Intrusion Detection Categories

(note supplied by Steve Tonkovich of CAPTUS NETWORKS)

Intrusion Detection systems fall into two broad categories and a single new one. All categories have both signature and anomaly bases detection systems. These are:

· Network based: Most of these systems are placed on the network, nearby the system or systems being monitored. They examine the network traffic and determine whether it falls within acceptable boundaries. Only the CaptIO NIDS is placed as pass through device.

· Host based: These types of systems actually run on the system being monitored. These examine the system to determine whether the activity on the system is acceptable.

· Traffic Limiting: We are creating a new category of Intrusion Detection System called Traffic Limiting or TLIDS. This category would be a hybrid of a Network Based IDS and the ability for the IDS to take action against this traffic by being a part of the network (pass through). This method of detection allows the device to have a lower chance of alarming on false positives and greater ability to handle legitimate traffic peaks.

Network Based Intrusion Detection System (NIDS)

Network based intrusion detection systems are those that monitor traffic on the entire network segment. A network interface card (NIC) can operate in one of two modes in a NIDS, these being:

· Normal mode: all packets seen on the ethernet segment are relayed through (passed through) to the destination system. This requires the NIDS to be an active network device and participate in the flow of traffic.

· Promiscuous mode: all packets seen on the ethernet segment are relayed to the host system via a span port or probe. NIDS working in promiscuous mode must be passive and cannot take action on that network segment. They would require another active interface for management and control.

A network card can normally be switched from normal mode to promiscuous mode, and vice-versa, by using a low-level function of the operating system to talk directly to the network card to make that change.

Key differences:

· The CaptIO works in "normal mode". This allows the CaptIO to reduce overhead and increase performance and security.

· All current Network based intrusion detection systems require that the NIC be run in "promiscuous mode".

· So even though the ISS RealSecure sensor can run on a Nokia platform, which is a pass through device, it is really just using the Nokia as a host and running a "promiscuous mode" interface.

· NFR & Snort are software only that runs on other vendor's operating systems and hardware, ISS normally as well. This in itself causes increased security risks since you now have to be involved in patching the vendor's operating system every time a new exploit is found. This could mean everyday you would be applying operating system patches to your NIDS and remote sensors!

· The systems listed above are useful tools that should be part of a larger security policy but act more as reactive analysis tools producing complex and lengthy log files and alarms that need to be reviewed by the network manager before any real action is taken. This will be discussed more in the "Network Based Intrusion Detection Functionality" section.

· Another point to consider is the use of switches, rather than hubs, in a network. Note that packets received on one interface of a switch are not always sent to other interfaces of the same switch. For this reason, a heavily switched environment, rather than a single shared segment, will often defeat the use of these passive based NIDS. In this instance the use of a span port or port mirroring must be used to analyze all traffic on that segment. This uses precious resources on core routing and switching gear. The CaptIO being a pass through device does not put any additional load on core equipment rather helps reduce the load on all equipment except the WAN router. (at this time)

Packet Sniffers and Network Monitors (probes)

Packet Sniffers and Network Monitors were originally designed to aid in the process of monitoring the traffic on an Ethernet network. The first of these were Novell LANalyser, Network General Sniffer and Microsoft Network Monitor.

Sniffers and Probes allow:

· Packets to be counted and monitored. Counting the packets that come past, and adding together their total size over a period of time (including overheads such as packet headers) gives a pretty good indication of how heavily loaded the network is. Most sniffers provide load graphs or meters to show the relative load of the network. Packets can be examined in detail. For example, you might want to capture a set of packets arriving at a web server to diagnose some problem with the server.

Unfortunately, from a security point of view, a packet Sniffer is of limited benefit. The problem with packet sniffers today is that they still rely on promiscuous monitoring of network traffic and cannot take action against real time traffic.

They are more of a monitoring and reactive troubleshooting tool. The task of capturing every packet on the network, disassembling it and manually taking action based on the contents of the packet is far too time-consuming, even for a horde of specially trained network gnomes. Much

like the logs and alarms produced by the competitive NIDS.

One final word about packet sniffers: These tools can be used to do evil as well as good. For example, packet sniffing can be used to find out someone's Unix password by sniffing telnet packets to the machine that they connect to. Many Unix operating systems also contain programs that allow a super user to monitor all traffic on that segment so once a system has been compromised it is very difficult to stop the intruder. This is a very good example of where host based intrusion detection would come into play.

Competitive NIDS: (definitions from vendor web sites)

Internet Security Systems:
RealSecure Network Sensor runs on dedicated workstations to provide unmatched network intrusion detection and response. Each RealSecure Network Sensor closely watches for attack signatures - the early indications that an attempted intrusion is in progress - as traffic travels over segments of your network.

RealSecure's customizable functions allow you to determine how RealSecure should respond when it detects suspicious activity. It can terminate the connection, send email or pager alerts, record the session, reconfigure select firewalls or taking other, user-directed actions. In addition, RealSecure Network Sensor can send an alarm to the RealSecure Manager or a third-party management console for immediate administrative attention and follow-up.

http://www.iss.net/

NFR:
NFR Network Intrusion Detection (NFR NID) is an intrusion detection system that unobtrusively monitors networks in real time for activity such as known attacks, abnormal behavior, unauthorized access attempts and policy infringements. Information associated with activity that may be suspicious is recorded and alerts raised as necessary.

Suspicious activity can be categorized into misuse and anomalies. Misuse is a known attack, like a hacker trying to break into your e-mail server. An anomaly is something out of the ordinary, like abnormally heavy traffic.

Misuse intrusion detection works by comparing network traffic to known attack patterns called "signatures." Anomaly detection systems "learn," or are told, what constitutes "normal" behavior, developing sets of models that are continually updated. Activity is then compared against these models.

NFR NID provides both misuse and anomaly detection.

http://www.nfr.com/

Snort:
Snort is a lightweight network intrusion detection system capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and more. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plugin architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient.

Snort has three primary uses. It can be used as a straight packet Sniffer like tcpdump(1), a packet logger (useful for network traffic debugging, etc), or as a full blown network intrusion detection system.

Snort logs packets in either tcpdump(1) binary format or in Snort's decoded ASCII format to logging directories that are named based on the IP address of the "foreign" host.

http://www.snort.org/

Network Based Intrusion Detection Functionality:

Although ISS, Snort and NFR claim to be NIDS, they would be better defined as a NIDS looking for host exploits, where Captus is a NIDS/TLIDS looking for network and DoS attacks.

All of the current NIDS are passive and reactive devices. They require sensors to be located on several different segments and then report back to a separate management tool.

The current competitive NIDS can:

· examine packets that pass through the network. (Promiscuous mode)

· for legitimate packets, allow them to pass, perhaps recording them for future analysis.

· only report back to a central management station used to parse the log files generated and create a tool for the network manager to react as necessary.

· (ISS Only) Where a packet endangers the security or integrity of a target system (host based intrusion), stop transmission of the packet by sending TCP "connection closed" or ICMP "port unreachable" messages to both the target system and the system sending the packet.

The problem here is that they cannot keep the internal network from being flooded since they are a passive device. The CaptIO can dynamically write packet filtering rules to stop this traffic since it is in-line with the network.

· If the source of the attack is spoofed, the ISS defense mechanism will again fail since they are trying to use IP response messages to communicate with the source.

If the CaptIO does not see a decrease in the source traffic or protocol, then it can take an action to filter out traffic from the source or protocol, keeping the backend servers and core network from having to use it's resources to process these requests. Then once the attack or spike of traffic subsides, automatically remove these filters to again allow all traffic to flow through.

· In this manner, RealSecure can perform an effective second layer defense for a target system when hosted behind a firewall or device like the CaptIO. This would be a good overall "security solution" for companies looking for both DoS and host based intrusion detection.

· (All NIDS) Monitor the network for obvious port scans. Before compromising a system, a cracker will often port scan the system to determine what vulnerabilities might exist. Accessing a web server host on the web server port (80) might be seen as a relatively harmless activity, but some access attempts are in fact deliberate attacks, or attempts at attacks.

For example, an access that looks like "GET /../../../etc/passwd HTTP/1.0" is probably a bad sign, and should be blocked. Identify IP spoofing attempts of various sorts. This is someone trying to look at the password file so they can attempt to login to the system.

· The ARP protocol that is used to convert IP addresses to MAC addresses is often a target for attack. By sending forged ARP packets over an ethernet, an intruder who has obtained access to one system can also "pretend" to be operating as a different system. This can lead to denial of service attacks of various sorts, as well as system hijacking, whereby an important server (such as a DNS server or authentication server) is "spoofed". Crackers can use this "spoofing" to redirect packets to their own system, and perform "man in the middle" type attacks on what would otherwise be a secure network. By keeping a register of ARP packets, a network based intrusion detection system can identify the source (ethernet address) of a compromised system and flush out would-be crackers.

The CaptIO would prevent this type of attack if the DNS servers were sitting off of a CaptIO. We could not prevent this from happening at another service provider if no CaptIO devices were installed.

· (ISS Only) When this unwanted activity is detected, network based intrusion detection can take action, including interfering with future traffic from the intruder, or reconfiguring a nearby OPSEC compliant firewall to block all traffic coming from the intruder's computer or network.

Again, this is being considered NIDS but is really looking into the payload portion of the packet for host based exploits. The CaptIO does not look

into the payload portion of the packet. This again is how we can process information at wire speed and still detect network anomalies.

Also, we have not found any customers to be actually implementing their ISS sensors in this manner. The rate of false positives is very high and would compromise legitimate traffic.

Host Based Intrusion Detection Functionality:

Once a network packet has arrived at the host that it was intended for, there is still available a third line of defense behind the firewall and NIDS. This is called "host based intrusion detection", and comes in several flavors.

The two main types of host based intrusion detection are:

· Network monitors. These monitor incoming network connections to the host, and attempt to determine whether any of these connections represent a threat. Network connections that represent some kind of intrusion attempt are acted on. Note that this is different to network based intrusion detection, as it only looks at network traffic coming to the host it is running on, and not all traffic passing the network. For this reason it does not require promiscuous mode on the network interface.

· Host monitors. These monitor files, file systems, logs, or other parts of the host itself to look for particular types of suspicious activity that might represent an intrusion attempt (or a successful intrusion). Systems administration staff can then be notified about any problems that are found.

Sample of Host Based IDS Vendors:

· ISS
· NFR
· Snort
· Recourse
· Tripwire
· Fcheck
· AIDE
· OpenWall
· LIDS

· Watch certain mailing lists (such as BUGTRAQ) for updates.

Anomaly Detection vs. Signature Based:

This excerpt was taken from the 8th USENIX Security Symposium, August 23, 1999. Although it is almost 2 years old it does a good job explaining the differences between anomaly detection and signature based (misuse detection). The Captus line of products uses anomaly detection based on predefined thresholds like the article below describes.

The biggest draw back of anomaly detection is the high rate of false positives. The drawback of signature based systems is that they do not learn new traffic patterns and have to be updated continuously. This leaves the customer network vulnerable and allows a large number of attacks to go undetected.

The Captus products have now introduced a multi dimensional threshold based system that allows it to take actions against these traffic flows that are outside "normal" traffic patterns. With these actions we can validate whether or not we are receiving real traffic from non-spoofed sources. Based on the response to these actions we can then determine if this is an attack or spike in traffic. This will minimize false positives and allows us to dynamically stop malicious attacks before they reach the core of the customer network. No other vendors today have this ability.

Begin Usenix article:

Intrusion detection techniques are generally classified into two categories: anomaly detection and misuse detection (signature based). Anomaly detection assumes that misuse or intrusions are highly correlated to abnormal behavior exhibited by either a user or the system. Anomaly detection approaches must first baseline the normal behavior of the object being monitored, then use deviations from this baseline to detect possible intrusions (CaptIO Thresholds).

Anomaly detection approaches have been implemented in expert systems that use rules for normal behavior to identify possible intrusions, in establishing statistical models for user or program profiles, and in using machine learning to recognize anomalous user or program behavior.

Misuse detection techniques attempt to model attacks on a system as specific patterns, then systematically scan the system for occurrences of these patterns. This process involves a specific encoding of previous behaviors and actions that were deemed intrusive or malicious. The earliest misuse detection methods involved off-line analysis of audit trails normally recorded by host machines. For instance, a security officer would manually inspect audit trail log entries to determine if failed root login attempts were recorded. Manual inspection was quickly replaced by automated analysis tools that would scan logs based on specific patterns of intrusion. Misuse detection approaches include expert systems, model-based reasoning, state transition analysis, and keystroke dynamics monitoring. Today, the vast majority of commercial and research intrusion detection tools are misuse detection tools that identify attacks based on attack signatures.

It is important to establish the key differences between anomaly detection and misuse detection approaches. The most significant advantage of misuse detection approaches is that known attacks can be detected fairly reliably and with a low false positive rate. Since specific attack sequences are encoded into misuse detection systems, it is very easy to determine exactly which attacks, or possible attacks, the system is currently experiencing. If the log data does not contain the attack signature, no alarm is raised. As a result, the false positive rate can be reduced very close to zero. However, the key drawback of misuse detection approaches is that they cannot detect novel attacks against systems that leave different

signatures. So, while the false positive rate can be made extremely low, the rate of missed attacks (false negatives) can be extremely high depending on the ingenuity of the attackers. As a result, misuse detection approaches provide little defense against novel attacks, until they can learn to generalize from known signatures of attacks.

Anomaly detection techniques, on the other hand, directly address the problem of detecting novel attacks against systems. This is possible because anomaly detection techniques do not scan for specific patterns, but instead compare current activities against statistical models of past behavior. Any activity sufficiently deviant from the model will be flagged as anomalous, and hence considered as a possible attack. Furthermore, anomaly detection schemes are based on actual user histories and system data to create its internal models rather than pre-defined patterns. Though anomaly detection approaches are powerful in that they can detect novel attacks, they have their drawbacks as well. For instance, one clear drawback of anomaly detection is its inability to identify the specific type of attack that is occurring. However, probably the most significant disadvantage of anomaly detection approaches is the high rates of false alarm. Because any significant deviation from the baseline can be flagged as an intrusion, non-intrusive behavior that falls outside the normal range will also be labeled as an intrusion - resulting in a false positive. Another drawback of anomaly detection approaches is that if an attack occurs during the training period for establishing the baseline data, then this intrusive behavior will be established as part of the normal baseline. In spite of the potential drawbacks of anomaly detection, having the ability to detect novel attacks makes anomaly detection a requisite if future, unknown, and novel attacks against computer systems are to be detected.

End Usenix article:

References:

Anup K. Ghosh & Aaron Schwartzbard: A Study in Using Neural Networks for Anomaly and Misuse Detection. http://www.usenix.org/