Intrusion Detection[1]: Implementation and Operational Issues

John McHugh, Alan Christie and Julia Allen
Software Engineering Institute, Computer Emergency Response Team/Coordination Center

Intrusion detection systems (IDSs) are an important component of defensive measures protecting computer systems and networks from abuse. This article gives an overview of the most commonly used intrusion detection (ID) techniques. It considers the role of IDSs in the overall defensive posture of an organization and provides guidelines for their deployment, operation, and maintenance.

Attacks on the nation's computer infrastructures have become an increasingly serious problem. While government agencies have been common targets, the distributed denial-of-service attacks that materialized last year primarily targeted commercial sites.

"Hackers attacked some of America's most popular Web sites yesterday for the third day in a row, walling off frustrated consumers from companies that provide news and stock trading as law enforcement officials launched a nationwide criminal investigation … The computer attacks earlier this week temporarily blocked access to Web sites that read like a *Who's Who* of the new economy, including Yahoo, eBay, Amazon, CNN.com and Buy.com," reported the *Washington Post* on Feb. 10, 2000.

This story reflects the serious and sophisticated nature of today's cyber-attacks. During the past 12 years, the growth of incidents reported to the Computer Emergency Response Team/Coordination Center (CERT/CC®) has roughly paralleled the growth of the Internet.

As e-commerce sites become attractive targets and the emphasis turns from break-ins to denial of service and widespread virus attacks, the situation will likely worsen. Many early attackers were motivated by the challenges of breaking into systems, but there is an increasing trend toward attacks motivated by financial, political, and military objectives.

In the 1980s, most intruders were experts who discovered vulnerabilities and developed methods for breaking into systems. The use of automated tools and exploit scripts was rare. Now, anyone can attack a network using readily available intrusion tools and exploit scripts that take advantage of widely known vulnerabilities.

Today damaging intrusions can occur in a matter of seconds. Intruders hide their presence by installing modified versions of system monitoring and administration commands, and by erasing their tracks in audit and log files. In the 1980s and early 1990s, denial-of-service attacks were infrequent and not considered serious. Now, these successful attacks can put e-commerce-based organizations such as online stock brokers and retail sites out of business as evidenced in February of 2000.

ID has been an active field of research for about two decades. One of the earliest papers in the field is James Anderson's *Computer Security Threat Monitoring and Surveillance* [1] published in 1980. Dorothy Denning's seminal paper *An Intrusion Detection Model* [2], published in 1987, provided a methodological framework that inspired many researchers and laid the groundwork for commercial products. Despite substantial research and commercial investments, ID technology is immature, and its effectiveness is limited [3]. Within its limitations, it is useful as one portion of a defensive posture, but should not be relied upon as a sole means of protection.

The Intrusion Perspective

Defining what constitutes an attack is difficult because multiple perspectives are involved. Attacks may involve any number of attackers and victims. The attacker's viewpoint is typically characterized by intent and risk of exposure. From a victim's perspective, intrusions are characterized by their manifestations, which may or may not include damage. Some attacks may produce no manifestations, and some apparent manifestations may result from system/network malfunctions. Some attacks involve the (involuntary) participation of additional machines, usually victims of earlier attacks. For an intrusion to occur, it requires both an overt act by an attacker and a manifestation, observable by the intended victim, which results from that act.

A victim's view of an attack is usually focused on these manifestations:

- What happened?
- Who is affected and what were the consequences?
- Who is the intruder?
- Where and when did the intrusion originate?
- How and why did the intrusion happen?

Meanwhile, the attacker may have quite a different view:

- What is my objective?
- What vulnerabilities exist in the target system?
- What damage or other consequences are likely?
- What exploit scripts or other attack tools are available?
- What is my risk of exposure?

The goal of ID is to characterize attack manifestations so as to positively identify all true attacks without improperly identifying false attacks. The motivation for using ID technology may vary. Some users may be interested in collecting forensic information to locate and prosecute intruders. Others may use ID to trigger actions to protect computing resources. Still others may use ID to identify and correct vulnerabilities.

Dimensions of Intrusion Detection

IDS can be characterized in a variety of ways. Here, we choose system structure, sensed phenomenology, and a detection approach. The viewpoints are somewhat generalized, and a given IDS may combine structural components, sense multiple phenomenologies, or use multiple detection approaches.

® CERT and CERT Coordination Center are registered in the U.S. Patent and Trademark Office.

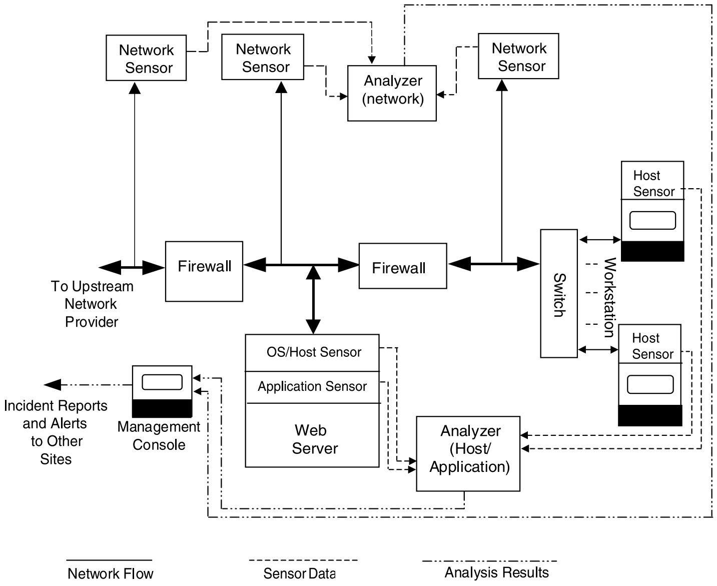

Figure 1. *IDS System Structure*

Figure 1 illustrates system structure and sensed phenomenology. The figure shows a small enterprise configured with firewalls to isolate its Web server. Computers configured as network sensors extract suspicious packets from the three main network segments and forward them to a network specific analysis station. The Web server and workstations are equipped with software to monitor suspicious interactions with the operating system and report them to a host-specific analysis station.

In addition, the Web server looks for abuses such as common gateway interface bin exploits that are specific to hypertext transfer protocol servers. The analyzers report to a management console that serves as the user interface for the IDS. The management console alerts the enterprise administrator who may, in turn, report intrusions to incident response organizations.

More elaborate configurations are possible. An analyzer may use inputs from any or all sensed phenomenologies in deciding whether or not an attack has taken place. Analyzer outputs may also be used as sensed data for other analyzers.

Intrusion Detection Approaches

ID can be viewed as an instance of the signal detection problem [4]. In this case, intrusion manifestations are viewed as the signal to be detected while manifestations of *normal* operations are considered to be noise. In classical signal detection approaches, both the signal and noise distributions are known. A decision process must determine if a given observation belongs to the signal-plus-noise distribution or to the noise distribution. Classical detectors use knowledge of both distributions in making a decision; but intrusion detectors typically base their decisions either on signal (signature-based) or noise (anomaly-based) detector characterizations. Each approach has strengths and weaknesses. Both suffer from the difficulty of characterizing the distributions.

In order for a signature-based IDS to detect attacks, it must possess an attack description that can be matched to sensed attack manifestations. This can be as simple as a specific pattern that matches a portion of a network packet, or as complex as a state machine or neural network description that maps multiple sensor outputs to abstract attack representations. If an appropriate abstraction can be found, signature-based systems can identify previously unseen attacks that are abstractly equivalent to known patterns. They are inherently unable to detect truly novel attacks and suffer from false alarms when signatures match both intrusive and nonintrusive sensor outputs.

Signatures can be developed in a variety of ways, from hand translation of attack manifestations to automatic training or learning using labeled sensor data. Because a given signature is associated with a known attack abstraction, it is relatively easy for a signature-based detector to assign names to attacks.

Anomaly-based detectors equate unusual or *abnormal* with intrusions. Given a complete characterization of the noise distribution, an anomaly-based detector recognizes as an intrusion any observation that does not appear to be noise alone.

Characterizing the noise distribution so as to support detection is nontrivial. Characterization approaches have ranged from statistical models of component/system behavior to neural networks and other artificial intelligence techniques to approaches inspired by the human immune system. The primary strength of anomaly detection is its ability to recognize novel attacks.

Its drawbacks include the need to train the system on noise plus the difficulties attendant in tracking natural changes in the noise distribution. Changes may cause false alarms while intrusive activities that appear to be normal may cause missed detections. It is difficult for anomaly-based systems to classify or name attacks.

Organizational Issues

Installing and effectively using intrusion detection systems on networks and hosts require a broad understanding of computer security. The complexity of information technology infrastructures is increasing beyond any one person's ability to understand them, let alone administer them in a way that is operationally secure.

An organization needs to fully appreciate the commitment required before deploying an IDS. Otherwise, the project runs the risk of wasting time, money, and staff resources in the initial phases of the IDS life cycle.

Before an organization makes an investment in security technologies, it must understand what assets require protection and the real and perceived threats against those assets. Threats can be characterized by the likely type of attack and attacker capabilities (i.e., resources and goals), and the organization's tolerance for loss of, damage to, or disclosure of protected assets.

Attacker motives can be arbitrary, based on curiosity or vandalism, or targeted to meet a specific objective such as revenge or gaining competitive advantage. Motives may make some forms of attack more likely than others. Gaining a competitive advantage may require compromising specific information such as a marketing plan. Each form of attack requires diverse detection strategies. For example, information retrieval is likely to be performed using a stealthy attack, while information corruption may require speed. Determining whether the potential attacker is inside or outside of the organization's infrastructure has a bearing on the type and placement of an IDS.

Often the most significant obstacle to the success of an information security improvement initiative is lack of management support.[2] In surveys conducted by security trade magazines during 1999 [5, 6], lack of management support was cited as one of the principle barriers to effective information security.

This is consistent with our experience at the SEI in implementing security improvement initiatives. Managers have many goals to meet, and they must often make tradeoffs. Security only becomes important when it impinges on the organization's high priority interests and reputation.

The deployment and operation of an IDS requires significant management support at the level of the corporate chief information officer and information security manager. Without this, the successful operation and use of this technology will be short-lived, sustained only by the interest of those internal champions who believe in its benefit. This is likely to last only until another high-priority item requires their attention.

Defense in Depth

ID is only one aspect of a layered defensive posture or *defense in depth*, which begins with the establishment of appropriate and effective security policies. Effective policies help ensure that threats to critical assets are understood, managers and users are adequately trained, and intrusion response actions are defined. A good security policy places ID into its proper perspective and context.

Establishing a layered security architecture is advantageous whether an IDS is deployed or not. In addition to formulating a security policy, the essential steps are:

- Implementing user authentication and access controls.
- Eliminating unnecessary services.
- Applying patches to eliminate known vulnerabilities.
- Deploying firewalls.
- Using file integrity checking tools such as Tripwire.[3]

Since most real-time commercial ID systems base their detection approach on known attempts to exploit known vulnerabilities, an administrator's time is often better spent minimizing vulnerability by applying patches or other security measures. Detecting and responding to penetration attempts that cannot succeed (such as UNIX-specific attempts against a network of Windows machines) is not an effective use of resources except as an indication of threat level.

The IDS Life Cycle

Vendors frequently release new IDS products and aggressively compete for market share. Evaluating these new systems is crucial, yet there is a lack of credible, comprehensive product evaluation information. Hiring and retaining personnel to competently administer security in general, and intrusion detection in particular, are increasingly challenging. Rapid changes in information technology make it difficult for an organization to implement an effective, long-term security strategy.

- **Evaluation and Selection:** If an organization plans to acquire an ID system, it should consider the resources available for its operation and maintenance and choose one that meets its needs within these constraints. This is difficult because there are no industry standards against which to compare ID systems. The new product cycle for commercial IDSs is rapid, and information and systems quickly become obsolete. Northcutt recommends use of product guides that are updated at least monthly [7].

  Marketing literature rarely describes how well a given IDS finds intruders. Neither does it tell how much work is required to use and maintain that system in a fully functioning network with significant daily traffic. IDS vendors usually specify which prototypical attacks can be found by their systems. However, without access to deployment environments, they cannot describe how well their systems detect real attacks while avoiding false alarms.

  Topics to consider include detection and response characteristics, use of signature- and/or anomaly-based approaches, accuracy of diagnosis (false alarm rate), ease of use, effectiveness of user interface, quality of vendor support, etc. A paper by Amoroso and Kwapniewski [8] provides guidance in selecting an IDS. The Computer Security Institute[4] has a number of relevant Web pages, including a list of questions for IDS vendors. Setting up a facility to objectively compare IDSs will be prohibitively expensive for all but the largest potential users, and some third-party or industry-sponsored effort is needed.
- **Deployment:** Issues to address here include placement of sensors to maximize protection for the most critical assets, configuring the IDS to reflect security policy, installing appropriate signatures and other initial conditions, establishing forensic procedures to preserve evidence for possible prosecutions, and determining when (if ever) and what automatic responses are allowed. Procedures must be developed to handle IDS alerts and to consider how alerts are to be correlated with other information such as system or application logs.
- **Operations and Use:** Once an IDS is deployed, it is necessary to monitor the system and to respond to the reported alerts. This means establishing roles and responsibilities for analyzing and acting on alerts, monitoring the outcomes of both manual and automatic responses, etc.

  IDSs themselves are logical targets for attack [9]. Smart intruders who realize that an IDS has been deployed on a network they are attacking will likely attack the IDS first, disabling it or forcing it to provide false information (distracting security personnel from the actual attack in progress). In addition, many commercial and research ID tools have security weaknesses resulting from flawed design assumptions. These may include failing to encrypt log files, omitting access control, and failing to perform integrity checks on IDS files.
- **Maintenance:** Activities include installing new signatures as they become available, as well as installing periodic IDS upgrades. Sensor placement should be revisited periodically to ensure that system or network changes have not reduced the effectiveness of the IDS.

Use of technology alone is not sufficient to maintain network security. An organization needs to attract, train, and retain qualified technical staff to operate and maintain ID technologies. In today's market, there is a decreasing availability of qualified intrusion analysts and system/network administrators who are knowledgeable about and experienced in computer security.

ID Technology

Commercial ID technology (such as ISS RealSecure[5] and Tripwire) is evolving and is often dynamic to the point of instability. New vendors appear only to be absorbed by others. One consequence of this rapid change is that product lists, surveys, and reviews quickly become outdated. Because of the volatility of the market, we advise using a Web search to locate current products, reviews, etc.

Commercial product literature is generally weighted towards marketing, which often makes it difficult to determine the product's functionality and detection approach. Virtually no commercial literature addresses issues such as frequencies of false alarms, missed detections, or the system's sensitivity to traffic loads.

Public domain systems, such as Shadow[6] and Snort[7], are unlikely to have the same level of support as commercial systems, so a higher level of technical expertise is required to install and manage them. However, the effort required results in the payoff of a better understanding of ID and its strengths and limitations.

Based on several limited experiments, we found that commercial ID tools are easier to install than public domain tools. None of the tools had an understandable, easy-to-use configuration interface. However, the commercial tools did employ graphical interfaces while the public domain tools did not. All of the tools required labor-intensive signature tuning. We found no indication of any integration between vulnerability scanners and configuration interfaces despite the fact that most IDS vendors sell vulnerability analyzers.[8] The configuration process could be made simpler if signatures associated with detected vulnerabilities could be loaded automatically.

The commercial products that we installed did not provide sufficient supporting data (such as raw packets) to verify events they claimed to detect. The use of proprietary algorithms and signatures made it difficult to determine why an alert occured. Distinguishing between intrusions and false alarms required manual investigation. In most cases, the analyst had to examine logs for supporting evidence.

IDS products based on current signature-based analysis do not provide a complete ID solution, but do produce useful results in specific situations and configurations. The majority of intrusion detection systems we examined appeared to provide good capabilities for enhanced network monitoring and might be more useful in this capacity than for intrusion detection.

Conclusion

ID technology is immature and should not be considered as a complete defense. However it can play a significant role in an overall security architecture. If an organization chooses to deploy an IDS, there is a range of commercial and public domain products to choose from with varying effectiveness and deployment costs. Since any deployment will incur ongoing operation and maintenance costs, the choice should be made considering the full IDS life cycle.

When an IDS is properly deployed, it can provide warnings indicating that a system is under attack, even if the system is not vulnerable to the specific attack. This warning can be used to alter the defensive posture of the installation to accomplish greater resistance to attack. In addition, an IDS may be used to confirm secure configuration and operation of other security mechanisms such as firewalls.

Acknowledgements

The authors wish to thank the members of the CERT/CC staff who contributed to *State of the Practice of Intrusion Detection Technologies,* on which this is based: William Fithen, Jed Pickel, Ed Stoner, James Ellis, Eric Hayes, Jerome Marella, and Bradford Willke. Stefan Axelsson of Chalmers University provided insight into the characterization of ID as a signal detection problem. Many members of the ID community have influenced our thinking on this topic. Among them are Ed Amoroso, Dick Kemmerer, Gene Kim, Jay Larrew, Teresa Lunt, Roy Maxion, Phil Porras, and Mark Woods.

References

1. Anderson, James P., *Computer Security Threat Monitoring and Surveillance*. James P. Anderson Co., Fort Washington, Pa., 1980.
2. Denning, Dorothy E., An Intrusion Detection Model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, February 1987.
3. Allen, J., Christie, A., Fithen, W., McHugh, J., Pickel, J., and Stoner, E., *State of the Practice of Intrusion Detection Technologies*. CMU/SEI-99-TR-028, Carnegie Mellon University, Software Engineering Institute, January 2000. Available at www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html
4. Egan, James P., *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.
5. Larson, Amy K., Global Security Survey: Virus Attack. *Information Week*, July 12, 1999.
6. Briney, Andy, Got Security? *Information Security Magazine*, July 1999.
7. Northcutt, Steven, *Network Intrusion Detection*. New Riders, Indianapolis, Ind., 1999.
8. Amoroso, Edward and Kwapniewski, Richard, A Selection Criteria for Intrusion Detection Systems. *Proceedings of the 14th Annual Computer Security Applications Conference*. IEEE Computer Society Press, December 1998.
9. Ptacek, Thomas H. and Newsham, Timothy N., *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection*, 1998. Available from www.snort.org/IDSpaper.pdf

Notes

1. For a more complete discussion of this subject, see reference 1.
2. One individual told an author of this article that he obtained management sponsorship by demonstrating how easy it was to break into his manager's confidential computer files. This approach is not necessarily recommended, but at least in this case, appears to have been effective.
3. See www.tripwire.com—both commercial and public domain versions are available.
4. See www.gocsi.com
5. See www.iss.net
6. See www.nswc.navy.mil/ISSEC/CID
7. See www.snort.org
8. ISS is integrating their IDS and vulnerability scanner.

About the Authors

Julia Allen is a senior member of the technical staff (MTS) working in the CERT/CC area at CMU's Software Engineering Institute. She has an master's degree in electrical engineering from the University of Southern California.

Voice: 412-268-6760
E-mail: jha@cert.org

Alan Christie has spent the last ten years at the Software Engineering Institute focusing much of his attention on process, technology, and security issues. He is a senior MTS and has a doctorate degree in nuclear engineering from Carnegie Mellon University.

Voice: 412-268-6324
E-mail: amc@cert.org

John McHugh is a senior MTS at the Software Engineering Institute (CERT/CC). He has a doctorate in computer science from the University of Texas at Austin and is a past chair of the IEEE Technical Committee on Security and Privacy.

Voice: 412-268-7737
E-mail: jmchugh@cert.org

CMU, SEI
4500 Fifth Ave.
Pittsburgh, Pa. 15213
Fax: 412-268-6989
Internet: www.sei.cmu.edu, www.cert.org

Politics in the Red Zone

It is 1:00 a.m. Wednesday morning, Nov. 8, 2000. I am lying in a St. Louis hotel room glued to the television set watching the presidential election turn into a soap opera. For a minute I thought the networks hired software engineers to make their projections. Then I reasoned, no, they got more than half of the states right.

By the time I logged in a couple of hours of sleep, a short run, and a refreshing shower the politicians had spun the results up tighter than a NASCAR tachometer approaching the red zone. That diverted my concerns to more insipid issues like what does the red zone on a tachometer really mean? How do automobile manufacturers determine it? Why do engineers disdain their managers?

The red zone theoretically is the engine speed at which you are in real danger of having your engine fall apart. However, we all know that definition is not entirely true. I've ventured into the red zone a time or two and all I got was a citation from a local law enforcement officer asking for a donation to the policeman's ball.

Obviously engine manufacturers realized whimsical people, like myself, touch red zones before believing them to be hot. So they built a buffer between the advertised red zone and the real point of no return.

In the early days of the automobile industry, the red zone was determined by trial and error. When a blown engine was brought into the shop, the service manager would ask the driver, if still coherent, "Did you get a look at the RPMs just before the piston ripped through the hood?"

As the industry flourished, the red zone was measured in the laboratory using sophisticated monitoring equipment. Bottom line is you really should not spend much time in the red zone. Your Yugo is more reliable, effective, and economic if you stay out of the red zone.

Wouldn't it be nice if software engineers came equipped with a stress meter and red zone indicating when the engineer might fall apart (like a mother who discovers her daughter didn't make the cheerleading squad)?

As long as we are wiring engineers—as if we are not wired enough—we could add a green zone indicating peak performance, a yellow zone for idleness, and an orange zone for when they have stalled.

Unfortunately colleges and universities are not rolling out engineers with factory-installed stress meters. No dials, no knobs, no zones, just a handful of offers and more bravado than Pavarotti on opening night.

It appears that some managers, left over from theory X, are toying with the auto industry's early trial and error approach. "Excuse me, what was Gilbert's workload just before his head ripped through the monitor? I don't know but police suspect Mountain Dew may have been involved."

Somehow I doubt this approach will last. With competition for talented engineers fierce, a manager employing this crash and burn tactic will be disenfranchised faster than a senior citizen voting in Palm Beach County, Fla.

On the other side of the scale, cautious theory Y managers rev up their engineers but forget to put them in gear. More concerned about being accepted as a manager than producing a product, they are long on stories, jokes, and antidotes, and short on direction and resolve to lead à la Rush Limbaugh.

So does that leave us with a theory Z management approach? No way! Theory Z managers waffle more than Al Gore with his campaign strategy. It's too firm, too soft; I'll decide, no you decide … this waffling is an engineer's nightmare. Software engineers have a hard enough time with requirements. Adding a manager that flops on decisions daily is like asking Robin Williams to teach yoga.

What can a manager do? How about trying something completely radical, foreign to engineer and manager alike—something that makes your management upper lip sweat when mentioned? How about talking with the engineers? Not talking at engineers but holding a meaningful dialogue where you listen and try to understand. Spend a day working with the engineer. You do remember engineering don't you? You do know how to listen don't you? You do know the difference between discussion and dialogue, don't you? Can you fill out a butterfly ballot? Never mind.

—*Gary Petersen, Shim Enterprise Inc.*