IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 10, Issue 5 (Mar. - Apr. 2013), PP 38-45
www.iosrjournals.org

Intrusion Detection Techniques In Mobile Networks

M.Senthilkumar, S.Saminathan
1. Asst.Prof in Computer Science, Srinivasan College of Arts & Science, Perambalur, Tamilnadu, India.
2. Asst.Prof in Computer Science, Srinivasan College of Arts & Science, Perambalur, Tamilnadu, India.

Abstract: *The rapid proliferation of wireless networks and mobile computing applications has changed the landscape of network security. The recent denial of service attacks on major Internet sites have shown us, no open computer network is immune from intrusions. The wireless ad-hoc network is particularly vulnerable due to its features of open medium, dynamic changing topology, cooperative algorithms, lack of centralized monitoring and management point, and lack of a clear line of defense. The traditional way of protecting networks with firewalls and encryption software is no longer sufficient and effective.*

Many intrusion detection techniques have been developed on fixed wired networks but have been turned to be inapplicable in this new environment. We need to search for new architecture and mechanisms to protect wireless networks and mobile computing application. In this paper, we examine the vulnerabilities of wireless networks and say that we must include intrusion detection in the security architecture for mobile computing environment. We have showed such architecture and evaluated key mechanisms in this architecture such as applying mobile agents to intrusion detection, anomaly detection and misuse detection for mobile ad-hoc networks.

Keywords – *Algorithm, Architecture, Computing Network, Dynamic Changing Topology, Wireless Networks.*

I. Introduction

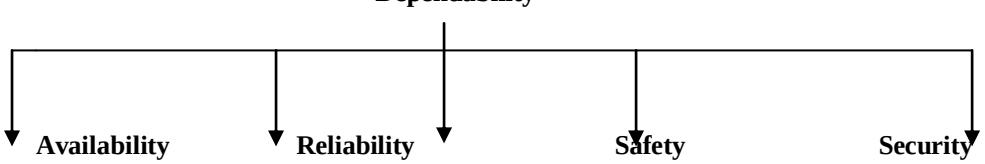
In the last three years, the networking revolution has finally come of age. More than ever before, we see that the Internet is changing computing, as we know it. The possibilities and opportunities are limitless; unfortunately, so too are the risks and chances of malicious intrusions.

It is very important that the security mechanisms of a system are designed so as to prevent unauthorized access to system resources and data. However, completely preventing breaches of security appear, at present, unrealistic. We can, however, try to detect these intrusion attempts so that action may be taken to repair the damage later. This field of research is called **Intrusion Detection.**

1.1 Computer Security and its Role

One broad definition of a secure computer system is given by Garfinkel and Spafford as ***one that can be depended upon to behave as it is expected to***. It is always a point of benefit to integrate security with dependability and how to obtain a dependable computing system.

Dependability is the trustworthiness of a system and can be seen as the quality of the service a system offers. Integrating security and dependability can be done in various ways. One approach is to treat security as one characteristic of dependability on the same level as availability, reliability and safety as shown in the figure.



A narrower definition of **security** is ***the possibility for a system to protect objects with respect to confidentiality, authentication, integrity and non-repudiation***.

Confidentiality: Transforming data such that only authorized parties can decode it.
Authentication: Proving or disproving someone's or something's claimed identity.
Integrity checking: Ensuring that data cannot be modified without such modification being detectable
Non – repudiation: Proving that a source of some data did in fact send data that he might later deny sending

1.2 Threats of security

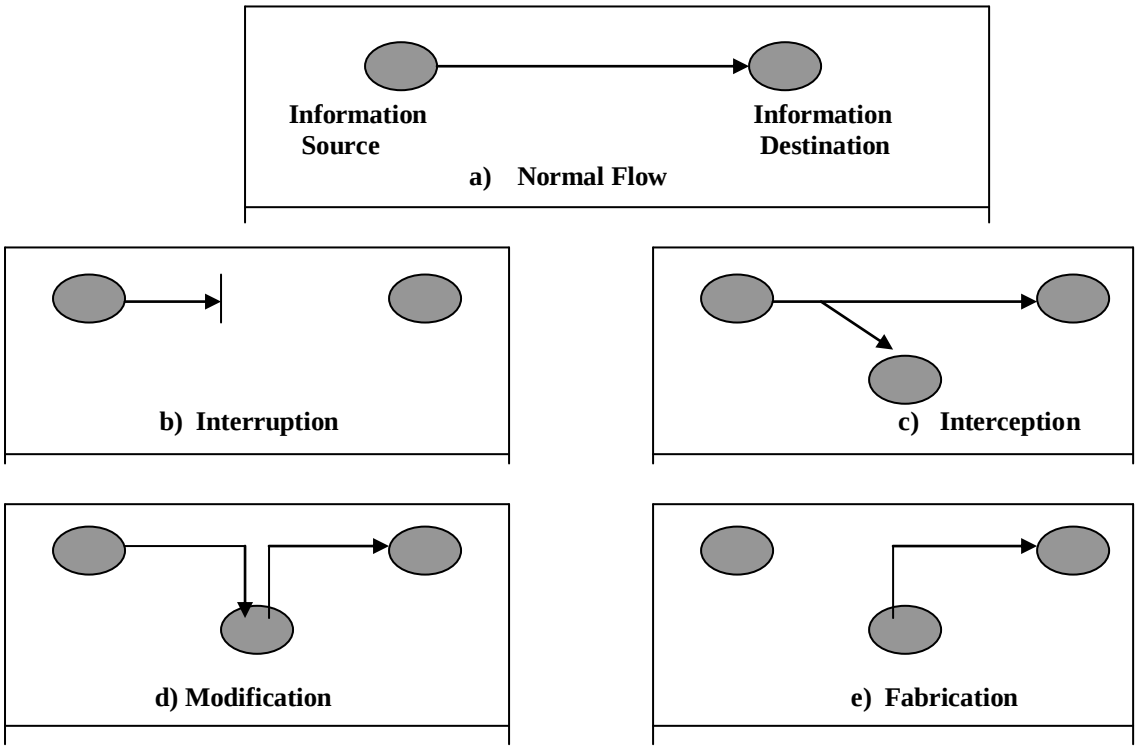
Threats can be seen as potential violations of security and exist because of vulnerabilities, i.e. weakness, in a system. There are two basic types of threats: **accidental threats** and **intentional threats**.

1.2.1 Accidental Threat:

An accidental threat can be manifested and the result is either an exposure of confidential information or cause of an illegal system state to occur i.e. modification of an object. Exposures can emerge from both hardware and software failures as well as from user and operational mistakes thus resulting in the violation of confidentiality. It can also be manifested as modification of an object, which is the violation of object integrity. An object here can be both information and resource.

1.2.2 Intentional Threat:

An intentional threat is an action performed by an entity with the intention to violate security. Examples of attacks are interruption, modification, interception and fabrication of data as shown in the figure



1.3 Vulnerabilities Of Mobile Wireless Networks

The nature of mobile computing environment makes it very vulnerable to an adversary's malicious attacks.

Firstly, the use of wireless links renders the network susceptible to attacks ranging from passive eavesdropping to active interfering as attacks on these links can from any direction and target at any node. This means that a wireless ad-hoc network will not have a clear line of defense, and every node has to be prepared for encounters with an adversary directly or indirectly.

Secondly, mobile nodes are autonomous units that are capable of roaming independently. Since tracking down a particular mobile node in a global scale network cannot be done easily, attacks by compromised node from within the network are more damaging and harder to detect.

Third, decision-making in mobile computing environment is sometimes decentralized and some wireless network algorithms rely on the cooperative participation of all nodes and the infrastructure.

Furthermore, mobile computing has introduced new type of computational and communication activities that seldom appear in fixed or wired environment. Applications and services in a mobile wireless network can be a weak link as well.

To summarize, a mobile wireless network is vulnerable due to its features of open medium, dynamic changing topology, cooperative algorithms, lack of centralized monitoring and management point, and lack of a clear line of defense.

1.4 Need For Intrusion Detection

A computer system should provide *confidentiality*, *integrity* and *assurance* against denial of service. However, due to increased connectivity (especially on the Internet), and the vast spectrum of financial possibilities that are opening up, more and more systems are subject to attack by intruders. These subversion attempts try to exploit flaws in the operating system as well as in application programs and have resulted in spectacular incidents like the Internet Worm incident of 1988.

There are two ways to handle subversion attempts. One way is to prevent subversion itself by building a completely secure system. We could, for example, *require* all users to identify and authenticate themselves; we could protect data by various cryptographic methods and very tight access control mechanisms. However this is not really feasible because:

1. In practice, it is not possible to build a completely secure system. Miller gives a compelling report on bugs in popular programs and operating systems that seems to indicate that (a) bug free software is still a dream and (b) no-one seems to want to make the effort to try to develop such software. Apart from the fact that we do not seem to be getting our money's worth when we buy software, there are also security implications when our E-mail software, for example, can be attacked. Designing and implementing a totally secure system is thus an extremely difficult task.
2. The vast installed base of systems worldwide guarantees that any transition to a secure system, (if it is ever developed) will be long in coming.
3. Cryptographic methods have their own problems. Passwords can be cracked, users can lose their passwords, and entire crypto-systems can be broken.
4. Even a truly secure system is vulnerable to abuse by insiders who abuse their privileges.
5. It has been seen that that the relationship between the level of access control and user efficiency is an inverse one, which means that the stricter the mechanisms, the lower the efficiency becomes.

The history of security research has taught us a valuable lesson – no matter how many intrusion prevention measures are inserted in a network, there are always some weak links that one could exploit to break in.

We thus see that we are stuck with systems that have vulnerabilities for a while to come. If there are attacks on a system, we would like to detect them as soon as possible (preferably in real-time) and take appropriate action. This is essentially what an Intrusion Detection System (IDS) does. An IDS does not usually take preventive measures when an attack is detected; it is a reactive rather than pro-active agent. It plays the role of an informant rather than a police officer.

II. Background On Intrusion Detection

In the last three years, the networking revolution has finally come of age. More than ever before, we see that the Internet is changing computing, as we know it. The possibilities and opportunities are limitless; unfortunately, so too are the risks and chances of malicious intrusions.

It is very important that the security mechanisms of a system are designed so as to *prevent* unauthorized access to system resources and data. However, completely preventing breaches of security appear, at present, unrealistic. We can, however, try to detect these intrusion attempts so that action may be taken to repair the damage later. This field of research is called **Intrusion Detection**

A simple firewall can no longer provide enough security as in the past. Today's corporations are drafting intricate security policies whose enforcement requires the use of multiple systems, both proactive and reactive (and often multi-layered and highly redundant). The premise behind intrusion detection systems is simple: Deploy a set of agents to inspect network traffic and look for the "signatures" of known network attacks. However, the evolution of network computing and the awesome availability of the Internet have complicated this concept somewhat. With the advent of Distributed Denial of Service (DDOS) attacks, which are often launched from hundreds of separate sources, the traffic source no longer provides reliable temporal clues that an attack is in progress. Worse yet, the task of responding to such attacks is further complicated by the diversity of the source systems, and especially by the geographically distributed nature of most attacks.

Intrusion detection techniques while often regarded as grossly experimental, the field of intrusion detection has matured a great deal to the point where it has secured a space in the network defense landscape alongside firewalls and virus protection systems. While the actual implementations tend to be fairly complex, and often proprietary, the concept behind intrusion detection is a surprisingly simple one: Inspect all network activity (both inbound and outbound) and identify suspicious patterns that could be evidence of a network or system attack.

2.1 Classification of Intrusion Detection Systems

Intrusions can be divided into 6 main types

- Attempted break-ins, which are detected by atypical behavior profiles or violations of security constraints.
- Masquerade attacks, which are detected by atypical behavior profiles or violations of security constraints.
- Penetration of the security control system, which are detected by monitoring for specific patterns of activity.
- Leakage, which is detected by atypical use of system resources.
- Denial of service, which is detected by atypical use of system resources.
- Malicious use, which is detected by atypical behavior profiles, violations of security constraints, or use of special privileges.

2.2 IDS REQUIREMENTS

At least one past effort has identified desirable characteristics for an IDS. Regardless on what mechanisms an IDS is based, it must do the following:

- Run continuously without human supervision,
- Be fault tolerant and survivable,
- Resist subversion,
- Impose minimal overhead,
- Observe deviations from normal behavior
- Be easily tailored to a specific network
- Adapt to changes over time, and
- Be difficult to fool.

We have developed a similar set of requirements along two themes: functional and performance requirements.

2.2.1 Functional Requirements

As the network-computing environment increases in complexity, so do the functional requirements of IDSs. Common functional requirements of an IDS being deployed in current or near-term operational computing environments include the following:

- ✓ The IDS must continuously monitor and report intrusions.
- ✓ The IDS must supply enough information to repair the system, determine the extent of damage, and establish responsibility for the intrusion.
- ✓ The IDS should be modular and configurable as each host and network segment will require their own tests and these tests will need to be continuously upgraded and eventually replaced with new tests.
- ✓ Since the IDS is assigned the critical role of monitoring the security state of the network, the IDS itself is a primary target of attack. The IDS must be able to operate in a hostile computing environment and exhibit a high degree of fault-tolerance and allow for graceful degradation.
- ✓ The IDS should be adaptive to network topology and configuration changes as computing elements are dynamically added and removed from the network.
- ✓ The IDS should be able to learn from past experiences and improve its detection capabilities over time. A self-tuning IDS will be able to learning from false alarms with the guidance of system administrators and eventually on its own.
- ✓ The IDS should be able to be easily and frequently updated with attack signatures as new security advisories and security patches become available and new vulnerabilities and attacks are discovered.
- ✓ Decision support tools will be necessary to help system administrators respond to various attacks. The IDS will be required not only to detect anomalous events, but also to take automated corrective action.
- ✓ The IDS should be able to perform data fusion and be able to process information from multiple and distributed data sources such as firewalls, routers, and switches. As real-time detection demands push networked-based solutions to re-programmable hardware devices that can download new capabilities, the IDS will need to be able to communicate with the hardware-based devices.
- ✓ Data reduction tools will be necessary to help the IDS process the information gathered from data fusion techniques. Data mining tools will be helpful in running statistical analysis tools on archived data in support of anomaly detection techniques.
- ✓ The IDS should be capable of providing an automated response to suspicious activity.
- ✓ Rapid changes in network conditions and limited network administration expertise make it difficult for system administrators to diagnose problems and take corrective action to minimize the damage that intruders can cause.

- ✓ The ability to detect and react to distributed and coordinated attacks will become necessary. Coordinated attacks against a network will be able to marshal greater forces and launch many more and varied attacks against a single target. These attacks can be permutations of known attacks, be rapidly evolving, and be launched at little cost to the attackers.
- ✓ Distributing the computational load and the diagnostic capabilities to agents scattered throughout the network adds a level of fault-tolerance, but it is often necessary for the system administrator to have control over the IDS from a central location.
- ✓ The IDS should be able to work with other Commercial Off-the-Shelf (COTS) security tools, as no vendor toolset is likely to excel in or to provide complete coverage of the detection, diagnosis, and response responsibilities. The IDS framework should be able to integrate various data reduction, forensic, host-based, and network-based security tools. Interoperability and conformance to standards will further increase the value of the IDS.
- ✓ IDS data often requires additional analysis to assess any damage to the network after an intrusion has been detected. Although the anomalous event was the first detected, it may not be the first attempt to gain unauthorized access to the network. Post event analysis will be needed to identify compromised machines before the network can be restored to a safe condition.
- ✓ The IDS itself must also be designed with security in mind. For example, the IDS must be able to authenticate the administrator, audit administrator actions, mutually authenticate IDS devices, protect the IDS data, and not create additional vulnerabilities.

2.2.2 Performance Requirements

An IDS that is functionally correct, but that detects attacks too slowly is of little use. Thus we must enumerate several performance requirements for IDSs. The IDS performance requirements include:

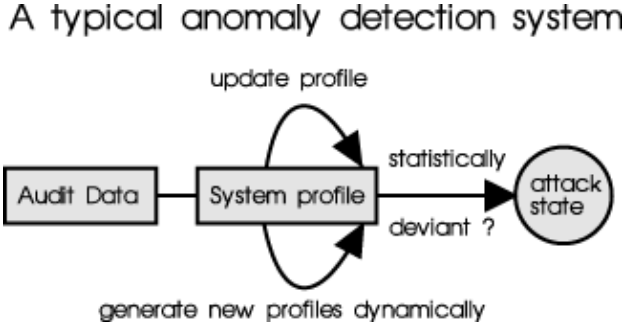
- ❖ To the extent possible, anomalous events or breaches in security should be detected in real-time and reported immediately to minimize the damage to the network and the loss or corruption of confidential information.
- ❖ The IDS must not place undue burden or interfere with the normal operations for which the systems were bought and deployed to begin with. This requirement makes it necessary for the agents to be cognizant of the consumption of network resources for which they are competing.
- ❖ The IDS must be scalable. As new computing devices are added to the network, the IDS must be able to handle the additional computational and communication load.

We can divide the techniques of intrusion detection into two main types.

2.3 Anomaly Detection :

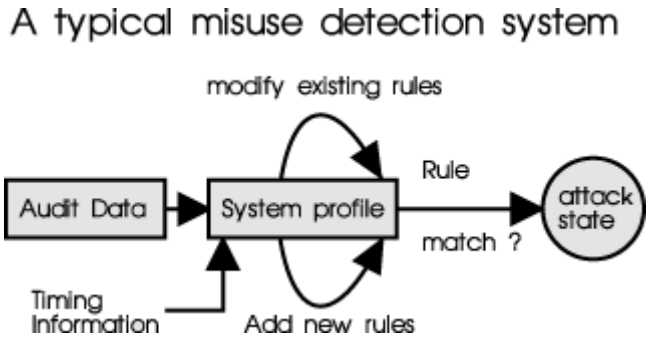
Anomaly detection techniques assume that all intrusive activities are necessarily anomalous. This means that if we could establish a "normal activity profile" for a system, we could, in theory, flag all system states varying from the established profile by statistically significant amounts as intrusion attempts. However, if we consider that the set of intrusive activities only intersects the set of anomalous activities instead of being exactly the same, we find a couple of interesting possibilities: (1) Anomalous activities that are not intrusive are flagged as intrusive. (2) Intrusive activities that are not anomalous result in false negatives (events are not flagged intrusive, though they actually are). This is a dangerous problem, and is far more serious than the problem of false positives.

The main issues in anomaly detection systems thus become the selection of threshold levels so that neither of the above 2 problems is unreasonably magnified, and the selection of features to monitor. Anomaly detection systems are also computationally expensive because of the overhead of keeping track of, and possibly updating several system profile metrics. Some systems based on this technique are discussed in Section 4 while a block diagram of a typical anomaly detection system is shown in Figure below.



2.4 Misuse Detection:

The concept behind misuse detection schemes is that there are ways to represent attacks in the form of a pattern or a signature so that even variations of the same attack can be detected. This means that these systems are not unlike virus detection systems -- they can detect many or all *known* attack patterns, but they are of little use for as yet unknown attack methods. An interesting point to note is that anomaly detection systems try to detect the complement of "bad" behavior. Misuse detection systems try to recognize known "bad" behavior. The main issues in misuse detection systems are how to write a signature that encompasses *all* possible variations of the pertinent attack, and how to write signatures that do not also match non-intrusive activity. A block diagram of a typical misuse detection system is shown in Figure below.



2.5 Network Based Intrusion Detection:

The most obvious location for an intrusion detection system is right on the segment being monitored. Network-based intrusion detectors insert themselves in the network just like any other device, except they promiscuously examine every packet they see on the wire.

2.6 Host Based Intrusion Detection

While network-based intrusion detectors are straightforward to deploy and maintain, there is a whole class of attacks closely coupled to the target system and extremely hard to fingerprint. These are the ones that exploit vulnerabilities particular to specific operating systems and application suites. Only host-based intrusion detection systems (the ones running as an application on a network-connected host) can correlate the complex array of system-specific parameters that make up the signature of a well-orchestrated attack.

III. Anomaly Detection Systems:

There have been a few approaches to anomaly intrusion detection systems, some of which are described below.

3.1 Statistical Approaches:

In this method, initially, behavior profiles for subjects are generated. As the system continues running, the anomaly detector constantly generates the variance of the present profile from the original one. We note that, in this case, there may be several measures that affect the behavior profile, like activity measures, CPU time used, number of network connections in a time period, etc. In some systems, the current profile and the previous profile are merged at intervals, but in some other systems profile generation is a one time activity.

An open issue with statistical approaches in particular, and anomaly detection systems in general, is the selection of measures to monitor. It is not known exactly what the subset of all possible measures that accurately predicts intrusive activities is. Static methods of determining these measures are sometimes misleading because of the unique features of a particular system. Thus, it seems that a combination of static and dynamic determination of the set of measures should be done. Some problems associated with this technique have been remedied by other methods, including the method involving *Predictive Pattern Generation*, which takes past events into account when analyzing the data.

3.2 Predictive Pattern Generation:

This method of intrusion detection tries to predict future events based on the events that have already occurred. Therefore, we could have a rule

E1 - E2 --> (E3 = 80%, E4 = 15%, E5 = 5%)

This would mean that given that events E1 and E2 have occurred, with E2 occurring after E1, there is an 80% probability that event E3 will follow, a 15% chance that event E4 will follow and a 5% probability that event E5 will follow.

Problem:

The problem with this is that some intrusion scenarios that are not described by the rules will not be flagged intrusive. Thus, if an event sequence A - B - C exists that is intrusive, but not listed in the rule base, it will be classified as unrecognized.

Solution:

The above problem can be partially solved by flagging any unknown events as intrusions (increasing the probability of false positives), or by flagging them as non-intrusive (thus increasing the probability of false negatives). In the normal case, however, an event is flagged intrusive if the left hand side of a rule is matched, but the right hand side is statistically very deviant from the prediction.

3.3. Neural Networks:

Another approach taken in intrusion detection systems is the use of **neural networks**. The idea here is to train the neural network to predict a user's next action or command, given the window of 'n' previous actions or commands. The network is trained on a set of representative user commands. After the training period, the network tries to match actual commands with the actual user profile already present in the net. Any incorrectly predicted events actually measure the deviation of the user from the established profile.

IV. Misuse Detection Systems:

There has been significant research in misuse detection systems in the recent past. Some of these systems are explained in depth in this section.

4.1 Expert Systems:

These systems are modeled in such a way as to separate the rule matching phase from the action phase. The matching is done according to audit trail events. IDES follows a hybrid intrusion detection technique consisting of a misuse detection component as well as an anomaly detection component. The anomaly detector is based on the statistical approach, and it flags events as intrusive if they are largely deviant from the expected behavior. To do this, it builds user profiles based on many different criteria (more than 30 criteria, including CPU and I/O usage, commands used, local network activity, system errors etc.). These profiles are updated at periodic intervals. The expert system misuse detection component encodes known intrusion scenarios and attack patterns (bugs in old versions of send mail could be one vulnerability). The rule database can be changed for different systems.

4.2 Keystroke monitoring:

This is a very simple technique that monitors keystrokes for attack patterns. Unfortunately the system has several defects -- features of shells like *bash*, *ksh*, and *tcsh* in which user definable aliases are present defeat the technique unless alias expansion and semantic analysis of the commands is taken up. The method also does not analyze the running of a program, only the keystrokes. This means that a malicious program cannot be flagged for intrusive activities. Operating systems do not offer much support for keystroke capturing, so the keystroke monitor should have a hook that analyses keystrokes before sending them on to their intended receiver. An improvement to this would be to monitor system calls by application programs as well, so that an analysis of the program's execution is possible.

4.3 Model Based Intrusion Detection

States that certain scenarios are inferred by certain other observable activities. If these activities are monitored, it is possible to find intrusion attempts by looking at activities that infer a certain intrusion scenario. The model-based scheme consists of three important modules. The *anticipator* uses the active models and the scenario models to try to predict the next step in the scenario that is expected to occur. A scenario model is a knowledge base with specifications of intrusion scenarios. The *planner* then translates this hypothesis into a format that shows the behavior, as it would occur in the audit trail. It uses the predicted information to plan what to search for next. The *interpreter* then searches for this data in the audit trail. The system proceeds this way, accumulating more and more evidence for an intrusion attempt until a threshold is crossed; at this point, it signals an intrusion attempt.

This is a very clean approach. Because the planner and the interpreter know what they are searching for at each step, the large amounts of noise present in audit data can be filtered, leading to excellent performance

improvements. In addition, the system can predict the attacker's next move based on the intrusion model. These predictions can be used to verify an intrusion hypothesis, to take preventive measures, or to determine what data to look for next.

However, there are some critical issues related to this system. First, patterns for intrusion scenarios must be easily recognized. Second, patterns must always occur in the behavior being looked for. And finally, patterns must be *distinguishing*; they must *not* be associated with any other normal behavior.

V. Ids Issues In Mobile Environment

Intrusion detection for traditional, wired networks has been the topic of significant research over the past few years. A problem arises, however, when taking the research for wired networks and directly applying it to wireless networks. Key assumptions are made when designing IDS s for wired networks, such as the difficulty for an attacker to penetrate the physical security of the system, the amount of network bandwidth available to the IDS, etc. Specific problems faced when building IDS for a mobile network are addressed below:

- Lack of Physical Wires
- Bandwidth Issues
- Difficulty of Anomaly/Normality Distinction
- Secure Communication Between IDS Agents
- Lack of Centralized Access/Audit Point
- Possibility of a Node Being Compromised
- Difficulty In Obtaining Enough Audit Data

VI. Conclusion

The diligent management of network security is essential to the operation of networks, regardless of whether they have segments or not. It is important to note that absolute security is an abstract concept – it does not exist anywhere. All networks are vulnerable to insider or outsider attacks, and eavesdropping. No one wants to risk having the data exposed to the casual observer or open malicious mischief. Regardless of whether the network is wired or wireless, steps can and should always be taken to preserve network security and integrity.

We have said that any secure network will have vulnerabilities that an adversary could exploit. This is especially true for wireless ad-hoc networks. Intrusion Detection can compliment intrusion prevention techniques (such as encryption, authentication, secure MAC, secure routing, etc.) to improve the network security. However new techniques must be developed to make intrusion detection work better for the wireless networks.

We have shown that an architecture for better intrusion detection in wireless networks should be distributed and cooperative by applying Mobile Agents to the network and given few of the implemented approaches for intrusion detection.

References:

[1]. Sundaram A., "An Introduction to Intrusion Detection", http://www.acm.org/crossroads/xrds2-4/intrus.html
[2]. Marti S., Giuli T.J., Lai K. Baker M., "*Mitigating Routing Misbehavior in Mobile Ad Hoc Networks*", Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM 2000, pp 255-265.
[3]. Andrew B.Smith, An Examination of an Intrusion Detection Architecture for eless Ad-Hoc Networks.
[4]. C. Krugel , T.Toth. , Applying Mobile Agent Technology to Intrusion Detection
[5]. Kumar.S "Classification and Detection of Computer Intrusion ".