Author's name:  Daniel O. Hill, CISA, CBCP
GSEC Practical Assignment, Version number 1.4b, (amended August 29, 2002)

# The Human Factor –

# Adding Intelligence and Action to Intrusion Detection

## Abstract

Intrusion detection systems need to communicate with analysts on multiple levels. They need to be scaleable, reliable, effective, and efficient; in addition, they need to be responsive to human intelligence and intuition. To be safe from attack themselves, they need to be invisible to hackers.

This paper explores the current state of Intrusion Detection Systems (IDS) technology with its roots dating from 1985. It identifies system requirements and essential elements in the context of an overall architecture; and it highlights several systems, available today, that fit nicely into the suggested architecture.

The future of IDS will be much like its past. Technology will continue to evolve, attacks will become more difficult to detect, and humans will be needed more than ever.

## The Problem

One would think that we would be further along than we are today. Intrusion detection systems seem to have all the glitter of something new and intriguing. A few highly refined, home-grown, centralized IDS systems are in place in university and research laboratory environments, but most of us are still hard at work trying to make good use of freeware, shareware, and commercially available IDS tools. Most of us are in the early stages of deployment. We know IDS is an evolving discipline, and we're looking for the next big thing. Many good people continue to work diligently to keep ahead of the hackers who come from all over the world to test our gates and exercise their collective creativity against us.

Serious thinking about intrusion detection systems began in 1985 with a technical report co-authored by Dorothy Denning and Peter Neumann of SRI International's Computer Science Lab. The report, "Requirements and model for IDES-A real-time intrusion detection system", was a detailed description of the design and application of an Intrusion Detection Expert System (IDES) [1]. In 1987, Dorothy Denning wrote a paper published in IEEE's Transactions on Software Engineering, "An Intrusion-Detection Model" [2]. The paper was based on the SRI work. It was widely read in the software engineering community, and it became the lighthouse on a distant shore for anyone interested in helping to bring intrusion detection into reality.

## So What's the Problem?

We've had challenges.

Our open and competitive technology market drives the development of a wide variety of IDS tools. Function and capability are constantly improving, allowing whole new IDS architectures to be considered. We have inexpensive network taps that copy all inbound and outbound traffic to a directly attached protocol analyzer. We have smart IDS sensors that can preprocess traffic and forward only the "interesting" packets. We have host-based sensors that monitor application servers and major network components by looking at incoming service requests, system behaviors, and checking the integrity of data and software.

The hackers are serious. They are getting smarter and more dangerous. They share their knowledge of system and protocol weaknesses, exploits and tools on hacker websites. They are using computers and the Internet in creative ways to increase their effectiveness. Automated probes, brute force techniques, and coordinated attacks launched simultaneously from hundreds of compromised computers, bring speed, power, and subtlety to the hacker's tool bag. Probes launched from around the world constantly test defenses and search for vulnerabilities. The number of test probes against a site of only moderate interest can run in the millions per week.

We appreciate the need for "defense in depth". We have sensors on network segments outside the firewall. If an attacker breaches the firewall, or becomes an employee, sensors inside the firewall provide a second layer of defense. If the attacker is able to avoid detection on the network, host-based sensors give us a third line of defense. Sensor data may be forwarded to a central analysis system where data from multiple sensors can be correlated and compared with historical data. Activity overlooked by a single sensor may be caught when data from multiple sensors is compared. A team of smart and dedicated humans has to decide on an architecture that is right for their environment, implement and tune it, keep it current, understand and act on what the system is telling them. And therein lies the challenge.

A poorly designed IDS can itself be vulnerable to attack. An untuned IDS can create an enormous amount of useless and incorrect data (false positives), and can miss true attacks (low detection rate or false negatives). We're learning as we go – working hard, spending money, making in-flight modifications, manually processing and correlating system output. We're rowing in the right direction. We're making progress, but we're still a long way from the lighthouse.

We’ve been busy.

Over the past fifteen years we have been totally occupied absorbing new technology, and we’ve been concentrating on the network. Technology in the network has been advancing so fast that we are only now thinking about centralized automated analysis of network and host activity. Since 1987 we have seen the rise of network and application servers, routers, switches, firewalls, local area networks, the world wide web, low cost high-speed Internet connections, gigabit personal computers, wireless networking, and secure network transmission – to name a few advances in computer technology.

Hand carried and hand operated network sniffers have become network IDS sensors. They can be remotely controlled, and can transmit files to a central location for analysis. Network IDS sensors are available, and are still maturing.

In 1987 network sniffers were in their infancy, and TCP/IP was no more than a small branch on the tree whose trunk was IBM’s SNA. The Internet was in place, but it was all command line functions. We were still using the basic TCP/IP application protocols – ftp for file transfer, telnet for terminal emulation, and smtp with a variety of local “post office” software for delivery of email. Workstations and wide-area telecommunications were slow and expensive. Compaq shipped its first Intel 386-based PC in 1987. Local-area networks were still in their infancy. Microsoft first became a publicly traded company in 1986, and shipped Windows 1.01 in August 1987. Network operating systems were only vague concepts.

The problems with system audit logs cited by Dorothy Denning in 1987 [2] continue to exist today. There are no standards for system audit logs, and logging is often turned off or is restricted to logins only, because more detailed logging impacts application performance.

Dorothy Denning’s model is a near real-time expert system. It analyzes system audit logs for activities that are outside policy boundaries, or outside established norms. It adjusts norms to accommodate shifting patterns of acceptable use. But it is all host-based intrusion detection, and does not consider a centralized analysis strategy.

Dr. Denning’s model is as valid today as it was in 1987. The statistical models for normalizing activity, detecting misuse, and identifying anomalies are still good. They are applicable to network and host activity, local and centralized analysis.

## An IDS Architecture

We need an IDS architecture that:

- Collects, organizes, and analyzes information from any number of network based and host based IDS sensors;
- Shows overall status at a glance;
- Generates a report hierarchy that allows humans to start their investigation at any level and work up, down, and across as the trail leads;
- Enables humans to interact with the system to test their intuitions;
- Allows humans to direct further analysis and redirect how data is analyzed;
- Is invisible to hackers and is secure, because it is physically isolated from the production network.

The architecture, when implemented, is a fully integrated intrusion detection system that makes effective use of current technology, is scaleable, and is responsive to human intelligence, intuition, and control. Such a system would have the following characteristics:

1. Network taps, invisible to hackers because they have no IP address. Taps copy all inbound and outbound traffic flowing on a single network segment. They are inexpensive, so can be placed on any segment where packet capture is desirable. Taps pass all copied traffic “off network” through a patch panel to a switch. The switch organizes tap data streams into VLANs for distribution to network IDS sensors. The whole IDS network is physically separate from the production network, and is totally invisible from the production network. Production network traffic flows one-way only, into the IDS network.
2. One or more servers on the IDS network that buffer frames from the network taps, preventing network IDS sensor overload.
3. Network IDS sensors on the isolated IDS network. The network IDS sensors perform first-level analysis, report results, and provide a human interface for further analysis and ad hoc reporting. All reports are sent to a central collecting point for further analysis, correlation, and reporting.
4. Host-based IDS sensors on critical network and application servers. Host IDS sensors perform preliminary analysis and send their output to a central collecting point for further analysis, correlation, and reporting. To prevent creating a “back-door” into the isolated IDS network, host sensors direct their output to a data sink – a server on the production network that provides destination-end protocol responses, but discards all data received. A network tap copies host IDS frames to the isolated IDS network.
5. A central analysis server located on the isolated IDS network to collect and process all IDS sensor output;

6. Central analysis tools that process and cross-correlate reports from multiple sensors and historical data, perform secondary analyses, and accurately detect misuse and anomalous behaviors;
7. Capability for sensors and central analysis tools to report results in real-time, near real-time, and past-time;
8. Central management tools that update "normal" profiles automatically, and with human assistance; tools that enhance expert system knowledge, store information about suspicious activity for later analysis and comparison, and accept new instructions for analysis and reporting;
9. Central reporting and alerting of misuse based on known patterns, attack signatures and knowledge. Reports may be related to reconnaissance activity, port scans, protocol state violations, policy violations, etc.;
10. Central reporting and alerting of anomalous behavior based on statistical analysis, "normal" profiles of past acceptable activity, fuzzy logic and neural networks, and data mining for correlation of current activities with historical data. Reports would identify activity outside established normal activity, profiles changing faster than usual or ranging too far from expected norms. Analysis would look for "low and slow" attacks attempting to "fly in under the radar", unusual or unexpected events in the network or in monitored hosts. Analysis would also attempt to correlate current activity with past activity that was within, but "on the edge" of normal and acceptable.
11. Interaction with humans
    a. Provide real-time alerts upon detecting "high confidence" intrusions and intrusion attempts that could have severe consequences.
    b. Provide multiple levels of analysis and reporting. High-level reports analyze and summarize mid-level reports. Mid-level reports analyze and summarize raw sensor data and attempt to correlate data from multiple sensors and historical data. IDS sensors would perform low-level analysis, working directly with network traffic frames and host activity data.
    c. Maintain links between reports and raw sensor data, to allow humans to dig down and across, and review raw sensor data as needed.
    d. Provide sensors with a two-way messaging capability that enables them to immediately forward alarms and high priority information to central analysis, and to receive ad hoc operating instructions.
    e. Humans can interact with the system at any level for ad hoc analysis, to request new reports, and to direct sensors to perform specific tasks. Tasks may be acted upon immediately, or may be based on conditions such as timeframes and events.

### Can We Get There From Here?

Dr. Denning may have had concerns as she considered the shortcomings of system audit logs [2]. Her model depended on vendors standardizing audit logs. The market didn't motivate them, and host audit logs are still in about the same state as they were in 1987. There are also no standard formats for IDS sensor output. There is some movement toward standardization and self-defining or XML-like data tagging, but all this is in early formative development.

System audit logs still need to report the details as described by Dr. Denning to enable host-based IDS sensors to collect and organize the data they need from them. The trend seems to be for host-based IDS sensors to independently gather all the system data they need. With all the advances in processor performance, there is no longer any excuse for less-than-adequate system monitoring.

System audit logs, network- and host-based IDS sensor data collection processes need to be open and adaptable, to allow both content and format of logs to quickly and easily adapt to new technology and changing attack modes.

Host monitoring has long been a performance concern. "Open and adaptable" monitoring increases the performance concern, and adds a security concern. System performance monitoring has been the exclusive territory of hardware vendors, and large performance monitoring software vendors. But we already have "open and adaptable" in Linux, and in IDS open source software – and everyone is adapting just fine. If we break something we usually find it in test, and can easily back it out with our configuration management tools. We have tools such as TripWire that calculate checksums to protect the integrity of software and data, and performance is no longer a problem. We can add as many inexpensive processors as needed, to give applications all the compute power they want.

### Systems and Tools in Place Today

A few home-grown IDSs have been evolving for several years, starting as research projects, competency work benches, and laboratory demonstrations. The systems are now protecting the networks and servers of the institutions that funded them. They all bear the imprint of Dorothy Denning [2].

### Florida State University

Florida State University (FSU) and Los Alamos National Laboratories (LANL) have taken quite different approaches to internal IDS development. Florida State maintains a strong academic research community, and uses a simulated network environment for IDS research. The network carries frames with encrypted payloads, and allows controlled testing of intrusion detection software. Analysis modules have been developed for both knowledge-based intrusion detection, and detection of anomalous activities on the network.

Florida State University has developed an advanced analysis method for detecting attack anomalies in secure Internet protocols such as SSL, SSH, HTTPS, S/MIME and IPSec. Secure Internet protocols are also known as security protocols, or simply secure protocols. A master's thesis written in November 2002 describes the FSU system. [3]

The FSU Behavioral Secure Enclave Attack Detection System (BSEADS) analyzes the use of security protocols and looks for anomalies. Only the secure protocols are analyzed. The encrypted payloads are not examined. The system first normalizes acceptable activities for each user, recording the timeframes of the normal observations. Profiles are updated by the system as activities change within normal limits. Activities that exceed the boundaries of the current normal profile are considered anomalous.

BSEADS was developed within a simulated network environment and architecture called the Secure Enclave Attack Detection System (SEADS). BSEADS source code was written in C++, and is provided in its entirety in the appendix of the referenced paper [3]. SEADS was developed to facilitate experimentation and testing of intrusion detection systems in networks carrying encrypted traffic. SEADS inserts normal security protocol traffic into the simulated network, and allows the researcher to introduce attacks for the IDS under test to detect. A knowledge-based IDS, KSEADS, was also developed at FSU to test detection of known misuse attacks in an encrypted environment.

Secure protocols are vulnerable to attacks such as man-in-the-middle, parallel session, sequence number guessing, and replay attacks. These are the hacker's favorites, because of the challenge inherent in planning and executing a successful attack. Anomaly detection is the only way to spot previously unknown attacks, and is becoming increasingly important as polymorphic attacks become more prevalent. [4]

### Los Alamos National Laboratories

The IDS work at Los Alamos National Laboratories has been directly for the production environment. Host-based IDS sensors preprocess system and application audit logs and transmit results to a central system. The central system performs further analysis using an expert-system approach, and reports its results to analysts for confirmation and follow-up action.

Los Alamos National Laboratories has been operated by the University of California for several years, and has a strong emphasis on research and development. LANL attracts PhDs and university students from all over the country and the world to its remote high desert facilities in Los Alamos, New Mexico. LANL's internal network supports approximately 9000 users. Host-based IDS sensors monitor activity on critical servers, and report in near real-time to a central analysis system. Network Anomaly Detection and Intrusion Reporter (NADIR), receives and processes sensor data, generates alerts as needed, and reports findings to a team of analysts. A paper presented in 1996 provides a high-level overview of the NADIR system [5].

NADIR has been operational since 1990. It uses automated audit record analysis and an expert system approach to identify misuse. The system features a distributed design. Agent software is installed in the monitored hosts, and a central server provides a collecting point for sensor data. The agents preprocess system and application audit logs, and look for signs of misuse and vulnerabilities on the host. The agents transmit data to the central server, which summarizes data received from the host agents into activity profiles. The central server then analyzes overall system and individual user activity against the expert system data and produces reports and alarms. Investigators resolve the few false positives the system reports, and take action to pursue identified cases of misuse.

The distributed design allows the sensors to concentrate on data collection and preliminary analysis, while the central server concentrates on in-depth analysis, alarms, and reporting functions. The result is more confidence in the detection system, more cycles available to application processing, and the ability to correlate information from multiple target systems. Correlation increases the ability of the system to detect distributed attacks that might otherwise go unnoticed.

### Publicly Available Systems

IDS vendor products have gone through multiple iterations, and buyouts have consolidated product lines. The market continues to evolve, with new products arriving on the scene monthly. Open source freeware also has a place, and in some cases is driving the market.

### Open Source Systems

Snort [6] may be the most widely recognized and most notable name in the open source world. Google brings over a million hits in response to the query "snort". Ok, some of the hits are about pig sounds; but without snort, the IDS system, why would you even want to enter a "snort" query? Snort IDS software is aggressively supported by a community of enthusiastic volunteers, and is fast becoming part of the basic IDS tool kit.

http://www.snort.org/about.html

IDScenter by Eclipse [7] is a “good but a little cranky” open source graphical user interface for Snort.

http://www.packx.net/packx/html/en/idscenter/index-idscenter.htm

Other open source tools are available from other sources too numerous to mention. Quality varies from elegant to questionable, and support suffers the same inconsistencies. Nothing is really free in the open source world, because the true cost is in the time spent bringing it up and keeping it operational in your environment.

**Vendor Provided Systems**

The following products are presented here primarily because they are currently leading the market, and they align well with the characteristics of the architecture as described earlier in this paper. While details remain to be filled in, major components of the architecture are represented.

Inexpensive network taps copy all inbound and outbound frames on a segment to a port designed to send frames one-way to a physically attached protocol analyzer. If the tap fails, traffic continues to flow on the network segment. The tap has no addresses, so is totally invisible to the network.

Network taps [8] include:

- Finisar (formerly Shomiti)
  http://www.finisar.com/virtual/virtual.php?virtual_id=117
- Intrusion – SecurNet IDS Taps
  https://www.intrusion.com/products/downloads/TapPO_1102.pdf
- NetOptics
  http://www.netoptics.com/11.html

Network Intrusion Detection (NID) sensors [9] include:

- Lancope’s StealthWatch
  http://www.scmagazine.com/scmagazine/2003_04/test_02/08.html

- NFR Security’s Network Intrusion Detection sensor
  http://www.nfr.com/products/nid/index.shtml
  The link provided below is a Users Guide for the NFR NID Sensor. The guide illustrates IDS architecture concepts and how network taps may be used. The URL shown must be copied and pasted into an Internet browser address window.
  https://eval.nfr.com/nid-v3/docs/NFR_NID_300_Series_Sensor_v3_0_Users_Guide.pdf

- Demarc Security's PureSecure
  http://www.demarc.com

- Internet Security System's RealSecure Network Sensor
  http://www.iss.net/products_services/enterprise_protection/rsnetwork/

Host Instrusion Detection (HID) sensors [10] include:

- Entercept, by Entercept Security Technologies (bought by Network Associates, Inc. on April 30, 2003)
  http://www.entercept.com/products/entercept/

- Internet Security System's RealSecure Server Sensor
  http://www.iss.net/products_services/enterprise_protection/rsserver/protector_server.php

- Internet Security System's RealSecure Server Sensor for Microsoft ISA Server
  http://www.iss.net/isaserver/

Central analysis systems [11] include:

- Internet Security System's SiteProtector
  http://www.iss.net/products_services/enterprise_protection/rssite_protector/

- Niksun's NetDetector
  http://www.niksun.com/index.php?id=194

- Sourcefire's Intrusion Management System – A vendor solution from the creators of Snort. Provides proprietary network intrusion detection sensors and a management console based on Snort open source software.
  http://www.sourcefire.com/products/products.htm

## A Glimpse of the Future

There is no silver bullet in our future, and no "next big thing". IDSs are high maintenance by nature, and will continue to be. Knowledge-based systems are only as good as the people who feed them, and attack signatures are constantly changing. While some analysts are talking about intrusion prevention as the next big thing, effective and efficient detection will be the heart and soul of IDS for many years.

Known attacks have been seen before, and have a recognizable signature. Suspected attacks are seen as deviations from established norms. The most any system can do reliably when encountering a suspected attack is to tell the humans and ask for help. Only a human can say to the intrusion detection system, “That deviation you saw is not really a threat to us. Thanks for letting us know about it, but you don’t have to tell us about that any more.” If a suspected attack is interesting to the humans, they can ask the IDS system questions about relationships it may have observed, and about any similar activities that may have occurred in the past. They can ask the sensors to look for specific kinds of host activity or network traffic, and the sensors can report back when the events occur.

We can expect the future of IDS to be much like its past. Priorities will change and we’ll absorb new IDS technology. IDS products will mature and become more integrated. Hackers will continue to find new vulnerabilities in network protocols, and new releases of software. Secure network protocols and detection avoidance will continue to be of interest to hackers. They will continue to be a threat and we will all work hard to keep ahead of them for a long time.

When a strong IDS system is imbedded inside a strong IT security architecture that is kept current and effective by a strong IT security management program supported by a well-trained staff, we have defense in depth. Hackers will knock but they’ll rarely get in. If they do get in, their activity will be observed. Any damage or disruption will be minimal and recoverable.

## References

[1] D. E. Denning and P. G. Neumann, "Requirements and model for IDES-A real-time intrusion detection system," Comput. Sci. Lab, SRI International, Menlo Park, CA, Tech. Rep., 1985.

[2] Denning, D. E., "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, vol. 13, pp. 222-232, 1987
URL: http://www.cs.georgetown.edu/~denning/infosec/ids-model.rtf

[3] Leckie, Tysen Glen, “Anomaly-Based Security Protocol Attack Detection” Masters Thesis, Florida State University College of Arts and Sciences, November 2002.
URL: http://websrv.cs.fsu.edu/research/reports/TR-021203.pdf

[4] Lemonnier, Erwan, “Protocol Anomaly Detection in Network-based IDSs” Defcom Sweden, Stockholm, 28 June 2001
URL:
http://erwan.lemonnier.free.fr/exjobb/report/protocol_anomaly_detection.pdf

[5] Jackson, Kathleen A., “A NADIR Progress Report” Los Alamos National Laboratories, a paper presented at UC Davis in November, 1996
URL: http://seclab.cs.ucdavis.edu/cmad/4-1996/pdfs/Jackson.pdf

[6] Snort open source IDS software
URL: http://www.snort.org/about.html

[7] IDScenter by Eclipse open source graphical user interface for Snort.
URL: http://www.packx.net/packx/html/en/idscenter/index-idscenter.htm

[8] Network taps
URLs:

- Finisar (formerly Shomiti)
  http://www.finisar.com/virtual/virtual.php?virtual_id=117
- Intrusion – SecurNet IDS Taps
  https://www.intrusion.com/
- NetOptics
  http://www.netoptics.com/11.html

[9] Network Intrusion Detection (NID) sensors
URSs:

- Lancope’s StealthWatch
  http://www.scmagazine.com/scmagazine/2003_04/test_02/08.html
- NFR Security’s Network Intrusion Detection sensor
  http://www.nfr.com/products/nid/index.shtml
- Demarc Security’s PureSecure

http://www.demarc.com

- Internet Security System's RealSecure Network Sensor
  http://www.iss.net/products_services/enterprise_protection/rsnetwork/

[10] Host Instrusion Detection (HID) sensors
URLs:

- Entercept Security Technologies (bought by Network Associates, Inc. on April 30, 2003)
  http://www.entercept.com/products/entercept/
- Internet Security System's RealSecure Server Sensor
  http://www.iss.net/products_services/enterprise_protection/rsserver/protector_server.php
- Internet Security System's RealSecure Server Sensor for Microsoft ISA Server Sensor
  http://www.iss.net/isaserver/

[11] Central analysis systems include:
URLs:

- Internet Security System's SiteProtector
  http://www.iss.net/products_services/enterprise_protection/rssite_protector/
- Niksun's NetDetector
  http://www.niksun.com/index.php?id=194
- Sourcefire's Intrusion Management System
  http://www.sourcefire.com/products/products.htm