Global Information Assurance Certification Paper

Tracking the Inside Intruder:
Anomaly-Based Log Monitoring and Analysis

By
Peter D. Jekel

GSEC Practical Assignment
Version 1.3

Tracking the Inside Intruder: Anomaly-Based Log Monitoring and Analysis

Abstract

Losses inflicted by inside intrusions have the potential to be more severe to a network or organization than outside attacks, yet are often ignored or misunderstood. Unlike aggressive, real-time attacks from outside your network, inside intruders are known for their stealth and their slow, careful steps. While many networks are well prepared for the frontal assault pounding them from the Internet, their soft underbelly has only token protection from the insider. Compounding the exposure that comes from complacent internal auditing is the failure by network administrators to fully appreciate the differences between outside and inside intrusion. Often, this lack of knowledge results in the misapplication of tools and a false sense of security. With an understanding of anomaly-based strategies and a modest time investment, you can add an effective internal intrusion detection system (IDS) to your existing log analysis. A well thought-out system of analysis will actually shorten the amount of time you spend searching for internal intrusion and help expose serious holes that might otherwise escape notice.

Disclaimer: Although the illustrations used in this paper are based on real events, they have gone through a virtual Cusinart of obscurification. To protect the innocent, the guilty, and especially my prospects for future employment, all organizations and people are fictitious and events represent a composite of incidents gleaned over a period of years. With incidents set in fictitious American Widgets Companies, any resemblance to real organizations or people is not only coincidental but outright miraculous.

The Decline of Inside Intrusion - Myth or Menace.

Even as I raise the specter of internal abuse, respondents to the FBI/CSI (Computer Security Institute) yearly computer crime survey have reported a drop in inside intrusion over the past 3 years. The statistical decline fuels an industry debate on the meaning. The experts are divided on the subject. Is the decline real or have internal abusers simply become more capable and stealthy? Dan Verton, senior writer with computerworld.com, explores the question in his April 2002 article *"Insider threat to security may be harder to detect, experts say"*, and notes that opinions on the subject appear polarized. Information technology (IT) chiefs are convinced the majority of incidents and the greatest risks are from the outside. Security experts warn that the more serious risks and a larger number of incidents come from the inside. The convincing arguments on both sides leave you wondering whether the risk presented by an insider is an overblown menace or an unseen disaster swirling in undercurrents of your network? Experience tells me, failure to understand the risk, benign neglect, and lack of knowledge, blind many networks to the threat posed by inside abuse. Inability to recognize the threat will cripple the best surveys and leave a network's most valuable assets vulnerable.

Understanding the Risk: ***If you build a network - they will come.***

A false feeling of safety, and a diminished sense of risk from the inside, comes partly from a failure to grasp the severity of loss that can be inflicted by a privileged abuser. The news wires are full of high profile kidde-script exploits but strangely silent on losses from within. The few horror stories that leak out bring wide eyes but fail to overcome the it-can't-happen-here factor. A myopic fascination with the Internet cyber-thugs, blurs the sound thinking of many and throws the obvious disparity in potential loss between inside and outside intrusion out of focus.

Why are the losses more severe from the inside? It's the nature of the relationship. A trusted employee has special knowledge of the company's most important assets. Whereas a script-kidde can embarrass an organization and may through sheer malicious luck destroy something valuable, an insider knows exactly where to look and why it's valuable. It's this special, privileged, position of the insider that creates such unique and unexpected severity.

One example is the highly placed employee. In his article *"Don't ignore threat from within"* on Itworld.com, George Lawton relates an incident where a honeypot was set out to catch inside intruders. They were surprised to find the company's chief operations officer (COO) trying to break in. Loss and compromise at this level could be catastrophic for a company and may easily go undetected.

Another area ripe with inside severity is the disgruntled employee, especially one with an IT background. An insider like this can cause millions of dollars in damage, crippling a company for years. Consider the case of Timothy Lloyd, a vindictive network administrator, who was recently sentenced to 41 months in prison and ordered to pay $2 million in restitution. Angry at being fired, Lloyd planted a logic bomb on his former employer's network that destroyed all of the company's contracts and proprietary software, resulting in a loss of about $10 million.

Despite the high risk of the examples cited, their importance is lost in the roar of relatively minor incidents barraging the Internet, such as web defacements. With all the clatter, it's not surprising that top-level management perceives a diminished risk from inside intrusion. How do you get their attention and create awareness? The problem can be especially hard since organizations are justifiably cautious about revealing inside losses, leaving few incidents to cite. To present a convincing case to top-level management, you have to create awareness the old-fashioned-way, with incidents from your own logs.

Ironically, if your IDS is incapable of recognizing the threat from within, you may need to build one that is capable before you get the funding to do it right. One incident that stands out in my mind and illustrates this point was a job I took at a fair-sized Widgets Company to help set up some internal auditing. Though a large business with a well-

funded network, they had only two on their network security staff and a budget near zero. The security widgeteer who hired me had to beg for the funding and feared for results. On my first day, she looked at me, lips pursed, a nervous hand clutching a well-earned cup of coffee and said, "I sure hope we find something. I'm going to have to justify your expense to management." I assured her that if the industry reports were correct, her house was already on fire, she only had three firemen, no hose, no truck, no water, and the wind was rising. This did not seem to comfort her but she did get her wish. It took only a few weeks for her to catch a shipping clerk zipping up some of the company's most sensitive documents. Funding increased.

Benign Neglect.

Another reason why experts, vendors, and IT chiefs are at such odds on frequency of internal intrusion has to do with a darkly kept secret few want to discuss. No one is checking the logs. Worse, on many Microsoft Windows-based networks, the logs are not even enabled.

One of my first jobs in IT Widgetdom, right out of college, was babysitting a mission critical NT Windows database server. Curious activity, such as unexpected shutdowns and unexplained entries in the application and system logs, prompted me to check the security logs. Although the server had been in service approximately two years, no one had ever enabled the security logs. A check of the rest of the company's servers showed most were in a similar state.

How had this oversight happened? You may be aware that a default installation of any Microsoft product leaves the security logs disabled. The system and application logs are activated by default but the security logs must be manually set before they record anything at all. One can only assume this scheme was a conscious decision on Microsoft's part. Try as I might, I can think of no reasonable explanation to justify a security-blind deployment.

Considering the dominant role of Microsoft products, perhaps the single greatest change to thwart inside intrusion would be to get Redmond to enable security logs in their default deployment. Hope is on the horizon. Recently, Bill Gates announced a new Microsoft corporate push for security. I'm not sure if Microsoft is accepting email help but with their minds eagerly focused on security, now might be a good time to bring this oops-we-forgot-to-turn-your-security-logs-on feature to their attention. The lesson here is that good security starts with your own vigilance and you should never assume you are protected even with the most basic of security options. Regardless of what actions may come from Microsoft's latest endeavor, you are responsible for the safety of your network. Check your Windows servers, even if you think they're enabled, to make sure your logs are enabled.

While you're kick-starting your server logs, remember to enable logging at the desktop level as well. Many networks deal with logs only at the enterprise level and ignore the individual workstations. If you have Microsoft NT-type workstations, enable logging all the way to the workstation-level. As your response time to incidents improves with effective inside tracking, you'll find workstation-level logging extremely helpful in resolving anomalies on the network.

What about Unix syslogs? I have yet to find a Unix/Linux system with the security logging disabled but I have found the default configuration on many in need of strengthening. An excellent tutorial on the subject can be found in Simson Garfinkel and Gene Spafford's book, Practical Unix & Internet Security. The book's section on logging covers a number of basic issues such as remote logging and integrity tools that are valid for both Microsoft and Unix systems.

Lack of Knowledge.

Understanding the methods and means employed by an inside intruder is vital to waging warfare with the abusers. If you're tooled up to fight a desert battle but your enemy is hiding in the jungle, you have a real problem. The impetus for this article comes from experience where I was asked to look into a case of possible internal intrusion at a mid-sized American Widgets Co. (AWC). Several months earlier on a different project I had discovered a trojan on the AWC network and brought it to the attention of the IT security chief. Taking the compromise seriously, AWC set up vigilant log monitoring.

The company had an efficient, well-designed network with a strong firewall bolting the front door and state-of-the-art log monitoring tools. After a number of curious events with no visible problems in the logs, they called me back to look at their set up. The person monitoring the logs showed me their swiss-army-knife monitoring tool, clicking off an impressive real-time display of the network's sizable activity. Alarms and thresholds were set as designed for signature-based intrusion. Ironically, even as we were reviewing the logs, a quiet but assertive inside attack was taking place. The problem wasn't the IT widgeteer monitoring the logs. He was well trained on the monitoring tool and competent. The problem was a lack of knowledge resulting in a misapplication of the monitoring tool. Using a signature-based IDS with a real-time monitor to detect inside intrusion is like trying to churn butter with an outboard motor. It's very responsive but ill suited to skimming the subtle layers of a network for the anomalous ripples spawned by an insider. (The difference between signature and anomaly-based analysis will be discussed under strategies.)

Understanding the nature of inside attacks is fundamental to tracking the insider. Like birds of prey swooping on a network, hard-hitting, swift, external attacks require instant violation detection, rapid analysis, and fast response times. Internal intrusion is more like walking in the tall grass with lions. Spotting their slow, stealthy approach requires listening for anomalies, historical analysis of their tracks, and a fast response time. With

techniques and tools designed to snare hawks and eagles, one could easily be led to the conclusion that the lion problem is exaggerated. Strategy is the key to both recognizing the threat and configuring your tools to meet the need.

Strategies for Tracking the Insider

A good start, in grasping the difference in strategy between tracking inside and outside intrusion, begins with contrasting some of the key concepts.

- Top-Down Overview vs. Bottom-Up Details
- Baselining Network Activity vs. Signature Databases
- Anomaly-Based Analysis vs. Responding to Violations
- Designing Reports for Resolution and Contrast vs. Intricate Detail

Employing any one of these strategies by themselves will yield some fruit, but it is their interrelation that creates a powerful tool. Whether you're dealing with Unix flavored syslogs or Windows event logs, you'll find the concepts valid for detecting inside abuse. After briefly examining the thought behind each of the key concepts, I'll end the paper with a practical example that combines the different strategies into a tool with some proof-of-concept Java code.

Top-Down Inspection

Top-down overview is the most basic of the ideas. The concept is simple. Look at the big picture, not the little one. As easy as it may seem, most of us gravitate unconsciously to the concept of looking at logs from the bottom-up, one event at a time. Indeed, many of our intrusion tools reduce data to individual events or a collection of events for a particular entity. The tools, when used as designed, actually force us into a bottom-up paradigm. Consider the Microsoft Windows Event Viewer. It's an "event" viewer. This, of course, is a tried and proven method for detecting outside intrusion. You identify a specific event or series of events in the logs, gather the necessary information, and take up the scent.

A bottom-up paradigm asks the question, *"How does the event by this person or entity relate to this server or computer?"* Using this type of approach to track the insider is an overwhelming task and probably why most security administrators would rather have a root canal than analyze the logs. In a typical Widget Corporation, domain controller logs may generate 60 MB a day, and firewall logs add another 300 MB to the collection. How will you ever find the handful of the significant events pointing to inside abuse in a massive data collection like that? With a top-down strategy, it is possible to inspect 60

MB of logs in 5 to 10 minutes and accurately spot the ripples of intrusion coursing through your network.

You may be tempted to think that dumping your logs into a database will solve the problem. Most log monitoring programs are linked to a database, allowing you to sort by an individual user, workstation, site, or a particular kind of event. Isn't that a top-down inspection? Like a helicopter hovering over the data mountain, you have a far-reaching view. At first, database dumping looks like top-down analysis. It has the potential to be used that way. The problem is, we take a good tool for top-down analysis and we apply our bottom-up strategies. Looking at a collection of individual records or events from a great height is still bottom-up analysis.

Top-down analysis asks the question, *"How does this person's or entity's total network usage relate to the rest of the network?"* The top-down paradigm strives to create an image of an entire network and a network entity's entire use of that network, over an interval of time. It is your ability to decipher this picture accurately and quickly that gives you the edge in spotting inside intrusion.

To build this top-down picture, you need to categorize the basic building blocks of your network's topology. The categories should be common to the entire network, such as:

- Workstation names (W210A112233)
- Computer usage (Workstation or W)
- User names (John D. Smith)
- Logon Ids (smithjd)
- Job classifications (Accountant)
- Site locations (210A)
- Company specific divisions (NE, NW)

Think of these basic categories as colors on the palette you will use to eventually paint your image.

As you may have deduced, this is where good network planning pays off. A well laid out network is easy to categorize. For instance, a coherent naming convention will allow you to easily distinguish between different sites, workstations, servers, and perhaps type of usage. On the other hand, if your network is awash in a sea of machines named catfish and barracuda, the brilliant colors of your palette may be reduced to a murky shade of gray. Take heart, poor topology is not fatal to the cause but it does increase the complexity.

Top-down inspection gives you the means of using topological components to baseline your network activity. The view from the top allows you to use the baseline to spot anomalies. Individual network entities, such as users or workstations associated with the anomalies, can then be evaluated. With sufficient resolution, contrast, and focus, the image created by the top-down components can quickly help you determine whether the potential exists for abuse.

If your network is crippled with poor topology or you're struggling with the top-down paradigm, consider Priscilla Oppenheimer's work, Top-Down Network Design. The book's section on addressing and naming gives a clear, detailed description of how to establish sound naming conventions for network segments and devices. This top-down approach lends itself to identifying topological components for quick anomaly-based analysis.

Baselining Network Activity

The term baselining has some ambiguity as you search the topic on the Internet and reference materials. One common definition interprets baselining as setting a minimum standard on your network, such as every desktop will have antivirus protection. Microsoft uses the term baseline for its vulnerability scanner, MBSA (Microsoft Baseline Security Analyzer). For our purposes, baselining is the process where you learn what is normal or typical for your network. The concept is not new. Network engineers have used the process from the beginning to gain a statistical picture of a network's data flow. Applying the same type of monitoring to network security and log analysis is a natural extension.

Without sounding too mystical, this is the place where you become one with your network. To baseline, you sift your log data through your top-down components to produce statistics and patterns of network activity. Over an interval of time, these patterns create an image of your network that you can contrast with current network traffic. Baselining asks the question, *"What is typical for this network?"* How many users normally log onto the network on Wednesday? How many users usually log onto 4 or more workstations? What type of workers cross corporate division boundaries and how often? Who uses TCP port 22 for encrypted connections and what is the source and destination of these connections? Once you've laid the foundation with baselining, these are all questions you'll be able to answer in a matter of minutes. Really!

Ah! But as you might have guessed there is a catch. Nothing is free. During the first 30 days of baselining, you'll encounter a steep, but rewarding, learning curve. Although the startup period is intensive, it's not difficult. Instead of minutes, you may be spending an hour or more tracking down strange traffic. Admittedly, this is a lot in the busy day of a security administrator. No doubt, you'll be asking a lot of questions of the other IT departments and you may be tempted to ask yourself, is this really worth it? You betcha! Baselining will deliver some of the highest value monitoring you will ever do.

Why? Because an unmonitored network has the same wonderful growth potential as those tennis shoes you ignored all year in your sweltering high school gym locker. Benign backdoors, rogue programs rife with exploits, and sheer stupidity have a way of multiplying like a fungus. If you haven't been looking from the top-down, the fungus has had plenty of time to spread. One example that comes to mind was a network where the

top-down logs showed curious multiple connections to a workstation in a sensitive area of human resources at the company headquarters. An inspection of the destination computer showed a critical and confidential directory shared to the world. When we inquired about the situation with the manager responsible for maintaining the information, we were told, "It's OK, the files are read-only." The well-intentioned form-cruncher had a great idea for improving the restrictive work process. Rather than going through the cumbersome 4 or 5 step process that had been set up by IT, he'd let the field staff grab the information directly from his computer. To protect the data, he'd make sure only they could read it. He knew how to share a folder but lacked a fundamental understanding of file permissions, opening the company to enormous risk. Every shipping clerk, sales rep, receptionist, and temporary worker, company-wide, had only to peruse the Windows Network Neighborhood to walk away with the goods. Closing a hole like that is what makes you feel good at the end of the day, and why the company hired you.

A word of caution, as you establish your baseline, react with careful deliberation to unresolved activity. If you've never looked at the logs in this way, you will find some surprises. The risk of false positives is high. Don't turn on the siren and pull out the cuffs until you've established an historical base that allows you to recognize, with confidence, unusual behavior. You may not be aware that a program written in-house has a configuration error that generates spurious connection attempts by some users to a privileged customer database. After your 30-day break-in period, the amount of time you spend on false positives is trivial. There is no substitute for time and experience when it comes developing an accurate baseline.

Finally, baselining is important because there is just no way to get around it if you intend to effectively monitor your network. Whether you decide to out-source your security monitoring or embark on the bold quest to gain the experience yourself, you will need to baseline your network. Generic or default parameters used by an IDS or vendor may miss vital details peculiar to your network. You know your network better than anyone else. Once you've determined your baseline, you can turn your monitoring over to an outside vendor with the confidence the parameters you consider crucial are being watched.

Anomaly-Based Analysis (AB Analysis)

Perhaps the greatest departure from monitoring your logs the old-fashioned-way is relying on anomalies to recognize significant incidents. In the past we've looked at violations of established security policy to trigger an alarm. An unauthorized person connects to a device and it's logged as a violation. Anomaly-based (AB) analysis looks at a deviation from the baseline to identify significant incidents. It asks the question, *"What's wrong with this picture?"* An anomaly could pass all of your security requirements and still ring an alarm.

When researching AB intrusion detection, you will find a number of articles contrasting it with signature-based detection. The difference is worth noting. Most antivirus programs use signatures, which are known properties of a virus, to detect its presence. The majority of intrusion detection systems also use this method of detection. Signature detection lends itself well to real-time monitoring and known exploits. However, it offers limited value to the stealthy, within-the-rules, intrusion often used by insiders. An insider may use his legitimate permissions to directly place a remote access trojan on a company computer, never violating the firewall rules or passing through the IDS.

An article by Recourse Technologies, titled *"Intrusion Detection: Reducing Network Security Risk"*, divides AB analysis into two categories, behavioral anomalies and protocol anomalies. Behavioral anomalies are deduced from statistical patterns. Protocol anomalies are a more common type of intrusion detection, look for a violation of the protocol's RFC (Request For Comment) guideline or the protocol's typical use on the network.

Protocol anomaly analysis is often implemented with signature-based detection techniques. This type of detail involves rapid and intricate analysis at the application layer and is beyond the scope of someone manually inspecting the logs but well within the range of an IDS. AB protocol analysis still has a place in daily log inspections and can be done effectively at a macro level. Protocols such as FTP, HTTP, TELNET, and SSH, prone to abuse, can be baselined for normal usage patterns. Baselining can also be done for rogue programs such as Kazaa or Morpheus, the popular peer-to-peer file sharing programs, which use port 1214. Deviations can be easily spotted and investigated. If you've never done this type of analysis, you might think the amount of traffic created on these ports is so voluminous it would take hours to inspect. Experience will show you this isn't the case.

Kazaa and Morpheus are good examples of how clear the abuse shows up with protocol anomaly analysis. If you're inclined to minimize the importance of finding and stopping potential copyright abuse, consider the infringement settlement by Integrated Information Systems (IIS). The recording industry accused IIS of allowing employees to use company resources to download mp3 music files, forcing the technology consulting firm to settle for $1 million. Peer-to-peer file sharing programs may also allow thousands of unknown users across the Internet the ability to download valuable, proprietary company files. There is no excuse for ignoring the problem since the abuse is easy to spot with protocol anomalies. Even on a robust network with 5,000 to 10,000 users, you may only see 150 connections using port 1214. Connection patterns will show a varied number of users and usually involve foreign ports such as port 80 (HTTP), port 53 (DNS), or port 25 (SMTP). The data transferred by these connections is typical for the protocol, 100K - 500K at most. On the other hand, Kazaa will show a large number of connections to a single internal user on port 1214. The foreign port is usually above the restricted 1024 range. The foreign IP addresses will show a multitude of cable and DSL modems with data transfers running into the megabytes. Seeing the daily pattern in your report jump

from 150 connections to 3000 connections and then noting that the excess involves a single user, takes only seconds. The same methodology can be applied to SSH (port 22) or FTP (ports 20 and 21) and most other protocols.

Another method of high level protocol analysis is to sift HTTP traffic in your firewall logs for terms or IP addresses associated with abuse. Even sifting for the word "hack" or "loph" (as in lOphtcrack), produces only a score of responses that can be quickly scanned for significance. A shipping clerk downloading a password cracking tool is an anomaly worth investigating.

Detecting behavioral anomalies can yield even greater results in exposing the skilled inside abuser. A shipping clerk noted logging onto three workstations instead of the single workstation she normally uses, creates a behavioral anomaly. The data image created by your report allows you to quickly see that two of the workstations cross corporate divisional boundaries, another behavioral anomaly. The report also shows one workstation is located in a sensitive area used only by application developers, a third behavioral anomaly. Who has time to look at the myriad of workstations sprinkled throughout a network? With a behavioral anomaly report, it can be done in a matter of minutes.

Anomaly analysis is essential to spotting internal intrusion. It's the scanning-engine that captures the chaos of data streaming through your network and turns it into an image with recognizable patterns. With the right scripts and AB strategy, 60 MB of domain logs can be reduced to 15 minutes of high-value analysis.

Designing Reports for Resolution and Contrast

Applying strategies to track the insider takes more than simply possessing the tools. What makes AB analysis more difficult to implement than other methods, is its empirical nature on your network. How do you use the top-down components to create the report you need? Understanding the concepts of resolution and contrast in producing an AB image will help define the structure of your report.

In an AB image of your network, data collected over a period of time is responsible for the amount of detail or resolution. Think of it as a photo image, the greater the resolution, the easier it is to identify structures, people, or places. Data plays the part of pixels in an AB analysis of your logs. Spotting the stealthy movements of an inside abuser requires analysis with high resolution. A day's worth of data is more helpful than real-time monitoring. A week's worth of data will expose the movements of a highly skilled internal intruder with greater ease than 24 hours worth of data. To grasp the scale and scope of a major incident, you may need the resolution given by several months of data.

With this concept in mind, you can see why real-time monitoring offers limited value to tracking the insider. Looking at your logs in real-time with signature-based tools is like inspecting a high-resolution picture one pixel at a time. Any hope of finding stealthy internal intrusion would fall to dumb luck. Consider the case of an inside attacker using the QAZ trojan. If you employed real-time monitoring as your first line of defense, you might go months before spotting internal intrusion - that is assuming you have the logs turned on. However, the QAZ trojan stands out like a forest fire in an AB report with resolution as low as 15 minutes worth of data.

Contrast deals with the sharpness of a network image and equates to the type, number, and arrangement of top-down components in your report. Using our analogy of a photo, you could think of AB contrast as pixel depth with the type, number, and arrangement of components representing the bit-strength of each pixel. A detailed report with a single top-down component is like looking at an 8-bit grayscale image, while a report with 5 components, arranged to gather its significance at a glance, is more like a 16 bit full color image. Good contrast increases the ease and speed it takes to identify anomalies. A report that displays a list of people categorized by the number of workstations they logged onto is helpful. If I see someone logging onto 150 computers, it's going to stand out as an anomaly. Seeing a list of the workstations along with person's job description is more helpful and adds contrast. With this report, if I see a non-IT person logging onto 15 workstations, it stands out as an anomaly. A description of the workstation's location and divisional boundary is even more helpful. Now, if I see a person logging onto 3 workstations, but two of them are obviously outside the person's normal work area, it will stand out as an anomaly.

With the proper type and arrangement of components in your report, it's easy to see how a large amount of data can be sifted quickly for anomalies and their significance. Do you have to make tradeoffs in what you look at, allowing an intruder to escape notice? Yes. There are tradeoffs in whatever you do. However, as you become more familiar with your network and as the resolution of your data increases, it will become very hard for even the most skilled intruder not to stumble and attract your attention.

Proof-Of-Concept Example.

Touting the ease of tracking the inside intruder and the strategy it requires demands a practical example. In this exercise, I'm using a console-based Java program that uses top-down components to parse the Windows security logs from a fictitious American Widgets Corporation. The AnalogUser code was developed for a proof-of-concept project in 2001. Although written in Java, I believe the user will find the parsing concepts can be ported to SQL or any other in-house language used on your network. The code and example files can be downloaded in both zip and tar.gz format from the following URL:
http://home.attbi.com/~code-example/ABLE/Anomaly_Parsing_Exercise.html

Caveat: There are admittedly many improvements that can be made in the code's datatypes, algorithms, and report structure. The code is submitted as proof-of-concept to illustrate tracking an inside abuser from the top-down.

You may rightly ask me why I decided to use Windows security logs for this exercise. A Unix-type platform has better native tools and more complete logging. One reason is the ubiquitous presence of Windows in the market place. Tracking the inside intruder requires one to deal with Windows security logs. Ignoring them or dealing with them improperly leaves your network at risk. My experience is that a large number of network administrators, including security administrators, deal exclusively with a Windows environment and lack a working expertise with Unix or Linux.

Another reason for using Windows security logs is the challenge of it. In its native form, few logs lend themselves more poorly to top-down AB analysis than Windows security logs. The user interface is designed with a bottom-up paradigm to view individual records on individual machines. The log format just wasn't designed for the type of analysis needed to track an intruder network-wide. However, with proper parsing, the Windows security logs from multiple machines can be combined to offer excellent resolution and high contrast for behavioral anomalies.

How does this exercise relate to Unix-type syslogs? The concepts can be applied in the same manner to syslogs with similar results. Rather than re-invent the wheel, use a search engine with key words such as "syslog" and "scripts" to find a large selection of existing scripts and tutorials free for the download. As always, buyer beware.

Example Background: The American Widgets Co. (AMW) has about 10,000 employees working in three shifts and a WAN running coast to coast. They use both TCP/IP and Netbios protocol over the WAN. Topology for American Widgets is well laid out with its computers easily identified by its branch office and serial number. The network employs 15 domain controllers that collect 45 to 60 MB of log data per day. To keep the exercise simple, assume safe logging practices, such as remote logging, are being followed. For analysis, our AMW security chief downloads the logs once a day from each domain controller and combines them into one file with a simple script.

Tools Used In The Exercise.

Dumpevt: The logs are downloaded from the domain controllers with a tool called dumpevt by SomarSoft Utilities. Dumpevt is a console-based program that allows you to work directly from the command line or use a script to grab a copy of the Windows event logs off an NT-type OS. You can download a copy of dumpevt at URL: http://www.systemtools.com/somarsoft/somarsoft_main.htm

Even though the download page calls the program a trial, the readme documentation states that the tool is free. Microsoft makes a similar tool called dumpel, but I've found

the command line options and parsing by dumpevt a little more intuitive, and it takes less code to parse the results in Java. The exercise Java code only works with SomarSoft's dumpevt. Examples of a dumpevt script can be found on the web site where the exercise code is posted.

Java: The only other tool necessary for the exercise is the installation of Sun Microsystems' Java language on the computer using the Java code. To run the exercise code you must be able to run Java from the command line. Even if you have Java installed on your computer, you have to make sure that the binary file named "java", is located in the environment path. Fully qualifying the path to the binary will also work. The code was written with the Standard JDK 1.3 (Java Development Kit 1.3). The latest Java Platform is available as a free download from Sun Microsystems at URL: http://java.sun.com/j2se/ .

Analyzing the Logs for Inside Intrusion.

The task given to the American Widgets security administrator was to find out if the logs showed signs of internal intrusion. Although they'd never had a problem with inside abuse, it occurred to them that they had never looked for it. With a mounting tide of articles warning of its prevalence, they decide it would be prudent to put the issue to rest.

Rather than looking at every single user, the security administrator reasons that the basic unit of observation for this task would be parsing the domain controller logs for anyone who has logged onto two or more computers. Despite the potential blind spot, she argues that at some point even highly skilled intruders will have to expose themselves to multiple domain logons. It is possible for skilled inside intruders to leave no trail or disguise their tracks behind a single machine, but this possibility is slight. The decision to limit the report to users of two or more computers is a tradeoff but worth the benefit of reducing the task to a manageable size.

From the domain security logs the AMW security administrator identifies the following top-down components to use in the analysis:

- $ User's Domain Logon Name
- $ Originating Computer
- $ Date and Time
- $ Logon Success or Failure

To increase the number of top-down components in her analysis, she compliments the log information by linking the user's logon domain name with the domain's authorized user database. She adds the following components:

- $ The user's full name.
- $ The user's job title.
- $ The user's location.

She also incorporates the domain's resource database and uses it to add components that will:

- $ Identify the originating computer's location.
- $ Identify the computer's corporate division.

With the usual mandate to fulfill this project without funding, she taps in-house resources and convinces an enthusiastic and easily manipulated programmer to write some simple code to parse the logs for this information.

For baselining, our AMW security administrator has the program log basic statistics, such as the total number of users logging onto the network during the reporting period and the total number users accessing 2 or more computers. She also breaks the number of users accessing 2 or more computers into groups defined by the number of computers they accessed. The initial header containing the statistics looks like this:

```
Source Log File: example.log
Destination Report File:  <date-time>.txt
Run Date/Time: <date>  <time>
---------------------------------------------
Workstation and User Statistics

Total number of users logging onto the domain  = 4125
Workstation Group List: 67,14,6,4,3,2,1

Number of Users Accessing 67 Workstations = 1
Number of Users Accessing 14 Workstations = 2
Number of Users Accessing 6 Workstations = 10
Number of Users Accessing 4 Workstations = 12
Number of Users Accessing 3 Workstations = 20
Number of Users Accessing 2 Workstations = 370

Total Users Accessing 2 or More Workstations = 415
```

After 30 days of baselining, the AMW administrator is surprised to find that the percentage of users accessing multiple workstations is only about 10 percent. This is a manageable number. The baseline also shows percentages within the different groups stay roughly constant. This format allows her to quickly see if any gross anomalies exist on the network. For instance, if she sees a group accessing 100 computers, well above the baseline list, it may be a good indication of a worm. An abnormally high number of users accessing 2 or more computers may indicate a more extensive compromise of the network by a worm such as Nimda or QAZ.

To enhance the baseline statistics, the report also gives a quick overview of network usage by listing the domain logon names for the users in the workstation groups. A typical workstation group file looks like this:

```
-----------------------------------------
Users Accessing 4 Workstations:

millerme        kimlt
-----------------------------------------
Users Accessing 3 Workstations:

johnsonra       bakerjn         goodim          hernandezjg
nguyenas
-----------------------------------------
```

Baseline experience tells our security administrator that the number of users accessing multiple workstations falls off dramatically above the 4 workstation category. Almost all the users in the higher categories are IT staff. After 30 days of baselining, she easily recognizes the names. User lists for the workstation access groups allow her to key in on gross behavioral anomalies. A non-IT person, accessing 10 computers, is quickly recognized as an anomaly.

Analyzing the components for subtle behavioral anomalies is accomplished with a summary of each user in the workstation access groups. Those accessing the most computers are detailed first. The AMW security chief formats the summary to accent key information, allowing her to scan the record like an image. In this way, she can recognize an anomaly without having to scrutinize particulars. Once an anomaly is identified, she can increase the resolution and contrast with little effort to grasp the significance.

The report is generated in an ASCII text format. The "~" symbol at the beginning of each record allows her to dump the report in a word processor and use the symbol in the search function to quickly advance through the records.

The sample user summary looks like this:

```
~-------------------------------------------------------
goodim          (Ina M. Good | Shipping Clerk 2 - NC GLPC 511N )

Workstations Accessed = 3

Workstation     Region    Location

\\W511N164152    NC      Great Lakes Production Center WI
\\W024A011322    NE      Widget Corporate HQ NY
\\W831C160181    SW      SW Regional Office CA

Activity Summary

\\W511N164152      | 8/12/2003  | 09:47:42  | S024ADC1| Successful Logon:^   | Logon ID:(0x0 0x8571930)    |
\\W024A011322      | 8/12/2003  | 10:47:43  | S024ADC1| Successful Logon:^   | Logon ID: (0x0 0x862EDC1) |
\\W024A011322      | 8/12/2003  | 14:54:09  | S024ADC1| Successful Logon:^   | Logon ID:(0x0 0x89D0D8A)  |
\\W024A011322      | 8/12/2003  | 15:03:06  | S024ADC1| Successful Logon:^   | Logon ID:(0x0 0x8A2AC24)   |
\\W831C160181      | 8/12/2003  | 07:59:22  | S511NDC1| Successful Logon:^   | Logon ID:(0x0 0x1C61A24)   |
\\W831C160181      | 8/12/2003  | 09:09:34  | S511NDC1| Successful Logon:^   | Logon ID:(0x0 0x1CB32D5)   |
```

In the summary example, behavioral anomalies show up with just a glance. Keying in on divisional boundaries and computer location, our administrator can see that the shipping clerk went outside her normal area. In addition to logging onto her own computer, goodim accessed a computer in New York at the company headquarters and a computer at the SW Regional Office. Like a snapshot, all of this information is gathered in an instant. With practice, scanning each record for anomalies takes between 1 and 2 seconds. An entire day's worth of records takes 400 to 800 seconds or about 15 minutes. Of course, that's just the beginning of the work. I only claimed you could identify the incidents in a short period of time. Investigating them is where the fun and time really add up.

Summary.

Internal Intrusion Basics:

- $ Recognize the risk: If you build a network they will come. You already have internal abusers on your network. The potential loss severity they pose to your network requires specific action to address the risk.

- $ Turn on the logs. If Bill (as in Gates) hasn't turned on your Windows security event logs, turn them on.

- $ Use the right strategy. Don't churn butter with an outboard motor. Real-time, signature-based monitoring works well for external threats but has limited value for tracking down inside abuse. Design a top-down system to spot log anomalies.

- $ Baseline and monitor: Become one with your network and watch for stealthy lions. Take time to baseline your network and monitor anomalies with vigilance.

Detecting, and responding to, inside abuse is essential to safeguarding your network. Whether you have an existing SQL log monitoring program, a state-of-the-art IDS, or a console screen and raw logs, you can effectively identify inside intrusion with top-down anomaly-based analysis. Many of us shy away from tackling log analysis, believing that the job is overwhelming. This perception is in part fostered by our legacy methods of inspecting the logs from the bottom-up, one record at a time. As the exercise demonstrates, adding anomaly-based techniques to your existing system may be easier than you think, allowing you to rapidly analyze results. The illumination it offers is crucial when tracking the stealthy movements of internal intrusion.

References.

Computer Security Institute. "CSI/FBI Computer Crime and Security Survey". 2002. 7 April 2002.
URL: http://www.gocsi.com/press/20020407.html

Garfinkel, Simson. Spafford, Gene. Practical Unix & Internet Security. Sebastopol: O'Reilly & Associates, Inc., 1996. 289 - 324.

Gaudin, Sharon. "Net saboteur faces 41 months". 4 March 2002.
URL: http://www.nwfusion.com/news/2002/0304lloyd.html

Gaudin, Sharon. "Case Study of Insider Sabotage: The Tim Lloyd/Omega Case". Computer Security Journal, Volume XVI, Number 3, 2000.
URL: http://www.gocsi.com/pdfs/insider.pdf

Integrated Information Systems Inc., Press Release. 12 April 2002.
"Integrated Information Systems Responds to 2001 Settlement with RIAA".
URL: http://www.iisweb.com/press/press releases 2002/RIAA press release 041202 Final1.pdf

Jekel, Peter. "Anomaly-Based Log Parsing Exercise". 14 May 2002.
URL: http://home.attbi.com/~code-example/ABLE/Anomaly_Parsing_Exercise.html

Lawton, George. Security experts say: "Don't ignore threat from within". 14 June 2000.
URL: http://www.itworld.com/Sec/2052/ITW1124

Lemos, Robert. Kane, Margaret. "Gates: Security is top priority". 17 January 2002.
URL: http://news.com.com/2100-1001-816880.html?legacy=cnet

Oppenheimer, Priscilla. Top-Down Network Design. Indianapolis: Macmillian Technical Publishing, 1999. 157 - 191.

Recourse Technologies. "Intrusion Detection: Reducing Network Security Risk — continued". Executive Perspectives. 24 December 2001.
URL: http://www.isp-planet.com/perspectives/ids_p3.html

RFC Editor Homepage. "The RFC Editor." 7 March 2002.
URL: http://www.rfc-editor.org

SomarSoft Utilities. Dumpevt Download Site.
URL: http://www.systemtools.com/somarsoft/somarsoft_main.htm

Sun Microsystems. Java 2 Platform, Standard Edition. March 2002.
URL: http://java.sun.com/j2se

Verton, Dan. "Insider threat to security may be harder to detect, experts say". 12 April 2002 URL: http://www.computerworld.com/storyba/0,4125,NAV47_STO70112,00.html