Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems

Harley Kozushko

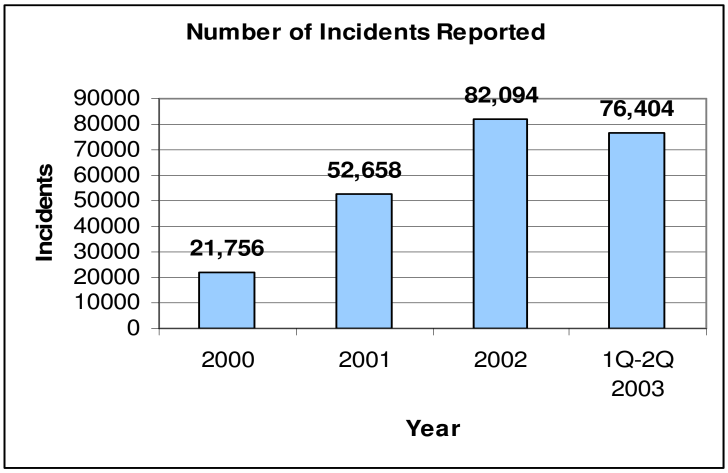
Thursday, September 11, 2003

Independent Study

Because the last few years have seen a dramatic increase in the number of attacks, intrusion detection has become the mainstream of information assurance. While firewalls do provide some protection, they do not provide full protection and still need to be complimented by an intrusion detection system. The purpose of intrusion detection is to help computer systems prepare for and deal with attacks. Intrusion detection systems collect information from a variety of sources within computer systems and networks. For most systems, this information is then compared to predefined patterns of misuse to recognize attacks and vulnerabilities. However, there are new techniques of intrusion detection including the use of support vectors and neural network machines. These techniques, along with behavioral data forensics, create a database of normal user behavior and will alert the security officer if a deviation from that normal behavior occurs. In the majority of intrusion detection systems, however, both network and host-based intrusion detection systems combine to deal with attack detection and prevention from both inside and outside sources. Still, the intrusion detection system itself has an inherent risk attributed to it because of the absence of human intervention in some response scenarios.

Because of the rapidly increasing network technology there is an increased need for security of that technology. As a result, intrusion detection has become an important technology market. According to industry estimates, the market for intrusion detection systems grew from $40 million in 1997 to $100 million in 1998. This growth was driven by reports of increasing security breaches. Graph 1 indicates a disturbing increase in the number of incidents reported from 2000 through the 2nd quarter of 2003. However, this market is low compared to the cost of malicious code, as Chart 1 describes. And as graph 2 indicates, vulnerabilities are also on the rise, with an alarming increase over the past few years, and the first and second quarters of 2003. With the costs of damages combined with the increasing possibility of intrusion, there is a great need for intrusion detection systems.

Graph 1: Number of Incidents Reported



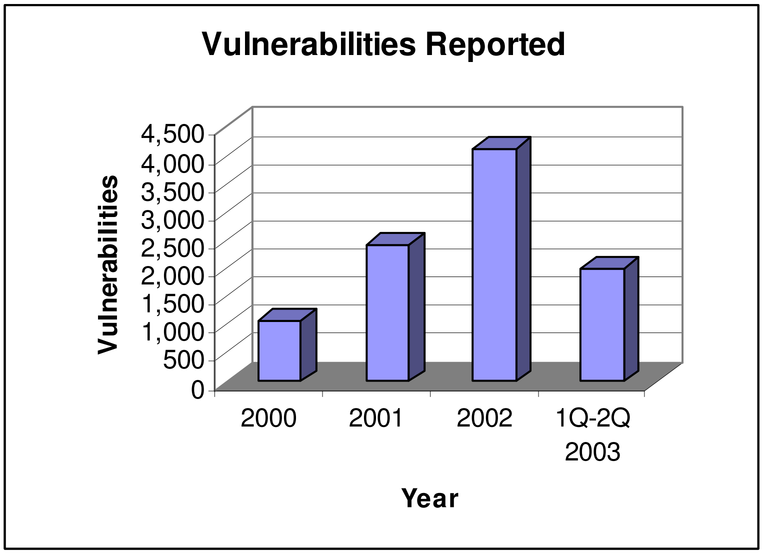
Source: Computer Emergency Response Team (CERT

Chart 1: Economic Impact of Malicious Code

Year	Code Name	Worldwide Economic Impact ($ US)	Cyber Attack Index
2001	Nimda	$635 Million	0.73
2001	Code Red	$2.62 Billion	2.99
2001	SirCam	$1.15 Billion	1.31
2000	Love Bug	$8.75 Billion	10
1999	Melissa	$1.10 Billion	1.26
1999	Explorer	$1.02 Billion	1.17

Source: Computer Economics Inc.

Graph 2: Vulnerabilities are up



Source: Computer Emergency Response Team (CERT)

Because the number of attacks and vulnerabilities are rising, network administrators are looking to extend firewalls. Intrusion detection is considered by many to complement network firewalls, extending the security management capabilities of system administrators to include security audit, monitoring, attack recognition, and response. However, a common question is how exactly intrusion detection complements firewalls. One way of characterizing the difference is provided by classifying security violation by source. That is, whether security violations come from outside of the network or from within. Firewalls act as a barrier between the network, which is internal to the company, and the outside world. Filtering incoming traffic according to a security policy creates this barrier.

This would be a sufficient protection if it weren't for these facts:

1. Not all access to the Internet occurs through the firewall. Users, for various reasons ranging from ignorance to impatience, sometimes set up unauthorized modem connections between their systems that are connected to the network and outside Internet service providers. The firewall cannot mitigate risk associated with connections it never sees.

2. Not all threats originate outside of the firewall. The vast majority of loss due to security breaches is traced to inside the company. Again, the firewall only sets barriers between the internal network and the Internet. If the traffic reflecting security breaches never passes the firewall, it cannot detect the problem.

3. Firewalls are subject to attacks themselves. Attack strategies for circumventing firewalls have been widely publicized since the first firewalls were fielded. A common strategy is to use tunneling to bypass firewall protections. Tunneling is the practice of encapsulating a message in one protocol that might be blocked by firewall filters, inside a second message.

It is because of these three facts that intrusion detection systems are needed even though a firewall may already be in place. There are two types of intrusion detection systems: host-based and network-based. However, there are many differences between the two. While their roots are similar, their operational use is different. Intrusion detection is based on analyzing a set of discrete, time-sequenced events for patterns of misuse. Intrusion detection sources both network-based and host-based, are sequential records that reflect specific actions and indirectly reflect behavior. Host-based technology examines events like what files were accessed and what applications were executed. Network-based technology examines events as packets of information exchange between computers (network traffic).

There are two types of network-based intrusion detection technologies. Promiscuous-mode network intrusion detection is the traditional technology that "sniffs" all the packets on a network segment for analysis. Promiscuous-mode systems place a single sensor on each segment. Network-node intrusion detection systems sniff just the packets bound for a single destination computer. Network-node systems are characterized by a set of distributed agents on mission critical machines. Both host and network technologies are necessary for comprehensive intrusion detection, but each has advantages and disadvantages that should be measured against the requirements for the target machine.

However, first it is important to define current intrusion detection. Today, intrusion detection encompasses event log analysis for insider threat detection; network traffic analysis for threat detection; security configuration management; and file integrity checking. Clearly, current intrusion detection requires properties of both network and host-based intrusion detection systems. These systems are known as hybrid systems.

Intrusion detection is network-based when the system is used to analyze network packets. This is in contrast to host-based intrusion detection, which relates to processing data that originates on computers themselves, such as event and kernel logs. Network packets are usually "sniffed" off the network, although they can derive from the output of switches and routers. The most common protocol targeted is TCP/IP. Network sources are unique because of their proximity to unauthenticated, or outside, users. They are positioned to detect access attempts and denial of service attempts originating outside the network.

There are many attack scenarios that would not be detected by host-based technology, thereby highlighting the differences between the two. Unauthorized access occurs when an outsider comes in over the network and logs into the system uninvited. This can be detected by host-based systems once the attacker is inside, but the ultimate goal is to detect them before they get access, or during the process of getting access. Another scenario is password downloads. Unauthorized password file downloads gives attackers the ability to attack other systems. The Network Security Monitor, one of the first network intrusion detection systems looked for the pattern "/etc/passwd" in FTP traffic outbound from the network. Still another scenario is denial of service. These attacks are named because they result in a resource not being available to service its users. One of the best examples of DOS was when Amazon.com, E-trade, and other e-commerce pioneers were shut down by a distributed denial of service attack in February 2000. The damage estimates ranged upwards of several million dollars for each site. Insiders can also cause DOS attacks as well as outsiders, but these types of attacks leave many clues, so outsiders usually initiate DOS attacks.

Figure 1. A Standard Network Intrusion Detection Architecture

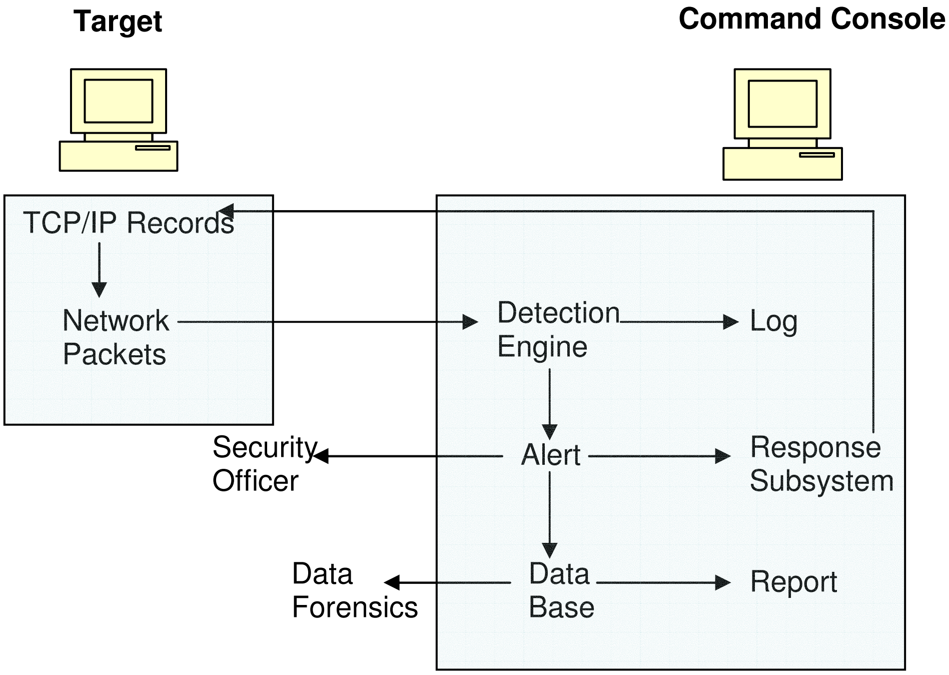


Figure 1 shows traditional sensor-based network intrusion detection architecture. A sensor is used to "sniff" packets off of the network where they are fed into a detection engine which will set off an alarm if any misuse is detected. These sensors are distributed to various mission-critical segments of the network. A central console is used to collect the alarms from multiple sensors. However, in order to better understand the traditional sensor-based architecture, the lifecycle of a network packet should be examined.

1. The network packet is created when one computer communicates with another.

2. The packet is read, in real time, off the network through a sensor that presides on a network segment located somewhere between the two communicating computers. The sensor is usually a stand-alone machine or network device.
3. A sensor-resident detection engine is used to identify predefined patterns of misuse. If a pattern is detected, an alert is generated.

4. The security officer is notified about the misuse. This can be done through a variety of methods including audible, visual, pager, email, or through any other different method.

5. A response to the misuse is generated. The response subsystem matches alerts to predefined responses or can take responses from the security officer.

6. The alert is stored for correlation and review at a later time.

7. Reports are generated that summarize the alert activity.

8. Data forensics is used to detect long-term trends. Some systems allow archiving of the original traffic to replay sessions.

A few years ago all commercial network intrusion detection systems used promiscuous-mode sensors. However, these technologies were subject to packet loss on high speed networks. A new architecture for network intrusion detection was created that dealt with the performance problem on high speed networks by distributing sensors to every computer on the network. In network-node intrusion detection each sensor is concerned only with packets directed at the target in which the sensor resides. The sensors then communicate with each other and the main console to aggregate and correlate alarms.

However, this network-node architecture has added to the confusion over the difference between network and host-based intrusion detection. A network sensor that is running on a host machine does not make it a host-based sensor. Network packets directed to a host and sniffed at a host are still considered network intrusion detection.

Figure 2: A Distributed Network-Based/Host Resident Intrusion Detection Architecture

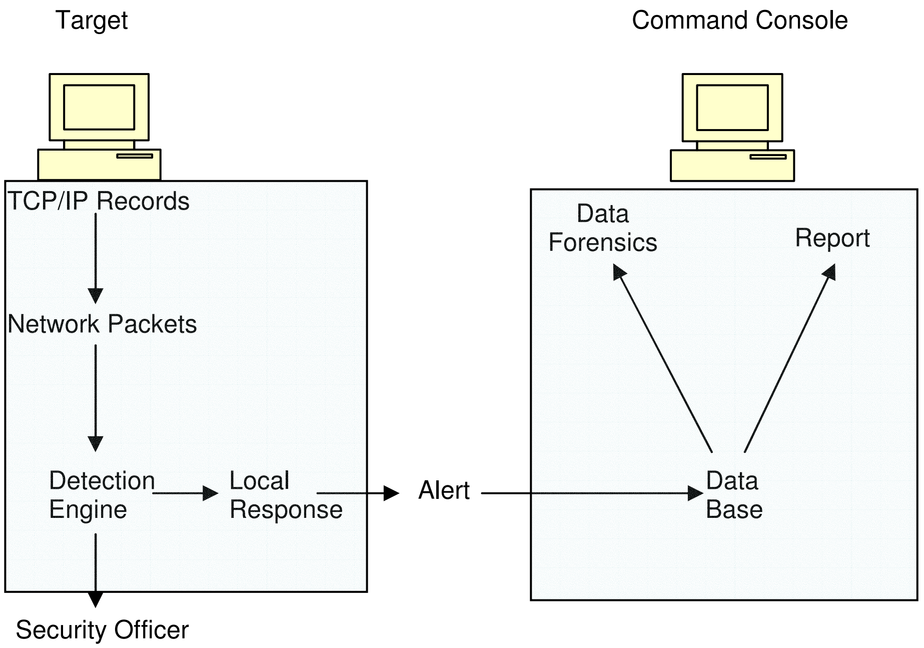


Figure 2 represents the network-node intrusion detection architecture. An agent is used to read packets off the TCP/IP stack layer where the packets have been reassembled. The packet is then fed into the detection engine located on the target machine. Network-node agents communicate with each other on the network to correlate alarms at the console.

1. A network packet is created.

2. The packet is read in real-time off the network through a sensor resident on the destination machine.

3. A detection engine is used to identify pre-defined patterns of misuse. If a pattern is detected, an alert is generated and forwarded to a central console or to other sensors in the network.

4. The security officer is notified.

5. A response is generated.

6. The alert is stored for later review and correlation.

7. Reports are generated summarizing alert activity.

8. Data forensics is then used to look for long-term trends.

However, the architectures require operational modes in order to operate. Operational modes describe the manner the intrusion detection system will operate and partially describe the end goals of monitoring. There are two primary operational modes to use network-based intrusion detection: tip-off and surveillance.

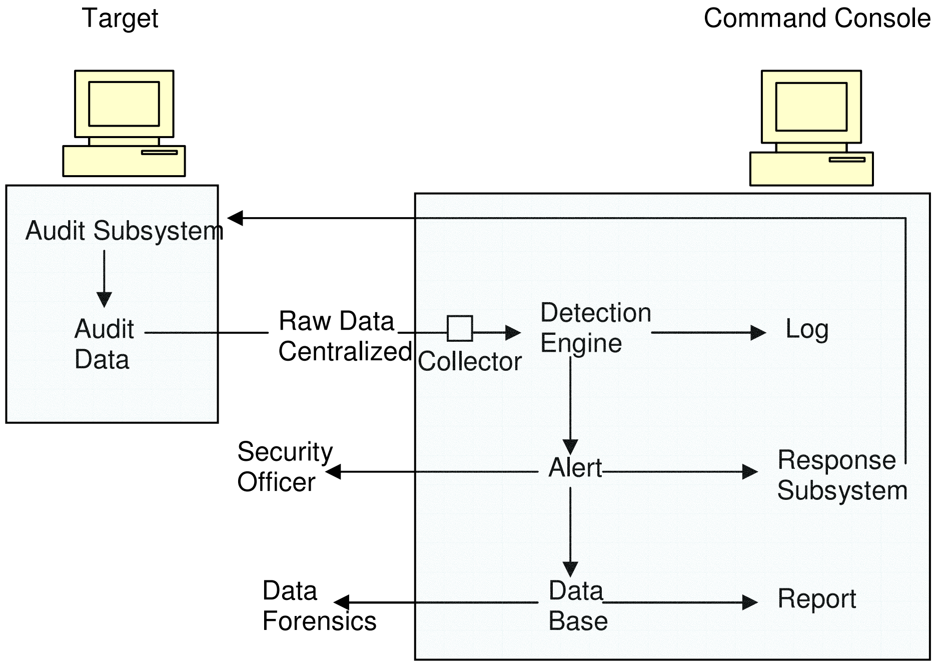
The system is used to detect misuse as it is happening. This is the traditional context for intrusion detection systems. By observing patterns of behavior, suspicious behavior can be detected to "tip off" the officer that misuse may be occurring. The defining characteristic for tip-off is that the system is detecting patterns that have not been detected before. During surveillance, targets are observed more closely for patterns of misuse. Surveillance is characterized by an increased observance of the behavior of a small set of subjects. Unlike tip-off, surveillance takes place when misuse has already been suspected. Surveillance results from a tip-off from either the intrusion detection system or another indicator.

In order for there to be a tip-off a data source needs to be searched for suspicious behavior. Host-based intrusion detection systems analyze data that originates on computers, such as application and operating system event logs and file attributes. Host data sources are numerous and varied, including operating system event logs, such as kernel logs, and application logs such as syslog. These host event logs contain information about file accesses and program executions associated with inside users. If protected correctly, event logs may be entered into court to support the prosecution of computer criminals.

There are many attack scenarios that host-based intrusion detection guards against. One of these scenarios is the abuse of privilege attack scenario. That is when a user has root, administrative or some other privilege and uses it in an unauthorized manner. Another scenario involves contractors with elevated privileges. This usually happens when an administrator gives a contractor elevated privileges to install an application. Most security policies restrict nonemployees from having root or administrator privileges, however it might be easier to elevate the user and reduce privileges later. However, the administrator might forget to remove the privileges. A third attack scenario involves ex-employees utilizing their old accounts. Most organizations have policies in place to delete or disable accounts when individuals leave. However, they take time to delete or disable, leaving a window for a user to log back in. Another scenario involves modifying web site data. There have been many cases, against government agencies in particular, that result in uncomplimentary remarks posted on web sites. While these attacks originate from outside the network, they are perpetrated on the machine itself through alteration of data.

With a review of what attacks host-based intrusion detection systems prevent, it's important to examine the architecture to see how it prevents those attacks. In the centralized architecture, data is forwarded to an analysis engine running independently from the target. Figure 3 represents the typical life cycle of an event record running through this type of architecture. And Figure 4 represents distributed real-time host-based intrusion detection architecture. The difference between the two is that in Figure 3 the raw data is forwarded to a central location before it is analyzed and, in Figure 4, the raw data is analyzed in real time on the target first and then only alerts are sent to the command console. There are advantages and disadvantages to each method. However, the best systems offer both types of processing.

Figure 3: A Centralized Host-Based Intrusion Detection Architecture



1. An even record is created. This occurs when an action happens; such as a file is opened or a program is executed like the text editor like Microsoft Word. The record is written into a file that is usually protected by the operating system trusted computing base.

2. The target agent transmits the file to the command console. This happens at predetermined time intervals over a secure connection.

3. The detection engine, configured to match patterns of misuse, processes the file.

4. A log is created that becomes the data archive for all the raw data that will be used in prosecution.

5. An alert is generated. When a predefined pattern is recognized, such as access to a mission critical file, an alert is forwarded to a number of various subsystems for notification, response, and storage.

6. The security officer is notified.

7. A response is generated. The response subsystem matches alerts to predefined responses or can take response commands from the security officer. Responses include reconfiguring the system, shutting down a target, logging off a user, or disabling an account.

8. The alert is stored. The storage is usually in the form of a database. Some systems store statistical data as well as alerts.

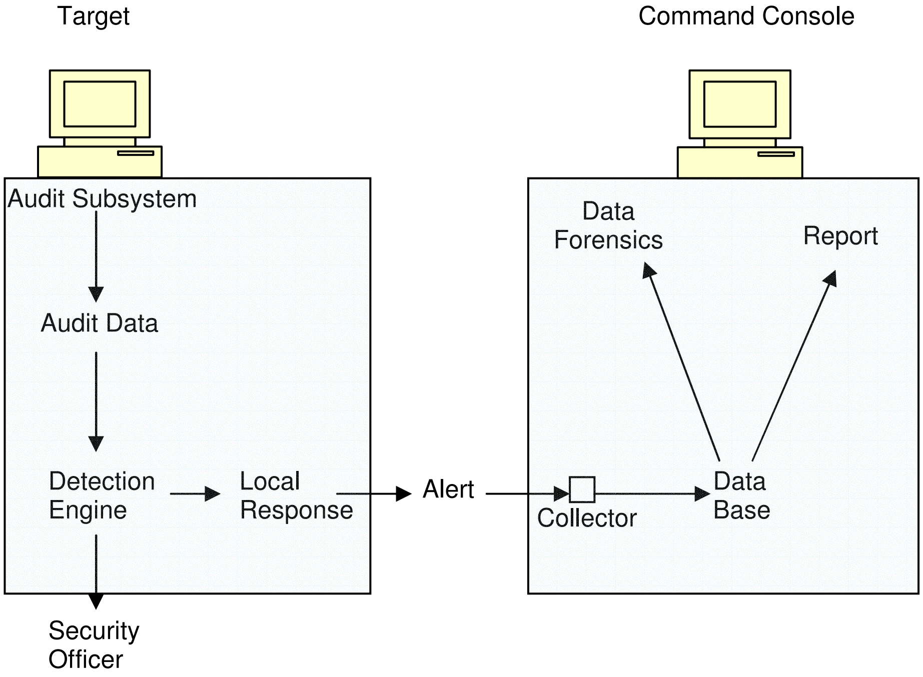
9. The raw data is transferred to a raw data archive. This archive is cleared periodically to reduce the amount of disk space used.

10. Reports are generated. Reports can be a summary of the alert activity.

11. Data forensics is used to locate long-term trends and behavior is analyzed using both the stored data in the database and the raw event log archive.

The lifecycle of an event record through a distributed real-time architecture is similar, except that the record is discarded after the target detection engine analyzes it. The advantage to this approach is that everything happens in real-time. The disadvantage is that the end users suffer from system performance degradation.

Figure 4: A Distributed Real-Time Host-Based Intrusion Detection Architecture



1. An event record is created.

2. The file is read in real-time and processed by a target resident detection engine.

3. The security officer is notified. Some systems notify directly from the target, while others notify from a central console.

4. A response is generated. The response may be generated from the target or console.

5. An alert is generated then sent to a central console.

6. The alert is stored. Statistical behavioral data outside alert data are not usually available in this architecture.

7. Data forensics is used to search for long-term trends. However, because there is no raw data archive and no statistical data, this capacity is limited.

8. Reports are generated.

Table 1 summarizes the advantages and disadvantages of centralized detection architecture. There is little impact in performance on the target machine because all the analysis happens elsewhere. Multi-host signatures are possible because the centralized engine has access to data from all targets. Finally, the centralized raw data can be used for prosecution provided the integrity of the data is preserved.

Table 1: Advantages and Disadvantages of a Centralized Detection Architecture

Advantages	**Disadvantages**
No performance degradation on target Statistical behavioral information Multi-host signatures Raw data archive for prosecution support	No real-time detection No real-time response

Table 2 illustrates the advantages and disadvantages of a real-time distributed intrusion detection system. This table is a mirror image of Table 1 with a few minor additions.

Table 2: Advantages and Disadvantages of a Distributed Real-Time Architecture

Advantages	**Disadvantages**
Real-time alerting Real-time response	Performance degradation on target No statistical behavioral information No multi-host signatures No raw data archive for prosecution support Reduced data forensics capabilities Gaps in data analysis when system is offline

Host-based and network-based systems are both required because they provide significantly different benefits. Detection, deterrence, response, damage assessment, attack anticipation and prosecution support are available at different degrees from the different technologies. Table 3 summarizes these differences.

Host-based systems are designed more to deter insiders, but can't effectively deter outsiders. Insiders fear that misuse will be detected through a host-based system but an outsider will know that host-based detection will have little effect in detecting efforts to break in. The exact opposite is true for network intrusion detection systems. An outsider is more likely to be deterred knowing that the computer may be identified, whereas the insider won't need to execute any of the transactions normally detected by network intrusion detection.

To put it simply, host-based intrusion detection detects insider misuse while network intrusion detection detects outsider misuse. Also, network intrusion detection focuses more on abuse of vulnerabilities while host-based systems focus on abuse of privilege. This makes sense because insiders do not have to exploit vulnerabilities because they are already in the network and have their own privileges. However, outsiders must exploit vulnerabilities to get inside the network and gain privileges. Once the outsider gains privileges, that outsider should be considered an insider.

Host-based systems provide poor real-time response and cannot effectively protect against one-time catastrophic events. They are, however, excellent at detecting and responding to long term attacks, such as data thieving or disgruntled employees. Network intrusion detection is the exact opposite. It is effective at real-time detection and response as it pertains to the network. Network intrusion detection can also be effective at detecting long-term attacks suck as sniffer programs regularly reporting information from outside the firewall.

Host-based systems stand out when it comes to determining the extent of a compromise after loss. They usually maintain large databases that indicate historical information that could serve as evidence in the prosecution of the misuser. Network intrusion detection can be used to trace a series of connections through a network. This could be helpful in finding the misuser to whom the evidence from the host-based system can be attributed to.

Host-based systems maintain a large database of behavioral information that can be mined for trends indicative of misuse. An example could be identifying a user who is scanning for sensitive data. The data forensics capabilities are designed for this purpose. Network systems have similar capabilities, but the network environment limits the use of these capabilities. Network detection systems can be used for identifying problem IP addresses so as to have the firewall block those addresses in the future.

Prosecution support is based on the integrity of the source data. Host-based systems can provide some level of integrity, and they provide the forensics tools to present the data in a valid format. Unfortunately, network data packets can't be trusted by their very nature. Spoofing is common practice, such as spoofing IP addresses; and so renders network data invalid.

Table 3: Comparing Network- and Host-Based Benefits

Benefit	**Host**	**Network**
Deterrence	Strong deterrence for insiders.	Strong deterrence for outsiders.
Detection	Strong insider detection. Weak outsider detection.	Strong outsider detection. Weak insider detection.
Response	Weak real-time response. Good for long-term attacks.	Strong response against outsider attacks.
Damage Assessment	Excellent for determining extent of compromise.	Very weak damage assessment capabilities.
Attack Anticipation	Good at trending and detecting suspicious behavior patterns.	None.
Prosecution Support	Strong prosecution support capabilities.	Very weak because there is no data source integrity.

Policies are what drive the operation of an intrusion detection system. Effective policy can significantly reduce performance degradation and resource costs associated with operating a host-based detection system. Audit and detection policies are two primary policies that need to be managed effectively. If the audit and detection policies fail to be managed, the deployment will likely fail.

Audit policy defines which end user actions will result in an event record being written to an event log. Reducing the number of event log records collected can reduce the performance related issues with host-based systems, so an effective audit policy is crucial. Audit policies are complex affairs involving multiple flags in numerous locations. The large majority of events are caused by system object access. To control the audit subsystem to a fine detail, audited events are restricted to access of mission-critical files, folders and objects only.

However, there are two points to consider here. Is it more desirable to track the people who fail to access the sensitive data, or the ones who are successfully accessing sensitive data? Capturing legitimate behaviors introduces the concept of positive and negative auditing. Negative auditing is traditional exception auditing. This includes events such as failed login attempts, failed file reads and failed file modifications. Positive auditing is the process of logging both successful and failed events. This represents a reduction in the number of false-positives. Many of the alerts that operators may consider noise or false-positives are actually useful positive audits. A specific example of positive auditing is keeping track of all users who access a certain directory. These records are useful for both attack anticipation and damage assessment.

However, there are trade-offs to the choice of audit policy. Enabling event failures would indicate potential problem areas, while enabling full auditing would provide too much information for view and cause significant performance penalties. The net result is that exception auditing became the standard audit policy.

The key to a successful intrusion detection system is an effective audit policy. An effective audit policy is one that provides a suitable number of event records. That is, not so much that they cannot be effectively analyzed, and not so little that interesting behaviors are lost. An effective audit policy involves understanding the detection requirements specific to an organization and the location of mission-critical system objects.

Host-based intrusion detection, like network-based, has a detection engine that identifies patterns of misuse in the target data. Detection policy defines the patterns that are detected in the event log records. Signature recognition is the most common detection mechanism in host-based systems. These signatures are pre-defined patterns that have been defined as suspicious by the security officer.

Signatures are rules that define a sequence of events and a set of transitions between the events. For example a typical pattern being searched for is unauthorized modification to files in the directory that holds employee financial information. The key to a good detection policy is properly configured signatures, with the appropriate number of active signatures detected in real-time.

Data sources to which the signatures can be search for usually originate from log files. File logs, operating system logs, and application logs are valuable data sources for most host audit policies. Syslog is a generic logging utility available in most UNIX systems. An advantage to using Syslog as an audit source is that the ASCII format fosters simple text searches that are portable across any UNIX system. Additional effective audit data sources include firewalls, event logs, and backup applications.

Network and host-based intrusion detection systems offer very similar benefits. Both systems work well enforcing outsider deterrence. Network-based systems can put attackers on notice that their actions may lead to legal action. This serves as a wake-up call to inexperienced hackers that they are not as safe as they thought. Similarly, host-based systems act on the principle that people who know that their actions are being monitored are less likely to commit misuse. In addition, both systems detect a wide range of activity. Network-based systems detect more incoming network activity while host-based systems detect more insider activity. Finally, both systems can react and/or alert the security officer to the possible misuse.

However, it is necessary to mention a disadvantage to network-based intrusion detection systems. If the incoming traffic is encrypted, then the detection engine is rendered useless because it cannot search for patterns of misuse in the payload of network packets.

Signatures are the most deterministic of the analysis methods but they are not the only host-based technique. Statistical analysis, metarules, and artificial intelligence such as neural networks are also used to form a non-deterministic depiction of behavioral trends that can be very effective at detecting misuse.

Host-based intrusion detection systems also analyze user statistics to determine misuse. This method is called statistical analysis. Statistical analysis provides some of the most powerful features in intrusion detection. It can be used to identify trends in behavioral data and damage assessment. Essentially, statistics are gathered from the history of the user's behavior and compared with their short-term behavior. When the difference is sufficiently large between the two, a warning flag is raised automatically. This method has the advantage of detecting misuse to which there are no signatures available as in newly developed exploits.

Intrusion detection using neural networks and support vector machines are being researched in universities, but haven't made it to the market quite yet. The construction of SVM intrusion detection systems consists of three phases. The first is preprocessing, which uses automated parsers to process the randomly selected raw TCP/IP dump data into machine readable form. The second phase consists of training SVMs on different types of attacks and normal data. The data have 41 input features and fall into two categories: normal (+1) or attack (-1). The SVMs are trained with normal and intrusive data. The final phase involves measuring the performance on the testing data. In theory, SVMs are learning machines that plot the training vectors in high dimensional feature space, labeling each vector by class. Furthermore, SVMs classify data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in feature space. The SVMs are based on the concept of structural risk minimization, which recognizes true error on unseen examples. The process to which the data is classified involves partitioning the data into two classes: normal and attack, where attack represents a collection of 22 different attacks belonging to the four classes, either: DOS attacks, unauthorized access from a remote machine, unauthorized access to a local super user privileges, or surveillance and other probing. The object is to separate normal (1) and intrusive (-1) patterns. The SVMs are trained with normal and intrusive data. The primary advantage of SVMs is binary classification and regression which implies low expected probability of generalization errors; however there are many more advantages. Another advantage is speed as real-time performance is of primary importance to intrusion detection systems. In addition, the SVMs are very scalable. They are relatively insensitive to the number of data points and the classification complexity does not depend on the dimensionality of feature space. A final advantage is that because attack patterns are dynamic in nature, SVM can dynamically update training patterns.

The neural network intrusion detection system also consists of three phases. The first involves using automated parsers to process the raw TCP/IP dump data into machine-readable form. The second phase involves training neural networks on different types of attacks and normal data. The input has 41 features and the output assumes one of two values: intrusion (22 different attack types), or normal data. The training of the neural networks was conducted using feed forward back propagation algorithm using scaled conjugate gradient decent for learning. The network was required to train until the mean error rate of 0.001 was met. The final phase involves testing the performance of the IDS. According to the reference paper, the system detected with 99.25% accuracy.

However, once an attack has been detected a response needs to be executed. These responses come in the form of predefined response scenarios which are enacted automatically by the response subsystem, or manual responses specified by the security officer. Automated responses imply no human intervention however, so they are inherently dangerous.

It has been a long held myth that automatic response can be used effectively to stop intruders before misuse occurs. However, in actuality automated responses can be used by attackers as an excellent denial of service mechanism to the network. Automated responses usually have a detrimental effect on system operation, varying degrees of effectiveness, and usually require recovery of the system at some cost. Examples include logging off the user, disabling a user's account, shutting down the target system, stopping a good process, breaking the network connection, and denying connection requests.

Logging off users is used when a user has been positively identified as misusing the system. This is not very effective because the user simply can log back on. This becomes a denial of service when a good user has been mistaken for a bad user and is logged off. It can happen that a good user is repeatedly logged off and thus is prevented from doing their job.

Disabling an account is much more effective because the bad user is now unable to logon and is effectively banned from entrance into the system. Although effective, this requires intervention from the administrator or help desk to reenable an account, and that costs the organization money. If a bad user abuses the account of a good user, then the good user is denied access to the account until the issue is resolved.

Shutting down the target system is potentially one of the most severe of all automated responses. Shutting down a machine denies service to all of its users and may take some time to recover. Also, information loss may result, depending on the types of applications running at the point of shutdown.

Stopping a process, in theory, just shuts down the offending process, leaving all the good processes to do their legitimate jobs for authorized users. Unfortunately, most of the target processes are good processes including sendmail. This method can be effective at shutting down sniffers and password crackers that are stealing cycles as well as attacking security.

Breaking the network connection is used primarily by network intrusion detection systems when a villain connection has been discovered. As with logging off a user, this is not an effective deterrent because the user need only reestablish the connection.

Denying network requests is the network equivalent of disabling an account with the same internet risks. That is, the attacker could spoof a legitimate user's IP address whose service could then be blocked.

However, there are examples of good candidates for an automated real-time response. They are characterized by the following traits: the occurrence is rare, the event is catastrophic, the event can't be spoofed, and the event is positively identifiable as misuse.

Behavioral data forensics describes data mining techniques used in analyzing large intrusion detection databases. Most people think computer forensics is about construction data and configuration information from a computer that has been compromised. Traditional computer forensics also involves finding temporary files and other evidence that can be used to piece together a story. In addition, behavioral data forensics is about analyzing behavior. It involves error detection and eradication by looking at historical events rather than static configuration information. Trending and damage assessment are two major advantages of behavioral data forensics.

Behavioral data forensics is usually used for attack anticipation, trending and damage assessment, however; it can indicate almost every aspect of business operations. Behavioral data forensics has many uses some of which are:

- Detecting insiders misusing corporate computers by identifying trends of misuse and irregular activity.

- Detecting hackers attempting to attack corporate computers by identifying attack trends.

- Improving corporate policy by creating policies that better fit observed behavior rather than predict behavior. Misuse trends could be attributed to bad policies or the lack there of. This is an example of how intrusion detection systems could be used to address measurable problems rather than hype or speculation.

- Distinguishing between process and automated problems by identifying root causes. Occasionally problems are the result of automated processes; such as poorly configured batch/scheduled programs.

- Identifying areas of inefficiency. Most misuse, both internal and external, ultimately results in reduced organizational efficiency. Attackers and inefficient processes distract employees from their duties. Behavioral data forensics can be used to identify these types of inefficiencies.

- Preventing minor issues from becoming large problems by identifying trends when the associated problems are still minor. For example, a server that is showing a trend toward processor failures can be replaced before it fails completely and stops the business. Catching these sorts of small issues early can save an organization millions.

- Balancing workload between the users and the target system by gathering data that show relative workloads. Behavioral data comparing can show uneven workloads. The resulting data could then be used to redirect tasks and more evenly distribute workload.

Behavioral data forensics is heavily dependent on target database and data sources. User-centric monitoring means the database is optimized to provide user data, while target-centric monitoring means the database is optimized to provide target data. For example, user-centric monitoring analyzes all the logins a specific user had across the entire enterprise, while target-centric monitoring analyzes all the logins to a specific set of targets.

Target-centric monitoring is crucial to an effective intrusion detection system. Illustrating this requires looking at search times for large reports. In traditional intrusion detection database organization the signature data is stored in tables representing targets. A search for a signature to all users on a single target will be very efficient and pulled from a single table. However, a search for a signature attributed to one person cross all targets will require a search for a single user in each target table, which is very inefficient.

Consider a standard damage assessment activity where a user has been positively identified as a misuser and now the security officer has to determine the extent of the damage caused by the misuser. The security officer has to determine if any other systems were compromised. User-centric monitoring makes it possible to generate a single report that shows all servers accessed by the misuser in some period of time. With target-centric monitoring the security officer must work with very inefficient searches that could take hours depending on the size of the corporation.

Intrusion detection systems provide a wide range of capabilities from security configuration to log monitoring to network packet analysis. Determining how to operate the system will significantly influence the overall effectiveness of deployment. The most common question asked during the purchasing process for an intrusion detection system is related to the number of man hours required to operate the system. The fear is that resource costs will drive the total cost of ownership.

There are many operational models that are in operation today, including background, on-demand, scheduled and real-time. Each model can be related to certain organizations depending upon what priorities and resources the company holds.

Background operations do not require human intervention. The goal of background operation is to reduce human intervention and labor-hours overall, which is desirable because people and their time cost money. Maximizing background operation will lower total cost of ownership.

Active intrusion detection components are those that are persistent and execute constantly. Most active programs are design to run in the background, including network sensors to sniff data off the network and host-resident agents to scan event log data. Examples of background operations include automating detection and response, gathering positive auditing used for data forensics and centralizing data for prosecution support and damage assessment.

Unfortunately, background operations have an element of associated risk, especially because they do not have any human intervention. So, while background operations are inexpensive from a resource standpoint, they can lead to serious problems. For example, denial of service could occur when automated processes that centralize data fail. Because of the background operation there would be no human intervention to correct the situation.

On-demand operation is the lowest overhead operational mode. Essentially, the intrusion detection system does nothing unless it is told to. After deployment, on-demand components lay idle until they are ordered to do a specific operation. Both host-based and network-based systems can operate in this manner.

On-demand operations are used in a number of different ways. They can be used for on-demand monitoring and surveillance. This model has significantly fewer risks than those associated with active components because there is no ongoing execution to cause operational problems. Examples of on-demand operation include: performing a one-time vulnerability scan to determine the state of a security configuration, verifying that an audit policy is set up as required, assessing damage after a loss has been determined and turning on monitoring after a threat has been suspected.

Scheduled operation performs intrusion detection tasks at regularly scheduled intervals. Scheduled operations usually reduce the staff necessary to operate a system. Scheduled operations can be applied for a number of reasons. One of those reasons is performing a resource intensive operation at a time when resources are abundant, like during the night or holiday. Another reason is saving time and resources by producing regularly scheduled reports like automatically generating an end of the week or month report that highlights all the security incidents.

Many scheduled operations are low risk, however, scheduled operations imply automated operation, and there is inherent risk in any automated operation where human intervention is not involved in dealing with emergencies. Still, many normal, repetitive, everyday operations may be scheduled, to free people so they can focus on more critical needs.

Real-time operation, as it pertains to intrusion detection, usually implies ongoing monitoring with detection and response in minutes or seconds. Most commercial intrusion detection systems already operate in real-time. However, host-based systems focus more on long-term attacks and trends so the real-time requirement is not as significant.

Real-time operation is a derivative of background operation. Real-time operation, in conjunction with real-time response, does not require human intervention. However, hidden costs are driven by certain factors. One factor is that real-time implies an agent running on the target system that may have performance implications. Also real-time detection implies automated response without human intervention and so has significant inherent risk. Finally, real-time implies that a team will be available to handle the alerts as they arise, significantly increasing resource code.

In conclusion, I believe that combined network-based and host-based intrusion detection systems effectively prevent attacks from insider as well as outsider sources. While there are new methods of intrusion detection, most systems utilize signatures to search for patterns of misuse and either report to the security officer or automatically respond to the misuse. Some intrusion detection systems even detect misuse without using signatures but by using behavioral data forensics. However, because of the inherent risk of some automated responses, there still needs to be a human who can oversee and ensure the state of the system.

Bibliography:

Books

Proctor, Paul E. The Practical Intrusion Detection Handbook. New Jersey: Prentice Hall PTR, 2001.

Hartmanis, J.; Goos, G.; van Leeuwen, J. Information Security Second International Workshop. Germany: Springer, 1999.

Northcutt, Steven. Network Intrusion Detection, An Analyst's Handbook. Indianapolis: New Riders, 1999.

Papers

Bace, Rebecca. "An Introduction to Intrusion Detection and Assessment: for System and Network Security Management." ICSA White Paper, 1998.

Mukkamala, Srinivas; Janoski, Guadalupe; Sung, Andrew. "Intrusion Detection Using Neural Networks and Support Vector Machines." IEEE IJCNN May, 2002.

---. "Intrusion Detection Using Support Vector Machines." 29 May 2003 <http://www.cs.nmt.edu/~IT/papers/hpccsandiagofinal.pdf>.