4.1 IDS Concepts

There are six basic approaches to intrusion-detection and prevention. Some of these methods are implemented inside various software packages, and others are simply strategies that an organisation can employ to decrease the likelihood of a successful intrusion.

Historically, when IDSs were first developed, hubs were used very frequently. Today, switches are used rather than hubs. With a hub, after a packet has travelled from its source network to the destination network (being routed by its destination IP address), it finally arrives at the network segment on which the target is located. After it gets to that final segment, the MAC address is used to find the target. All the computers on that segment can see the packet, but because the destination MAC address does not match the MAC address of their network card, it ignores the packet.

At some point, enterprise individuals realized that if they simply chose not to ignore packets not destined for their network card, they could see all the traffic on the network segment. In other words, one could look at all the packets on that network segment. Thus, the packet sniffer was born. After that it was simply a matter of time before the idea came about of analysing those packets for indications of an attack, thereby giving rise to intrusion-detection systems.

4.1.1 Pre-emptive Blocking

Pre-emptive blocking seeks to prevent intrusions from happening before they occur. This is done by observing any danger signs of imminent threats and then blocking the user or IP address from which these signs originate. Examples of this technique include attempts to detect the early Footprinting stages of an imminent intrusion, then blocking the IP or user that is the source of the Footprinting activity. If you find that a particular IP address is the source of frequent port scans, and other scans of your system, then you would block that IP address at the firewall.

This type of intrusion detection and avoidance can be quite complicated in which could potentially block a legitimate user by mistake. The complexity arises from distinguishing legitimate traffic from that indicative of an impending attack. This can lead to the

problem of false positives, in which the system mistakenly identifies legitimate traffic as some form of attack.

Usually, a software system will simply alert the administrator that suspicious activity has taken place. A human administrator will then make the decision whether or not to block the traffic. If the software automatically blocks any addresses it deems as suspicious, you run the risk of blocking out legitimate users. It should also be noted that nothing prevents the offending user from moving to a different machine to continue the attack. This sort of approach should only be one part of an overall intrusion-detection strategy, and not the entire strategy.

4.1.2 Anomaly Detection

Anomaly detection involves actual software that works to detect intrusion attempts and to then notify the administrator. This is what many people think of when they talk about intrusion-detection systems. The general process is simple: The system looks for any abnormal behaviour. Any activity that does not match the pattern of normal user access is noted and logged. The software compares observed activity against expected normal usage profiles. Profiles are usually developed for specific users, groups of users, or applications. Any activity that does not match the definition of normal behaviour is considered an anomaly and is logged. Sometimes, we refer to this as “trace back” detection or “trace back” process. We are able to establish from where this packet was delivered. The specific ways in which an anomaly is detected include:

- Threshold monitoring
- Resource profiling
- User/group work profiling
- Executable profiling

4.1.2.1 Threshold Monitoring

Threshold monitoring pre-sets acceptable behaviour levels and observes whether these levels are exceeded. This could include something as simple as a finite number of failed login attempts or something as complex as monitoring the time a user is connected and the amount of data the user downloads. Thresholds provide a definition of acceptable behaviour. Unfortunately, characterizing intrusive behaviour only by the threshold limits can be somewhat

challenging. It is often quite difficult to establish proper threshold values or the proper time frames at which to check those threshold values. This can result in a high rate of false positives in which the system misidentifies normal usage as a probable attack.

4.1.2.2 Resource Profiling

Resource profiling measures the system-wide use of resources and develops a historic usage profile. Looking at how a user normally utilizes system resources enables the system to identify usage levels that are outside normal parameters. Abnormal readings can be indicative of illicit activity underway. However, it may be difficult to interpret the meaning of changes in overall system usage. An increase in usage might simply indicate something benign like an increased workflow rather than an attempt to breach security.

4.1.2.3 User/Group Work Profiling

In user/group work profiling, the IDS maintains individual work profiles about users and groups. These users and groups are expected to obey these profiles. As the user changes his activities, his/her expected work profile is updated to reflect those changes. Some systems attempt to monitor the interaction of short-term versus long-term profiles. The short-term profiles capture recent changing work patterns, whereas the long-term profiles provide a view of usage over an extended period of time. However, it can be difficult to profile an irregular or dynamic user base. Profiles that are defined too broadly enable any activity to pass review, whereas profiles that are defined too narrowly may inhibit user work.

4.1.2.4 Executable Profiling

Executable profiling seeks to measure and monitor how programs use system resources, paying particular attention to those whose activity cannot always be traced to a specific originating user. For example, system services usually cannot be traced to a specific user launching them. Viruses, Trojan horses, worms, trapdoors, and other software attacks are addressed by profiling how system objects such as files and printers are normally used, not only by users, but also by other system subjects on the part of users. In most conventional systems, for example, any program, including a virus inherits all of the privileges of the user executing the software. The software is not

limited by the principle of least privilege, but to only those privileges needed to properly execute. This openness in the architecture permits viruses to covertly change and infect totally unrelated parts of the system.

Executable profiling enables the IDS to identify activity that might indicate an attack. Once a potential danger is identified, the method of notifying the administrator, such as by network message or e-mail, is specific to the individual IDS.