International Journal of Computer Theory and Engineering, Vol.2, No.6, December, 2010
1793-8201

Intrusion Detection Tools and Techniques – A Survey

Karthikeyan .K.R[1] and A. Indra[2]

Abstract—**Attacks like Denial of Service, Buffer overflows, Sniffer attacks and Application-Layer attacks have become a common issue today. Recent security incidents and analysis have demonstrated that manual response to such attacks is no longer feasible. Network security attacks aren't some theoretical concept that can be put into the background and dealt with later. Attacks of various types happen every day out in the wilds. Firewalls and spam filters are in place but they have simple rules such as to allow or deny protocols, ports or IP addresses. Some DoS attacks are too complex for today's firewalls, e.g. if there is an attack on port 80 (web service), firewalls cannot prevent that attack because they cannot distinguish good traffic from DoS attack traffic. In this paper we define and discuss various types and techniques of Intrusion Detection and the IDS tools that are employed to detect these attacks. We also present a description of types of security attacks possible in the OSI protocol stack, detection techniques, features of various Intrusion Detection tools and what type of attacks can be dealt with using these tools and various feasible operating system platforms.**

Index Terms—**Anomaly detection; Denial of service; Intrusion Detection; Intrusion Detection Tools; Network security; Network Attacks; Signature detection.**

I. Introduction

Suppose a strange man is standing in front of your house. He looks around, studying the surroundings, and then walks to the front door and tries to open it. The door is locked. Efforts in vain, he moves to a nearby window and gently tries to open it. It, too, is locked. It seems your house is secure. So why to install an alarm? This is a common question for intrusion detection advocates. Why bother detecting intrusions if you've installed firewalls, spam filters, and activated passwords for authenticity? The answer is simple: because intrusions still do occur!! Just as people sometimes forget to lock a window, for example, they sometimes forget to correctly update a firewall's rule set. Computer systems are still not 100 percent safe even with the most advanced protection. In fact, most computer security experts agree that, given user-desired features such as network connectivity, we'll never achieve the goal of a completely secure system. As a result, we must develop and deploy intrusion detection techniques and tools to discover and react against computer attacks.

An intrusion is a formal term describing the act of compromising a system. And detecting either failed or successful attempts to compromise the system is called an Intrusion Detection. Simple... right? In a nutshell, Intrusion detection systems or IDS do exactly as the name suggests: they detect possible intrusions. The goal of IDS tools is to detect computer attacks or illegal access, and to alert the concerned people about the detection or security breach. An IDS installed on a network can be viewed as a burglar alarm system installed in a house. Through their methods are different, both detect when an intruder/attacker/burglar is present, and both subsequently issue some type of warning signal or alert [1].

Monitor, detect, and respond to any unauthorized activity are the adages of Intrusion detection systems. Network attacks such as DoS attacks can be detected by monitoring the network traffic. There are two basic types of intrusion detection: Host-based and Network-based. Each has a distinct approach to monitoring and securing data, and each has distinct advantages and disadvantages.

Host-based intrusion detection systems (HIDS) are IDSs that operate on a single workstation. HIDS monitor traffic on its host machine by utilizing the resources of its host to detect attacks. [2]

Network-based intrusion detection systems (NIDS) are IDSs that operate as stand-alone devices on a network. NIDS monitors traffic on the network to detect attacks such as denial of service attacks; port scans or even attempts to crack into computers by monitoring network traffic [2].

The TABLE 1 shows the difference between HIDS and NIDS giving the merits and demerits of each. HIDS and NIDS can be combined to form a separate hybrid class of *Network Node IDS (NNIDS*) where agents are deployed on every host within the network being protected. A NNIDS operates much like a hybrid per-host NIDS since a single agent usually processes the network traffic directed to the host it runs upon. The main reason for introducing this type of hybrid IDS was the need to work online with encrypted networks and their data destined to the single host (only the source and destination can see decrypted network traffic).

II. Network Security Attacks

A broad definition of network security can be considered by defining its two components, security and networks. According to oxford dictionary, security is the freedom from

[1] 4th Semester, M.Tech in Computer Network Engineering, Department of Computer Science
R.V College of Engineering, Mysore Road, Bangalore–560059, Karnataka, India
[2] ISRO Telemetry Tracking and Command Network, Peenya, Bangalore – 560 058, Karnataka, India
E-Mail: karthikkr16@yahoo.com , indraa@istrac.org

danger or anxiety, no sense of threat. A computer network as we know is a group of interconnected computers. Security is described through the accomplishment of basic security properties, namely *Data confidentiality, Authentication, Access control, Data Integrity and Non-repudiation* [3].

Threats can be categorized into external or internal threats. Threats originating outside a company or an institution are external and in contrast an internal threat is one originating inside the organization. There are two types of internal threats: *Intentional attacks and Unintentional attacks.*

TABLE 1: NIDS Vs HIDS

Network based Intrusion detection systems	**Host based Intrusion detection systems**
• Resides on the computer/ application connected to a part on an organization's network and monitors network traffic on that segment looking for indication of ongoing or successful attacks. • Types of NIDS include Snort, Cisco NIDS, and Netprowler • NIDS uses a monitoring port, when placed next to a networking device like hub, switch. The port views all the traffic passing through the device. • Works on the principle of signature matching, ie comparing attack patterns to known signatures in their data base. • NIDS are suitable for medium to large scale organizations due to their volume of data and resources. So, many smaller companies are hesitant in deploying IDS. **Advantages:** • Large networks can be monitored by deploying a few devices with a good network design. • Ongoing network operat-ions won't be disrupted by deploying NIDS, since they are passive devices. • NIDSs are not susceptible to direct attack and may not be detectable by attackers. **Disadvantages:** • NIDS may fail to recognize attack when network volume becomes over-whelming. • Since many switches have limited or no monitoring port capability, some networks are not capable of providing all the data for analysis by a NIDS. • NIDS cannot analyze encrypted packets, making some of the traffic invisible to the process and reducing the effectiveness of NIDS. • Attacks involving fragm-ented or malformed packets cannot easily be detected.	• Resides on a particular computer or server, known as the host, and monitors activity only on that system looking for any malicious program running. • Types of HIDS, include Tripwire, Cisco HIDS, and Symantec ESM • Capable of monitoring system configuration data bases, such as windows registries, and stored configuration files like .ini, .cfg and .dat files. • Work on the principle of configuration and change management. An alert is triggered when file attributes change, new files created or existing files deleted. • Generally, most HIDS have common architec-tures, meaning that most host systems work as host agents reporting to a central console. **Advantages:** • Attacks that elude NIDS and local events can be detected by HIDS. • HIDS functions on the host system, where encrypted traffic will be decrypted and available for processing. • The use of switched network does not affect a HIDS. • HIDS can detect incons-istencies in the application. **Disadvantages:** • More management efforts required to install configure and manage HIDS. • Both direct attacks and attacks against the host operating system results in compromise and/or loss in functionality of HIDS. • HIDS is susceptible to some DoS attacks. • Host OS audit logs occupy large amounts of disk space and disk capacity needs to be added, which may red-uce system performance. • HIDS cannot scan /detect multi-host and non-host network devices.

Intentional attacks are malicious attacks carried out by disgruntled employees for various reasons such as, financial payment, or to cause harm to the organization. Unintentional attacks such as, deleting important data files cause unwarranted performance and financial damage to the organization. In this section we discuss some of the basic network attacks. Security threats and attacks may involve any layer of OSI stack, from physical to application layer [3].

A. Denial-of-Service (DOS) attacks

It is an attempt to forbid the authorized users from utilizing the requested service/ resource. A more advanced Distributed Denial of Service occurs when in a distributed environment the attacker sends or rather floods the server or a target system with numerous connection requests knocking the target system to the knees, leaving them no other option to restart their system. Some well known DOS attacks are:

1) *SYN Attack* where the attacker exploits the inability of the server to handle unfinished connection requests. Server is flooded with connection requests. The server crashes waiting for the acknowledgments of the requests.
2) *Ping of Death* where the attacker sends a ping request which is larger than 65,536 bytes which is the maximum allowed size for the IP, causing the system to crash or restart.

There are numerous DOS attacks cases happened in the past. The XBOX360 players suffered from DOS attacks, where the winners were kicked off the network by sore losers using a DOS attack. Worse still, hackers are selling DOS on demand to disgruntled players. That means anyone with a few bucks and a lack of morals can attack players who are simply too good at *Street Fighter IV*. 14 major websites including web sites of presidential Blue House, the defense ministry, New York Stock Exchange, the National Assembly, Shinhan Bank, the mass-circulation newspaper Chosun and the top Internet portal Naver.com came under DDoS attack originated from a small cable TV website in Seoul overloading and knocking them down.[8]

Another most heard was the Google's Self-Inflicted Denial-of-Service Attack. A Google employee, working on updating their malware notification service uploaded a simple little "/" as a malware site a few months ago (January 31, 2009), effectively declaring the entire Internet to be malware for nearly 55 minutes. Google lost a lot of money in ad revenue during those 55 minutes. In addition, Google suffered reputational losses. Google's self-inflicted denial-of-service attack is a stark reminder to all IT security professionals about what is the greatest threat and risk to operational security.

B. Eavesdropping Attacks

A form of external attack where there is an unauthorized interception of network communication and disclosure of exchanged information. This can be performed in different layers – for example, in network layer by sniffing into the exchanged packets or in physical layer by physically wiretapping the access medium.

C. Spoofing attack

The attacker impersonates an legitimate user. IP spoofing is a common example where the system is convinced that it is communicating with a trusted user and provides access to the attacker. The attacker sends a packet with an IP address of a known host by alerting the packet at the transport layer.

D. Intrusion attacks or User to Root Attack (U2R)

An unauthorized user tries to gain access to system or root through the network. Buffer overflow attack is a typical intrusion attack which occurs when a web service receives more data than it has been programmed to handle leading to data loss.

E. Logon Abuse attacks

A successful logon abuse attack would bypass the authentication and access control mechanisms and grant a user with more privileges that authorized.

F. Application-Level Attacks

The attacker exploits the weakness in the application layer – for example, security weakness in the web server, or in faulty controls in the filtering of an input on the server side. Examples include malicious software attack (viruses, Trojans, etc), web server attacks, and SQL injection.

III. Intrusion Detection Techniques

In this section we explain the intrusion detection techniques. Basically, there are two techniques in IDS: Anomaly based and Signature/Misuse based intrusion detection. Amiable, one of the main factors that should be considered while buying IDS which is whether to go for an Anomaly based or signature based detection technique. IDS vendors should be aware of the pros and cons of these techniques. We also explain the Target Monitoring and Stealth Probe techniques later in this section.

A. Anomaly based intrusion detection

First off, anomalies also known as outliers, exceptions or peculiarities are patterns in data that do not conform to a well defined notion of normal behaviour of a system [4]. The Figure 1 shows anomalies O_1, O_2 and O_3 that differ from the normal behaviour N1 and N2.

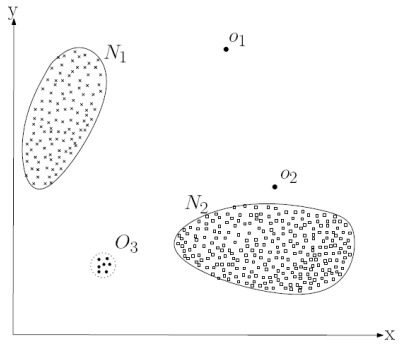

Figure 1: A simple example showing anomalies

Anomaly detection technique is designed to uncover the patterns of behavior that are far from normal and anything that widely deviates from it gets flagged as a possible intrusion. Anomaly detection can be categorized into static and dynamic [5].

In *static anomaly detector* it is assumed that a portion of the monitored system remains constant or static. The static portion of a system is composed of two parts: the system code and that portion of system data that remains constant. Static portions of the system can be represented as a binary bit string or a set of such strings (such as files). If this portion ever deviates from its original form, either an error has occurred or an intruder has altered the static portion of the system. Static anomaly detectors are said to check for *data integrity*.

In *dynamic anomaly detector* the definition of behavior is included. System behaviour is defined as a sequence (or partially ordered sequence) of distinct events. For example, audit records produced by the operating system are used by IDS to define the events of interest. In this case, the behavior can be observed only when audit records are created by OS. Events may occur in a strict sequence. More often, such as with distributed systems, partial ordering of events is more appropriate.

The system may rely on parameters that are set during initialization to reflect behaviour if it is uncertain whether behaviour is anomalous or not. Initial behaviour is assumed to be normal. It is measured and then used to set parameters that describe correct or nominal behaviour. There is typically an unclear boundary between normal and anomalous behaviour as depicted in Figure 2. If uncertain behaviour is not considered anomalous, then intrusion activity may not be detected. If uncertain behaviour is considered anomalous, then system administrators may be alerted by false alarms/ when there is no intrusion. [5]

Figure 2: Behaviour distinguished Anomalous from normal.

The most common way to draw this boundary is with statistical distributions having a mean and standard deviation. Once the distribution has been established, a boundary can be drawn using some number of standard deviations. If an observation lies at a point outside of the (parameterized) number of standard deviations, it is reported as a possible intrusion.

A dynamic anomaly detector defines an "actor", as the potential intruder. An actor is frequently defined to be a specific user, with an account. Alternatively, user or system processes are monitored. The mapping between processes, accounts, and users is only determined when an alert is to be raised. In most operating systems there is clear traceability from any process to the user/account for which it is acting. Likewise, an operating system maintains a mapping between a process and the physical devices in use by that process.

Anomaly based intrusion detection is useful for detecting attacks like:

1) Misuse of Protocol and Service Ports

Features of the standard protocols can sometimes be misused or modified by an attacker in order to tunnel through a firewall. Installation of backdoor services on well-known standard ports is another common misuse of service ports.

2) DoS Based on Crafted Payloads

When a malicious intruder creates an attack using a crafted

IP packet, the resulting Denial of Service (DoS) can occur on the network bandwidth, CPU cycles, memory resources, or application process/programs. Examples of this type of DoS include process table exhaustion, IP stack crashing, or a Web application .soft spot. The impact of this DoS attack would be an anomaly in service quality.

3) DoS Based on Volume (DDoS)

Anomaly based intrusion detection is the only reliable means for detection in the case of the DoS attack that floods the network with a large volume of traffic. This is because sophisticated attack traffic may not be distinguishable from regular network traffic on a per packet basis and the attack does not manifest a specific signature that can be captured by signature-based mechanisms. For example, the following traffic pattern anomalies can be observed as a result of the Distributed Denial of Service (DDoS): TCP control packet statistics for TCP SYN flood or relative volumes of TCP, UDP, and ICMP traffic for UDP or ICMP flood.

4) Buffer Overflow

The buffer overflow is the most common vulnerability exploited by attackers. Buffer overflow with shellcode execution is the most serious form of this exploit because a successful attack can result in arbitrary program execution on the victim system(s). Many exploited fields, such as user passwords for FTP, are supposedly made of printable ASCII characters based on the standard Request For Comments (RFCs) by the Internet Engineering Task Force (IETF). Excessive non-printable ASCII characters are anomalies of strong suspicion. Furthermore, shellcode embedded in these fields are sure signs of malicious intent.

5) Other Natural Network Failures

Failures in routers/switches can result in changes in traffic pattern observed at certain points of the network. This can be in the form of sudden drop in the volume of traffic due to broken connections, or in the form of traffic shift from one link to another due to traffic rerouting as a recovery action. All these changes are noteworthy and can be detected as traffic anomalies.

Anomaly Detection Techniques represents a broad spectrum of detection techniques. One can define profiles in terms of simple thresholds or more complex statistical distributions; and profiles can be self-learned or manually set, adaptive, or static [6].

Three broad types of anomaly based detection techniques are discussed in the following paragraphs.

1) Protocol Anomaly Detection

As mentioned earlier protocol anomaly refers to all exceptions related to protocol format and behaviour with respect to common practice on the Internet and standard specifications. This includes network and transport layer protocol anomalies in layers 3-4 and application layer protocol anomalies in layers 6-7. Unusual conditions are checked for in the process of IP defragmentation, TCP reassembly. When the IDS is inline, many exceptions leading to ambiguous interpretation by the end host can be averted.

When an IDS is monitoring application protocol behaviour, it must be able to perform deep application protocol parsing, which is also known as decoding.

The following anomalies are examples of protocol anomalies that could be detected when application protocol behaviour is being observed:

i. Illegal field values and combinations
ii. Illegal command usage
iii. Unusually long or short field lengths, which can indicate an attacker is attempting to introduce a buffer overflow
iv. Unusual number of occurrences of particular fields/commands
v. Running a protocol or service for a non-standard purpose or on a non-standard port

2) Application Payload Anomaly

Application anomaly must be supported by detailed analysis of application protocols to define accurate behaviour constraints for them. Application anomaly also requires understanding of the application semantics in order to be effective. One needs to know what type of encoding is legal for a given field, and what other applications can be embedded within it. One good example of application level anomaly is the presence of shellcode in unexpected fields. A reliable anomaly profile allows shellcode execution attacks to be detected without knowing what particular exploit code is involved, or even the existence of exploit code.

3) Statistical Anomaly Based Intrusion Detection

A normal TCP traffic follows a well-defined three-way handshake process for connection setup, data transfer phase, and then completes with the connection tear down. There is a stable balance among different types of TCP packets in the absence of attacks which is compared against short-term observations that will be affected by attack events. Statistical anomaly based IDS captures this behaviour and differentiates between the long term and short term observations in a given protected environment to avoid generating false alarms on normal traffic variations.

Profiles based on statistical measures could raise DDoS anomalies based on rare events of the difference between the long and short-term distributions or based on a rare occurrence of long bursts of high-rate traffic. A well-designed system would allow the user to set a sensitivity level to reflect how tolerant their network or servers are to traffic surge. The lower the sensitivity level, the more severe the traffic profile deviation must be before the algorithm raises a DDoS alarm. The normal profiles are continuously learned while the system is in detection mode, with safeguard against statistics poisoning under attacks. This allows the anomaly profiles to adapt to typical environmental changes that occur in an organization [6]. For example, some of the events that can be detected include: SYN Flood attacks, UDP Flood attacks, ICMP Flood attacks, TCP data segment flood attacks.

Advantages:

i. IDSs based on anomaly detection detect unusual behaviour and thus have the ability to detect symptoms of attacks without specific knowledge of details.
ii. Anomaly detectors can produce information that can in turn be used to define signatures for misuse detectors.

Disadvantages:

i. Anomaly detection approaches usually produce a large number of false alarms due to the unpredictable behaviours of users and networks.

ii. Anomaly detection approaches often require extensive "training sets" of system event records in order to characterize normal behaviour patterns.

B. Misuse/Signature Based Intrusion Detection

The second major category of IDS is known as misuse detection also referred to as signature-based detection because alarms are generated based on specific attack signatures. These attack signatures encompass specific traffic or activity that is based on known intrusive activity.

The following are the two techniques in misuse detection:

1) Expression matching

The simplest form of misuse detection is expression matching, which searches an event stream (log entries, network traffic, or the like) for occurrences of specific patterns/signatures. A simple example would be "^GET[^$]*/etc/passwd$" - this checks for something that looks like an HTTP request for the Unix password file. Signatures can be very simple to construct, however especially when combined with protocol-aware field decomposition.

2) State transition analysis

State transition analysis models attacks as a network of states and transitions (matching events). Every observed event is applied to finite state machine instances (each representing an attack scenario), possibly causing transitions. Any machine that reaches its final (acceptance) state indicates an attack as depicted in Figure 3

This approach allows complex intrusion scenarios to be modelled in a simple way, and is capable of detecting slow or distributed attacks, but may have difficulty expressing elaborate scenarios.

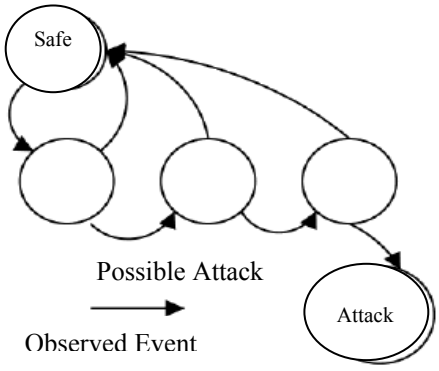

Fig 3: Schematic Structure of a State Machine

Advantages:

i. Misuse detectors are very effective at detecting attacks without generating an overwhelming number of false alarms.

ii. Misuse detectors can quickly and reliably diagnose the use of a specific attack tool or technique. This can help security managers prioritize corrective measures and track security problems on their systems.

Disadvantages:

i. Misuse detectors can only detect those attacks they know about therefore they must be constantly updated with signatures of new attacks.

ii. Many misuse detectors are designed to use tightly defined signatures that prevent them from detecting variants of common attacks. State-based misuse detectors can overcome this limitation, but are not commonly used in commercial IDSs.

C. Target Monitoring

Any change or modification in the target objects are reported by the Target Monitoring Systems. This is usually done through cryptographic algorithm that computes a cryptochechsum for each target file [7]. Changes such as file modification or program logon which would cause changes in the cryptochecksum are reported by the IDS. This type of system is the easiest to implement, because it does not require constant monitoring by the administrator. Integrity checksum can be computed at whatever intervals you wish, and on either all files or just the mission/system critical files.

Tripwire software will perform target monitoring using cryptochecksum by providing instant notification of changes to configuration files and enabling automatic restoration.

D. Stealth Probes

Stealth probes collects and correlate data to try to detect attacks made over long period of time, often referred to as "low and slow" attacks [7]. Attackers, for example, will check for system vulnerabilities and open ports over a two-month period, and wait another two months to actually launch the attacks. They take a wide-area sampling and attempt to discover any correlating attacks.

IV. TOOLS FOR IDS

The wide array of intrusion detection products available today (freely available of commercial) addresses a range of organizational security goals and considerations. We have provided a list of most common IDS tools [9] describing their features. TABLE 2 gives the comparisons of IDS tools.

SNORT - This lightweight network intrusion detection and prevention system excels at traffic analysis and packet logging on IP networks. It detects threats, such as buffer overflows, stealth port scans, CGI attacks, SMB probes and NetBIOS queries, NMAP and other port scanners and DDoS clients, and alerts the user about them. It develops a new signature to find vulnerabilities. It records packets in their human-readable form from the IP address.

OSSEC – HIDS – It is scalable, multi-platform, open source Host-based Intrusion Detection System (HIDS). It has a powerful correlation and analysis engine, integrating log analysis; file integrity checking; Windows registry monitoring; centralized policy enforcement; rootkit detection; real-time alerting and active response.

FRAGROUTE – It is a one-way fragmenting router - IP packets get sent from the attacker to the Fragrouter, which transforms them into a fragmented data stream to forward to the victim. Fragrouter helps an attacker launch IP-based attacks while avoiding detection.

METASPLOIT - It is an advanced open-source platform for developing, testing, and using exploit code. It ships with hundreds of exploits, as you can see in their online exploit building demo. This makes writing your own exploits easier, and it certainly beats scouring the darkest corners of the Internet for illicit shell code of dubious quality.

TRIPWIRE – It Detects Improper Change, including

additions to, deletions from and modifications of file systems and identifies the source. It Simplifies and Eases Management of Change Monitoring Policies.

V. Conclusion

IDS tools are becoming the need for the day and for security not only in the corporate world but also for network users. Security incidents are becoming more and more common and measures have to be taken to curb such incidents. Some vendors are selling heuristic detection systems where artificial intelligence (AI) is used to detect intrusions.

A more effective hybrid class of IDS called *Network Node IDS (NNIDS*) where agents are deployed on every host within the network being protected.

However, there are still many challenges to overcome. Improving, mining, and reducing intrusion detection data are critical to dealing with multisensory architectures of the future. Fast and flexible detection techniques are necessary to identify the vast variety of clever and unusual attacks we will undoubtedly encounter. Finally, cooperation with not only other IDS but also other network security components is mandatory to achieving a holistic network security posture for organizations of the future.

Acknowledgement

I would like to dedicate my work to my guide Smt. A. Indra without whom this paper would be incomplete. I would also like to dedicate this paper to all my friends, especially to my best friends Pranu (Timlee) and Shoeb, Karthika(KD), Kousi, and Santosh for their keen interest and support. I also thank my parents for their unlimited support.

References

[1] Paul Innella and Oba McMillan, Tetrad Digital Integrity, LLC "An Introduction to Intrusion Detection Systems" December 6, 2001
[2] Micheal E. Whitman and Herbert J. Mattord, "Principles of Information Security" page 289-294
[3] Christos Douligeris and Dimitrios N. Serpanos "Network Security Current Status and Future Trends"
[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar"Anomaly Detection: A Survey" August 15, 2007
[5] Anita K. Jones and Robert S. Sielken, Department of Computer Science University of Virginia "Computer System Intrusion Detection: A Survey"
[6] Dr. Fengmin Gong, Chief Scientist, McAfee Network Security Technologies Group, "Deciphering Detection Techniques: Part II Anomaly-Based Intrusion Detection" March 2003
[7] Carl Endorf, Eugene Schultz, Jim Mellander "Intrusion detection & prevention" page 18
[8] Deccan Herald, "Cyber attacks cripple websites", http://www.deccanherald.com/content/12577/cyber-attacks-cripple-websites.html
[9] "Top 5 Intrusion Detection Systems", http://sectools.org/ids.html

TABLE 2: Comparison of IDS Tools

Features / Tools	HIDS	NIDS	ATTACKS DETECTED / CONDUCTED	HUMAN-COMPUTER INTERFACE	LICENCE	PLATFORM SUPPORTED
SNORT	No	Yes	DOS and CGI Attacks, Intrusion attacks, Port Scans, SMB probes Layer 3 and above attacks.	GUI/ Command Line	Open Source	Linux, Windows, Free BSD, MAC OS
OSSEC HIDS	Yes	No	Attempts to access non-Existent files Secure Shell Attacks, FTP Scans, SQL Injections, File system attacks	GUI	Open Source	Linux, Windows, Free BSD, MAC OS
FRAGROUTE	NO	Yes	Insertion, Evasion, and Denial of Service	Command Line	Open Source	Linux, Free BSD
METASPLOIT	No	Yes	Vulnerability Exploitation	Command Line	Open Source	Linux, Windows, Free BSD, MAC OS
TRIPWIRE	Yes	No	Root Kit Detection, File Integrity Checks	Command Line	Open Source	Linux, Windows, Free BSD, MAC OS