COMPUTER SECURITY JOURNAL

VOLUME XVIII, NUMBERS 3-4, SUMMER/FALL 2002

DOUBLE ISSUE

Forensics and Investigation:
Taking Testimony Seriously
Get prepped for your day in court—page 1

Encryption:
Assessment of Public Key Infrastructure: Methodology and Issues
Got PKI? Here's a framework for ensuring its quality—page 15

Attacks and Countermeasures:
Productive Intrusion Detection
Alternative approaches to IDS—page 23

Industry Insight:
2002 Australian Computer Crime and Security Survey
The state of cybercrime down under—page 35

Issues and Trends:
Analysis of Riptech's Internet Security Report for Q3-Q4 of 2001
Jon Callas has some gripes—page 43

SPECIAL REPORT: MICROSOFT SECURITY

The Future of Microsoft Security
Will you be safe if Redmond locks down your desktop?—page 53

TCPA/Palladium FAQ
Ross Anderson's take on the Wintel trusted platform strategy—page 63

.NET vs J2EE Security
How does Microsoft's next big thing stack up alongside Java?—page 73

A Timeline of Microsoft Security
From Altar BASIC on—page 86

Productive Intrusion Detection

By Jennifer L. Bayuk

More than ever before, security officers are prominent in information systems management. They are accountable for delivering value to the organization. They are more often held to the same standards as peers in systems development, quality assurance, and operations support. The requirement for a security staff has rarely been questioned, but the value of its contribution is ever more critically assessed.

The traditional role of an Information Security Officer (ISO) is to plan and maintain access control mechanisms. That is, the ISO makes sure only authorized users get in. But all the press on hacking and intrusion detection has raised expectations. Not only should the ISO keep the hackers out, but also detect security vulnerabilities created by authorized users. There seems to be an assumption by upper management that the ISO will continue to prevent unauthorized access, but also somehow control all authorized user activity in the information systems environment.

To understand just how difficult a job that is, imagine being the person in charge of physical security at a police station. Now imagine that a police officer who is authorized to carry a weapon brings it into the lobby and starts shooting. Your job as a security officer is to stop the perpetrator before anyone is hurt.

This paper views the intrusion detection problem from the point of view of the person whose job it is to solve it, immediately, before anyone gets hurt (the ISO). It defines the intrusion detection problem, reviews practical attempts to address it, and proposes an alternative approach to its solution. The approach is specification-based and quantitative. It provides metrics for determining the extent to which an organization is successful in its implementation.

The Challenge

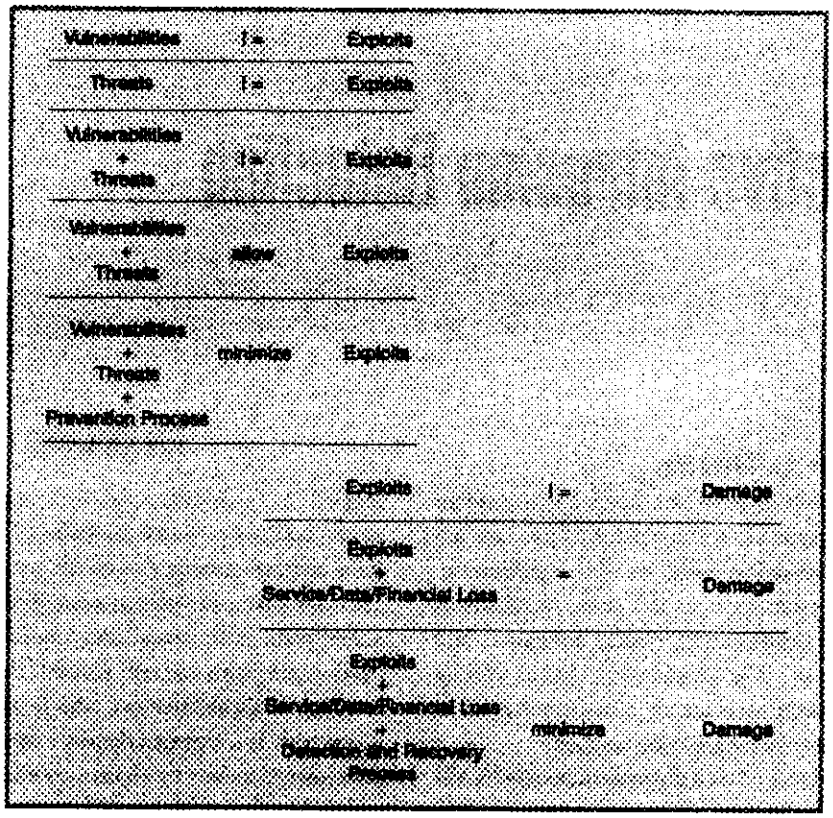
A good ISO will have a plan for dealing with every potential computer misuse scenario [15]. A good ISO will have identity associated with all system users. A good ISO will have a purpose associated with every network access path, even anonymous access paths. Unauthorized access paths are not allowed. Intrusions therefore include: an authorized user who is using an authorized access path in a way that does not correspond to its purpose system activity by an unauthorized person who has stolen the account of an authorized user (thus appearing to the ISO as an authorized user) use of an access path for which there is no business purpose, and thus is unauthorized, that is, "the dreaded new vulnerability for which there is yet no fix"

A good ISO will develop an incident response process to cover these misuse scenarios.

However, not every intrusion requires a response. Even the existence of "the dreaded new vulnerability for which there is yet no fix" may not concern the ISO. It is a well-worn theorem to the average ISO that vulnerabilities do not equal damage. This is because neither vulnerabilities nor threats equal exploits. It takes a threat combined with vulnerability to even introduce the possibility of exploit. The ISO carefully measures the probability that the threat will be enacted and devises prevention controls accordingly. But suppose, despite the preventive measures of the ISO, an exploit was committed that did cause damage. Even if it were enacted, the probable damage may be such that detection is not a cost-justifiable option. The logic of the ISO theorem is demonstrated in *Figure 1*.

To illustrate, consider that to describe a traditional intrusion, one must first define some entry point. When one defines intrusion with respect to a house, is the

Figure 1 **The ISO Theorem**



entry point the sidewalk, the yard, and the doorstep, as well as the front door? If a stranger opted to hang out on your front walk, would you call that stranger an intruder? We understand that is it not unusual for people to get lost, to be ignorant or unbalanced, or to just be deliberately obnoxious. We all tolerate some level of trespass in allowing for such behaviors.

With respect to information systems, the entry point could be anywhere in the set of systems belonging to the organization, or in the network connectivity constructed and maintained for the purpose of accessing the systems. From the street, researchers probe subnets, marketers spam servers, and script kiddies bang on web pages. From the sidewalk, developers spew applets, deployment groups create portals, and management systems scan networks. An intrusion does not concern an ISO unless it is accompanied by a threat to information system services lying beyond that entry point. Productive intrusion detection draws a clear line between random or harmless acts of trespass and intentional successful exploitation of vulnerabilities that result in damage.

Responding to the Challenge

The Academic Approach

Just about every formal reasoning technique or data analysis method has been tested in pursuit of intrusion detection's ultimate solution. In a recent survey of the field of intrusion detection to date, John McHugh concludes, "There is no underlying theory that relates detection approaches to detections or that allows useful predictions to be made concerning the relative powers of various techniques.....We suspect that none of the current research approaches will show themselves capable of reaching predefined goals and that incremental improvements will be limited." [11] Work continues on new approaches and extensions of various algorithms, but critical reviews of the false positive rates are based on solid technical foundations and have gone unanswered.[12,13]

Pessimism is not limited to intrusion techniques, but also extends to intrusion data. In his survey,

McHugh makes the point that early intrusion detection literature assumes that some sort of audit data will be sufficient to indicate intrusions, but provides no basis to justify the assumption. Nevertheless, this assumption has become more egregious over time. According to Ed Amoroso, first and foremost a researcher, but also a security officer at AT&T, the ISO is expected to devise operational processes that combine all manner of intrusion indicators including:

- Repetition of suspicious action
- Mistyped commands or responses during automated sequences
- Exploitation of known vulnerabilities (using scanning tools)
- Directional inconsistencies in inbound or outbound packets
- Unexpected attributes of some service request or packet
- Unexplained problems in some service request
- Out of band knowledge about an intrusion (e.g from hacker web pages)
- Suspicious character traffic (e.g. unencrypted traffic in a secure environment)[1].

Amoroso's list is not even the most comprehensive, but seems limited through his affinity with the history of the field. From Denning's time, intrusion detection systems have concentrated almost exclusively on behavioral activity as a source of intrusion data. Newcomers to the field include configuration checking tools, honey pots, performance monitoring commands and empty log files in the list of sources of intrusion evidence.[7]

In other words, the ISO is now expected to classify, store, and attempt to correlate virtually every byte in the company's network. An ISO must master the art of selecting a combination of data sources, organizing them into a coherent framework, correlating data from different sources, and automating analysis that yields "incidents." But there are no published successful precedents. The academic approach thus responds unsuccessfully to the challenge of intrusion detection.

The Commercial Approach

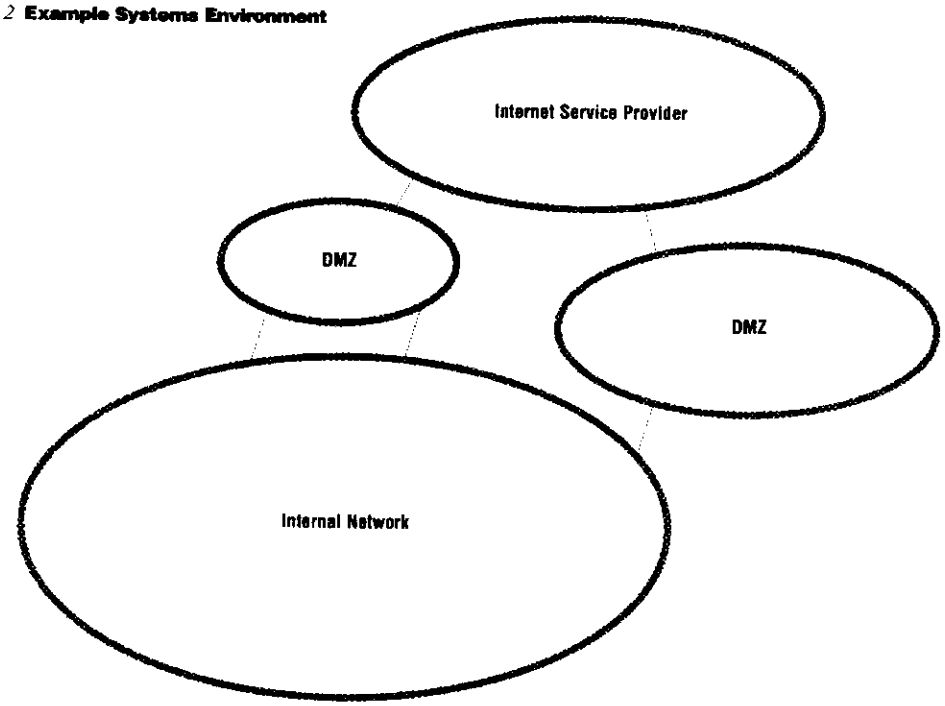
Commercial intrusion detection systems seem prevalent, but became widely available only in the late 1990s. Lance Eliot wrote in 1992, "Without alerting the thief, an expert system detected the unusual activity on the computer system. The expert proceeded to lock out accounts and encrypt key data that contained sensitive corporate details. Meanwhile, the expert system sent a fake broadcast to all accounts indicating that the computer would be going down for routine maintenance in 15 minutes. The expert system then dialed the operations manager, alerting that someone had broken into the system. Although such expert systems don't yet exist, crimebusting AI systems are being developed and tested."[6]

Lance's optimistic forecast was based on the plethora of research in anomaly-based intrusion detection that emerged following Denning's work in the mid 1980s.[5] Yet 10 years later, even the most enthusiastic of product reviews admit that the best systems detect less than 80% of attacks under realistic loads, and are forebodingly silent on the topic of false positive rate.[2,3] The ISO today has no hope that any technology available for intrusion detection meets the optimistic projections made by Eliot. No existing technology will identify all known attacks and at the same time keep the number of false alerts to a level acceptable to an operations environment.

The literature targeted at the ISO expresses the general consensus and the corresponding advice that intrusion detection is a process, requiring diligent operations support and sophisticated evaluation of results.[4,10,12,13,17] One popular "how-to" guide to Intrusion Detection describes it as an "art."[17] Commercial products and corresponding consulting projects focus almost exclusively on the methods used by some automated system to collect data and facilitate its correlation. However, all include a disclaimer about the reliability of the data collected and the ability of the "security administrator" to properly interpret it.

Of course, tools for the collection and correlation of intrusion-related data are rapidly being developed. They are designed to record events from multiple sources into one data base and allow a user to query by specific fields. Though all such systems are now proprietary, there is at least one effort to standardize the taxonomy for events: the Internet Engineering Task Force's Intrusion Detection Exchange Protocol.[19] Unfortunately, the framework has been slow to emerge. Moreover, it requires all compliant systems to adopt a new language and messaging structure. Because similar infrastructure exists in deeply embedded operations management technology, the cost-benefit of using any new framework, however promising, will be a major issue for the ISO to address. For now at least, it is up to each ISO to define their own data repositories and

Figure 2 **Example Systems Environment**



model their processes around the communication mechanisms currently available. The commercial approach provides tools rather than solutions, and thus responds incompletely to the challenge of intrusion detection.

The Productive Approach

The emphasis on data correlation nevertheless shows that the commercial approach agrees with the more recent academic view that the data repository is key. The Practical Intrusion Handbook advises, "You should focus less on the technical limitations in your intrusion detection tools and more on defining data and misuse that actually matters in your environment."[17] The integrity of the data must be beyond doubt.

The ISO must also be able to maintain that, if there is an intrusion in the network, then the selected data will provide evidence of it. The products accomplish this by using upfront definitions of how an intrusion is evident in data. They can be used to produce metrics. The number of machines you manage in a network management system is one measurement, call it A. The number of reports of security alerts is another, say B. The number of those reports that signified actual intrusions rather than maintenance activity is a third, C. Many would immediately recognize the value of the metric C/A. The number of intrusions per system should grow lower as systems grow more secure. From these measurements, an ISO may also define a productivity metric: B-C/A. As false positives grow smaller, that number should approach zero.

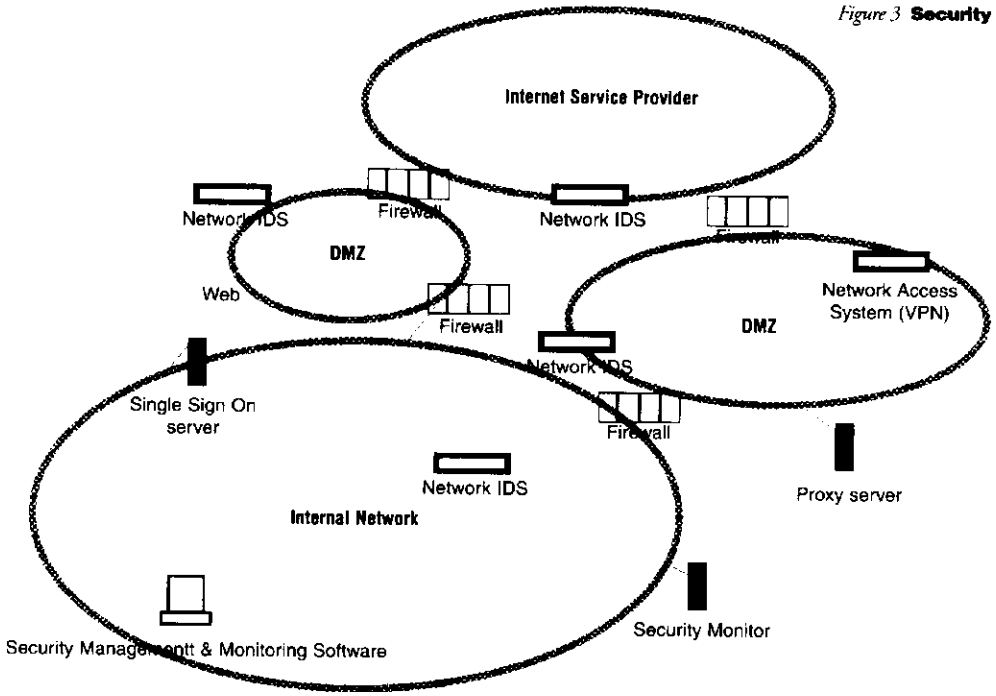
Assuming that potential intrusion incidents may be identified, the ISO must devise and manage processes whereby the incidents are properly analyzed to determine whether they warrant response or further investigation. This analysis should be considered part of the "intrusion detection system." The sections that follow demonstrate that a dual focus on defining an incident at the same time as prescribing how to analyze it will yield productive results.

A Defect Elimination Strategy

If the assignment to deploy an intrusion detection system is to be treated as any other system engineering task, the appropriate first activity is a requirements analysis. As both commercial and academic sources agree that intrusion detection is a good deal of data and a decision on what constitutes an intrusion, it would seem that a good way to start is to define intrusions in terms of data analysis.

Data must be gathered from the systems environment. An example of a common depiction of a sys-

Figure 3 **Security Components**



tems environment is presented in *Figure 2*. It shows a network logically divided into four security realms: one public Internet, two demilitarized zones (DMZs) and one internal network.

Figure 3 demonstrates a common approach to meeting the intrusion data collection requirement. The approach makes use of security systems that are set up to enforce a security policy. A central monitor collects data from these systems. The data collection is designed to provide evidence that the security systems are:

- Properly configured
- Producing expected audit trails
- Reliable and robust

It also checks network traffic for known intrusions and reports those back to a central security monitor. The assumption is that if all security components are working, then the environment must be secure.

The opposite of this approach is to start, not from the logical clouds, but from the ground up. This approach is to design a secure infrastructure based on individually secured components. Each component is individually equipped to produce data that provides evidence that the component itself is:

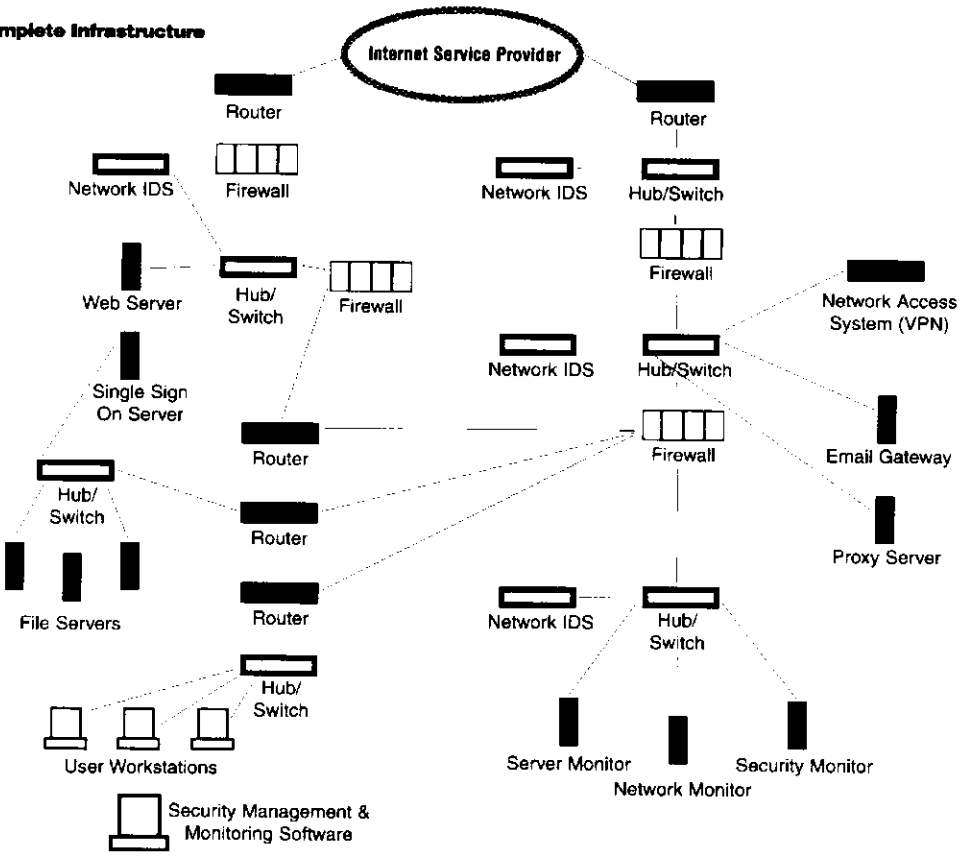
- Properly configured
- Producing expected audit trails
- Reliable and robust

Figure 4 illustrates this approach. Of course, a glance at *Figure 4* shows how obvious it is that methodology based on *Figure 3* could miss security-critical aspects of the infrastructure. For example, the router that connects the external firewall to the Internet Service Provider is a highly significant security control point.

Moreover, *Figure 5* demonstrates that the major flaw in *Figure 3* is that the infrastructure under the clouds can change, rendering the security monitoring of the originally planned security components inadequate. In this case, an alternate Internet Service Provider, perhaps added for disaster recovery, is inadvertently placed on the wrong switch, allowing Internet traffic to bypass the external firewall. Further, an even more problematic flaw is that the security products themselves may be faulty. Even in a well-designed and fastidiously implemented infrastructure, one may find security loopholes that permit intrusion.

The trick in collecting the right data to detect intrusions requires a gestalt shift away from the image of *Figure 2* to the image of *Figure 4*. Software designed for security management and monitoring often intentionally obscures the complexity of the information it is gathering. It claims to add value by reducing the security environment to selected parameters from a few choice components that are considered critical to composing the security profile of the infrastructure. But the ISO understands that the simplification may not be

Figure 4 **Complete Infrastructure**



ideal. Router monitoring may not be a feature of the security management station and routers may be fortified with only RADIUS and syslog. Access control servers at their best are only as good as user administrators and every good ISO knows administrative processes are subject to exception. Taking very few "security-specific" data points is leaving far too much of the security of the environment to systems and processes that were not designed with security in mind.

Here is where it helps to have first defined an intrusion. This definition can yield a determination concerning how an intrusion is evident in data. A recent survey of approaches to measuring security reveals that the security data gathered varies according to the purpose of the person measuring.[3] One observation in that survey was that all of these methods acknowledge the role of an "investigator." The investigator uses predefined criteria to assess the security of a given environment. The fact that an investigator may assign quantitative weights or values to his or her assessments does not change the fundamental qualitative approach.

However, there are two types of "investigators." One is an evaluator. The evaluator uses pre-defined criteria to gather data with which to assess the security of a given environment. One example is a collaborative hack, and the pre-defined criteria are the tools and techniques in the repository of the investigators. Another example is an internal audit, and the predefined criteria are an auditor's list of best practices. Because these evaluations are done from the perspective of a given point of view, an ISO must understand that the investigation is fundamentally a qualitative approach to what we would rather see as a quantitative problem.

The other type of investigator is an automator. The automated approach does not preclude value judgment in determining which data to gather. It just requires that these qualitative judgments be made in advance of the evaluation. An ISO may pre-establish formulas that should provide evidence of intrusion, automate the collection of the variables, then plug the variables into the formulas. This approach restricts the evaluative element of the intrusion detection process to formula-creation activity. It removes individual judgment from the data collection and response decision itself. Every positive result from a formula requires investigation.

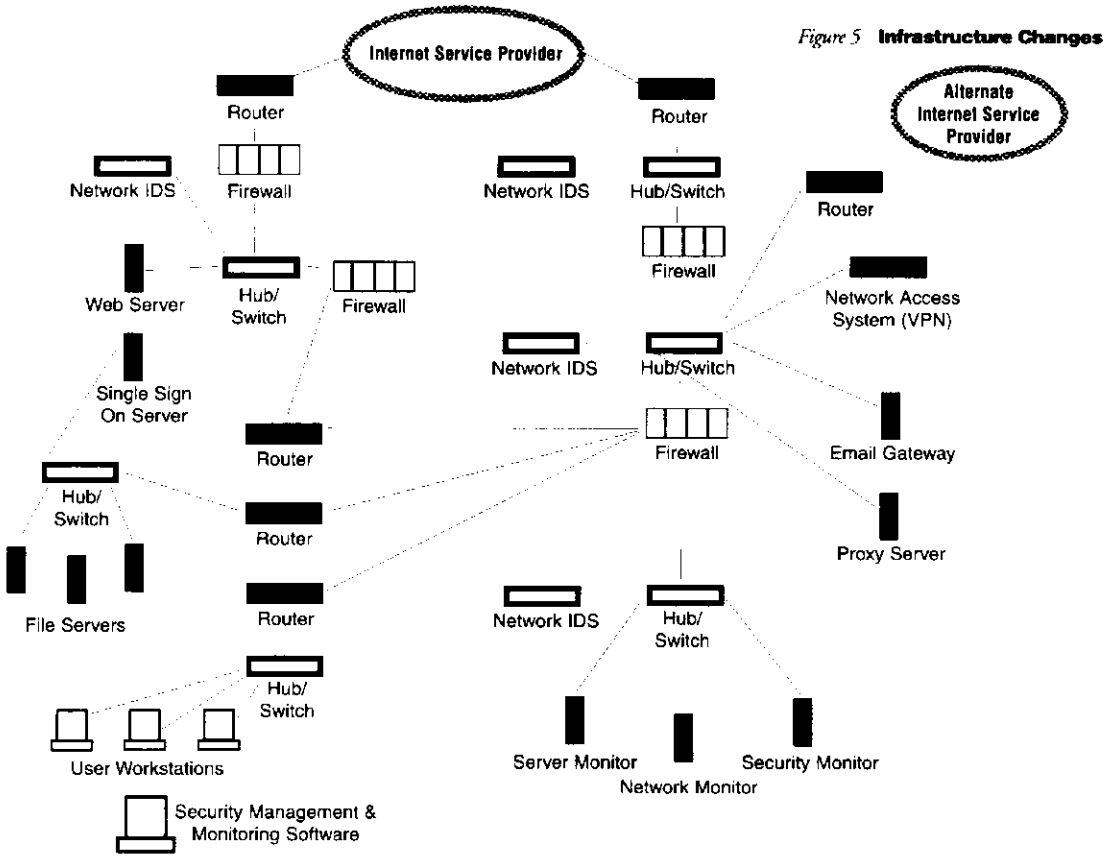


Figure 5 **Infrastructure Changes**

Of course, the hard part is to pre-establish the formulas. Many vendor products gather information according to formulas that are components of intrusion evidence. But rarely do vendor formulas cover all of an ISO's security goals, and may have components that are irrelevant to a given environment. In the formula creation process, the ISO may purposely ignore evidence of random or harmless acts of trespass and focus only on intentional successful exploitation. These decisions are difficult and may constantly evolve. However, once established, the evidence-gathering formulas allow an ISO to follow a methodology that is successful in non-information security endeavors. That is, a defect-elimination model.[16]

Define security defects as confirmed evidence of intrusion. A defect may be an explicit policy violation, evidence of data corruption, an unexplained configuration change, or a log indicating bypass of security mechanisms. It is evidence that the intruder was inside, regardless of whether it is possible to identify the entry point. Regardless of what technology may be used to secure systems, automated measurement of evidence of security defects may be applied.

The Productivity Payoff

The benefit of a defect elimination approach is that, where data clearly provides evidence of a defect, there is no lost productivity in the "false positive." For example, suppose that policy dictates a machine's security configuration variables be set to given values. A simple periodic check on the values could result in a policy violation alert. Though the values may have been changed due to system malfunction rather than due to an intruder, such an event does not constitute a false positive from a security investigation standpoint. Action must be taken to bring the system back into compliance. Policy compliance is a clear payoff for the time and energy spent in deploying and operating an intrusion detection system.

The key to success in this approach is applying a definition of defect to system variables. Static configuration files are easy, but if a defect definition is specific to the point of defining program behavior, more sophisticated tools will be required. Research in this area is not abundant, but exists.[9] In addition, commercial sandboxing or debugging technologies may be employed to provide verbose logs of program behavior. For ex-

ample, suppose security policy dictates that telnet should only be used in the context of emergency maintenance. There are several methods that an ISO could employ to ensure that security staff was alerted to each use of telnet:

- Install "sandboxing" security software that intercepts authorization calls at the kernel level and checks the calling program (set to alert, rather than block, the activity)
- Install a wrapper around the telnet daemon that sends an alert each time it is started
- Monitor the processes on the machine on a polling basis and alert if telnetd appears in the process list
- Once information security staff is alerted to the use of telnet, a predefined response procedure should be able to identify the source, compare it with an authoritative source for information on emergency maintenance, and confirm whether the activity constitutes an intrusion. Such an alert is not a lost productivity from an ISO's perspective as it provides a management control point: a verification that the telnet utility is not used on a non-emergency basis. This is another example of the intrusion detection system doubling as a policy compliance measurement device. The approach can turn an intrusion detection system into security productivity for the information security staff.

To measure correct configuration of a product, one must use a tool that is not part of the product. If an access control product has been bypassed, one may not see any evidence of the event in the activity logs of the product itself.

Where policy violations are not so clear, an intrusion detection system may provide a warning that will require a more technical investigation to determine whether a security defect is or is not evident. It may even be that the information security staff is not the ideal recipient for the initial alert. In some cases, the ISO may know of a potential security event, but refer the initial data analysis to another group to get a better diagnosis.

The JiNao IDS provides a very good example of this type of event.[8] JiNao detects intrusion in routers using Open Shortest Path First (OSPF), a routing protocol that relies on distributed authority for advertising the shortest route. A potential threat to systems comprised of routers using OSPF is that one compromised router may allow an intruder to advertise false routes. In one such scenario, the router that is the true authoritative source for a given route will continue to advertise the authorized route with a higher priority flag set. This phenomenon, called "fightback," is easily detectable. The JiNao system recognizes fightback and sends an alert. Every instance of fightback must be investigated because if it was not caused by an intruder, it indicates a potential system misconfiguration or malfunction. Hence there is no operations time wasted in investigation. This is especially true if the router support staff gets the first alert, and only passes it to the information security staff after it has been diagnosed as an intrusion. The distributed diagnostic activity increases productivity for the information security staff, as there will be absolutely no false positive sent to the information security staff.

Note the role of the ISO in defining the event and referring it to the network operations group. Where such initial analysis is distributed, the distribution must be clear and there must be accountability for responding to the security alert. Such accountability is achieved by linking the system that detected the intrusion to a trouble tracking system in use by the operations group. The ISO must have access to a report that correlates alert data with the operations trouble tracking system. All components except the report may be entirely outside of the line management responsibility of the ISO, but as long as there are systems in place that assure policy compliance with the system components, the data feed from the trouble-tracking system can be considered an extension of the intrusion detection system.

In this type of intrusion definition and response scenario, evidence initially gathered on potential intrusion is not necessarily a security defect. The end result of the intrusion detection system includes the analysis process. This type of intrusion detection system lets the information security staff spend more time on security defects than on operational issues.

Integration Issues

Just how these alerts are to be integrated into the infrastructure is more difficult for some technologies

than for others. Compare a component of security infrastructure to a program module. The module is useless unless the entire program is compiled and all modules are both linked and available. A diligent programmer will design test suites for each component individually and also for the program as a whole. If one component is not well designed, or if its tests are not well designed, a bug may be present in the component that is only recognizable when the program is operating as a compiled unit. At that point, it may happen that the flaw is uncovered by a debug messages from a different component. Thus, the more integrity checking and audit features any one component has, the more its functionality can contribute to the overall test results.

Note that defect measurement must verify all security requirements fulfilled by a security infrastructure component in a way that does not depend on operating the product itself. To measure correct configuration of a product, one must use a tool that is not part of the product. If an access control product has been bypassed, one may not see any evidence of the event in the activity logs of the product itself. For example, say a network intrusion detection device in a DMZ of Figure 4 sends an alert that there is unexpected activity in the DMZ. Also say that a diligent check of activity logs from the other security software reveals nothing. Yet integrity checking software at the operating system level of a web server reveals a corrupt security software configuration file. Security measurement tools that are not part of the corrupt security component have led to the discovery of the defect. The extent to which the security of the environment can be thus measured should be a product evaluation criterion.

Yet, when each security component is assessed independently for the ability to measure correct configuration and detect intrusion, it is common to find security product loopholes. Features meant to satisfy one security requirement actually introduce vulnerabilities being measured with respect to another. There is no shortage of examples. In applying intrusion detection data collection requirements to security software deployment efforts, it is possible to find many examples of huge, heavily funded software companies whose flagship security products:

- have no feature by which a user list can be exported to a non-proprietary format
- have no documentation that shows how configuration data displayed in the GUI corresponds to the configuration read into the product's software engine
- have proprietary formats for complicated rule bases, and only screen scrolling methods of determining correctness of a configuration
- allow backdoor cleartext passwords to administer the product via an open network
- have no way to just log successful access attempts, just failed access attempts, or both
- make use of ODBC compliant databases where access to the database itself is cleartext user ID and password in readable startup files
- rely on syslog for logging but do not have buffer strategies for dealing with syslog server unavailability
- alert via snmp with no receipt verification, and use write community strings on open networks

Vendor response to these issues is universal. They are following industry standard architecture practices. The ISO is left to conclude that industry standard security requirements do not yet include robust features that can be used to verify that a product is correctly configured and/or is not being misused.

Where these features are lacking, the ISO has three choices: to work around the deficiency by designing compensating controls, to eliminate the architecture components that cannot provide evidence useful in detecting intrusions, or to augment the existing architecture to provide new monitoring functionality. Catherine Meadows describes these choices as new security paradigms:[14]

- *Live With It:* apply available patches but do not attempt to change the underlying system architecture
- *Replace It:* replace the deficient component
- *Extend It:* add components to extend the system's capabilities to operate securely

Meadows observes that the first paradigm results in solutions that may be easy to introduce but provide little assurance. The second is usually prohibitively expensive. This leaves the third, a gap-filler approach that allows for the installation of new components where the business case justifies, but makes as much use of existing infrastructure as possible. Many security products require months of "extend it" technical analysis and tweaking before they reach a stage where they can contribute to security productivity.

System Dependencies

Once all the intrusions are defined and mapped to appropriate response mechanisms, solving the intrusion detection problem does not require more than funneling the evidence to trigger procedures. Yet the ISO still has the challenge of verifying that all the evidence is in. There should be no lapse in integrity checking, no gaps in logs, etc. Hence, the ISO needs a monitoring system to ensure that the evidence required by the intrusion detection system is in place. One may envision an infinitely recursive monitoring structure, and so the ISO must be decisive in identifying the appropriate cap. The apex of any system monitoring hierarchy must be a system whose integrity, availability and performance is such that, if it went down, any resulting havoc wreaked by missed security events would be a trivial aftermath. The ISO must use all possible measures to ensure the security of that system. In a financial company, envision a system whose requirements are so robust that the company is not be able to produce financial statements or confirm end of day positions when it is down. This system is one to whom the security monitoring system may be trusted.

Security measurement is thus dependent on processes independent of security software to verify that the security requirements are met. These processes are built upon such non-security technologies as job scheduling systems, network management systems, trouble-tracking systems, and performance management systems. With the automated approach comes an assumption that those processes exist in the information systems environment and may be exploited to provide assurance the security products are working.

If it exists, a robust monitoring system can be relied upon to produce measurements that an ISO may use to determine the extent to which the intrusion detection system is yielding business benefits. Of all those alerts designed with explicit security concerns in mind, how many were resolved before being escalated to the status of security incident? Did these alerts reduce the cycle time normally required to respond to network or system outages? Of those that were security incidents, how many were due to system administrator mistakes versus actual intruders? Did any educational benefit result?

Of course, there is always the case where an intrusion is evident despite the fact that carefully planned data collection and correlation did not reveal it. However, that event is cause for a requirements engineering effort to get the right data. It is not cause to claim that the intrusion detection efforts in place are failures. It means they are beneficial to the point at which the supporting metrics stop. All intrusion detection systems, even the anomaly detection and learning algorithms ones, can only detect the types of intrusion for which they have been designed. The trick is designing a system that provides alerts only for real intrusions for which there is reliable evidence.

Conclusion

Intrusion detection has come of age, but it is unrecognizable to its progenitors. It is a practical application of data correlation techniques honed from a wide variety of information systems specialties. Network Managers, Database Administrators, Operating Systems Engineers, and Application Developers have all made contributions to the loosely coupled systems that compose today's intrusion detection systems. An ISO that is comfortable in turning all available technologies toward security monitoring goals will be successful at the intrusion detection game.

This paper has provided a theoretical framework to which today's successful intrusion detection implementations may be favorably compared. The approach starts with identifying intrusions that cause damage, that is, those that require response. The next step is to define how damage is identified and measured. Finally, the measurements taken must be informative enough to be used to improve security management in general as well as the intrusion detection process. The framework thus also provides a road map from this point forward in building ever more efficient and productive intrusion detection systems. ❑

References

[1] *Amoroso E,* Intrusion Detection, Intrusion.Net Books, Sparta, NJ, 1999.

[2] *Axelson S,* "The Base-Rate Fallacy and the Difficulty of Intrusion Detection," ACM Transactions on Information and System Security, 3(3):186-205.

[3] *Bayuk J,* "Security Metrics: How to Justify Security Dollars and What to Spend Them On," Computer

Security Journal, 17(1):1-12.

[4] *Carr J,* "Intrusion Detection Systems: Back to Front?," Network Magazine, 16(9):44-48.

[5] *Denning D,* "An Intrusion Detection Model," IEEE Transactions on Software Engineering, SE-13,(2)222:232.

[6] *Eliot L,* "AI Crime Busters," AI Expert (AI Insider Column), 7(1):11.

[7] *Goan T,* "A Cop on the Beat," Communications of the ACM, 42(7):46-52.

[8] *Jou Y,* Gong F, Sargor C, Wu X, Wu S, Chang H, Wang F, "Design and Implementation of a Scalable Intrusion Detection System for the Protection of Network Infrastructure", DARPA Information Survivability Conference and Exposition (DISCEX 2000), IEEE Computer Society Press, January, 2000.

[9] *Ko T,* Ruschitzka M, Levitt K, "Execution Monitoring of Security-critical Programs in Distributed Systems: A Specification-based Approach," Proceedings of the 1997 IEEE Symposium on Security and Privacy, pp. 134-144.

[10] *Mahadevan C,* "Intrusion, Attack, Penetration – Some Issues," Information Systems Control Journal, 6:52-57.

[11] *McHugh J,* "Intrusion and Intrusion Detection," International Journal of Information Security, 1(1):14-35.

[12] *McHugh J,* "Testing Intrusion Detection Systems," ACM Transactions on Information and System Security, 3(4):262-294.

[13] *Moyer P,* "Deepening Defense By Shoring Up Firewalls with IDS," Computer Security Journal, 17(3):13-18.

[14] *Meadows C,* "Three Paradigms in Computer Security," Proceedings of the workshop on New security paradigms workshop, ACM Press, January 1998, pp. 34-37.

[15] *Newman P and Parker D,* "A Summary of Computer Misuse Techniques," Proceedings of the 12th National Computer Security Conference, Baltimore, MD, October 1989, pp.396-407.

[16] *Pande P, Neuman R,* Cavanagh R, The Six Sigma Way: How GE, Motorola, and Other Top Companies are Improving Their Performance, McGraw Hill Professional Publishing, 2000.

[17] *Proctor, P,* The Practical Intrusion Detection Handbook, Prentice Hall, Upper Saddle River, NJ, 2001.

[18] *Schnackenberg D,* Djahandari K, Sterne D, "Infrastructure for Intrusion Detection and Response", DARPA Information Survivability Conference and Exposition (DISCEX 2000), IEEE Computer Society Press, January, 2000.

[19] *Feinstein B, Matthews G, White, J,* "The Intrusion Detection Exchange Protocol (IDXP)," currently documented at: http://www.ietf.org/internet-drafts/draft-ietf-idwg-beep-idxp-05.txt.

[20] *Webster's Third New International Dictionary Unabridged,* Merriam Webster Co., Springfield, MA, 1993.

[21] *Yokum B, Brown K, Van Deveer D,* "Intrusion Detection Products Grow Up," Network World, 10/08/01, available at http://www.nwfusion.com /reviews/2001/1008rev.html.

Jennifer L. Bayuk *is responsible for information security policy, architecture, and engineering at Bear, Stearns, & Co., Inc. She has been a manager of information systems audit, a Big 5 audit manager, and a software engineer. Reach her at jbayuk@bear.com.*