection techniques are discussed. This implementation of genetic algorithm and digital signature algorithm is unique as it considers both temporal and spatial information of network connections during the encoding of the problem; therefore, it should be more helpful for identification of network anomalous behaviors.
We discussed a methodology of applying genetic algorithm into network intrusion detection techniques. A brief overview of Intrusion Detection System (IDS), genetic algorithm, Digital Algorithm (DSA) and related detection techniques are discussed. The system architecture is also introduced. Factors affecting the GA are addressed in detail. This implementation of genetic algorithm is unique as it considers both temporal and spatial information of network connections during the encoding of the problem; therefore, it should be more helpful for identification of network anomalous behaviors. Future work includes creating a standard test data set for the genetic algorithm proposed in this paper and applying it to a test environment. Detailed specification of parameters to consider for genetic algorithm should be determined during the experiments. Combining knowledge from different security sensors into a standard rule base is another promising area in this work.

In future in this system we improved cryptographic technique for secure transmission of packets in mobile Ad-hoc network .So that it will provide more security during data transmission from attacker & provide better prevention.
To increase the merits of the present work, we do have plans to investigate the following issues in our future research:
1) Possibilities of implementing hybrid cryptography techniques to further reduce the network overhead caused by security.

2) Observe the possibilities of adopting a key exchange mechanism in order to remove the requirement of pre distributed keys.

REFERENCE

[1] Elhadi M. Shakshuki, Senior Member, IEEE, Nan Kang, and Tarek R. Sheltami, Member, IEEE, "EAACK—A Secure Intrusion-Detection System for MANETs", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 60, NO. 3, MARCH 2013.

[2] Bezroukov, Nikolai. 19 July 2003. "Intrusion Detection (general issues)." Softpanorama: Open-Source Software Educational Society. Nikolai Bezroukov. URL: http://www.softpanorama.org/ Security/intrusion_detection.shtml (30Oct. 2003).

[3] Bridges, Susan, and Rayford B. Vaughn. 2000. "Intrusion Detection Via Fuzzy Data Mining." In Proceedings of 12th Annual Canadian Information Technology Security Symposium, pp. 109-122. Ottawa, Canada.

[4] Crosbie, Mark, and Gene Spafford. 1995. "Applying Genetic Programming to Intrusion Detection." In Proceedings of 1995 AAAI Fall Symposium on Genetic Programming, pp. 1-8. Cambridge, Massachusetts. URL: http://citeseer.nj.nec.com/crosbie95applying.htm l (30 Oct. 2003).

[5] Graham, Robert. Mar. 21, 2000. "FAQ: Network Intrusion Detection Systems." RobertGraham. com Homepage. Robert Graham. URL: http://www.robertgraham.com/pubs/network-intrusion-detection.html (30 Oct. 2003).

[6] Jones, Anita. K. and Robert. S. Sielken. 2000. "Computer System Intrusion Detection: A Survey." Technical Report.

[7] Department of Computer Science, University of Virginia, Charlottesville, Virginia.

[8] Li, Wei. 2002. "The integration of security sensors into the Intelligent Intrusion Detection System (IIDS) in a cluster

[9] Ms.Nivedita Naidu "An Effective Approach to Network Intrusion Detection System using Genetic Algorithm".