



**AN INTRODUCTION TO NETWORK AND HOST BASED
INTRUSION DETECTION**

Prepared By:

CHRIS PETERSEN

CTO

Introduction

Intrusion detection is many things. My definition of intrusion detection is tools and techniques used individually or in combination to detect network and system misuse. Notice the absence of the word “intrusion” in my definition; instead I use the word “misuse”. I do this because intrusion detection technologies detect the unauthorized action of the intruder as he “misuses” systems and networks. I’ll get more into this later, first I’d like to draw an analogy between intrusion detection and something you’re more familiar with – protecting your home.

If you’re like me, you live in a 5000 square foot house, have a garage filled with exotic cars, rooms alive with art, and a safe storing untold fortunes in cash and bearer bonds. OK, maybe that isn’t my reality and it probably isn’t yours. However, if this were your home, how would you protect it?

In the physical world you would probably try to prevent the intruder from getting into your home by installing locks on the doors and windows. In the event the locks are picked or the intruder is an insider (that new butler), you’d probably install motion detectors and cameras in key areas to detect and record any suspicious activity. Finally, to ensure your family would be safe in the event an intruder did gain access, you’d develop a response plan ensuring the police are at your door prior to any irreparable damage.

These actions correlate to the fundamental computer security controls of prevention, detection, and correction. Whether developing a protection plan for your home or your network, these are the three areas that must be addressed to ensure the plan is complete. Are locks on doors and windows going to defeat an experienced cat burglar? – not likely. And what good are motion

detectors and alarms if nobody is ready and able to respond. The same is true when protecting your network.

Preventive controls are the measures taken to *avoid* misuse *from* occurring. In the physical world they are the locks put on doors, the barricades erected in front of entrances, the fence around your property. In the digital world they are the controls ensuring only authorized users have access to your network and systems. They are your secure ID cards, the access controls on your file server, your firewall.

Detective controls are the measures taken to *detect* misuse *when* it occurs. In the physical world they are cameras in the corners, pressure pads in hallways, and motion detectors inside the fence line. In the digital world they are controls ensuring that if misuse does occur, it is detected. They are the audit logs of failed login attempts, the system logs of failed file access, and of course, intrusion detection systems.

Corrective Controls are the measures taken to *respond* to misuse *after* it occurs. In the physical world it is the alarm systems and the call to 911 that brings the police or appropriate response unit. In the digital world they are the controls ensuring that when misuse does occur, the organization is ready and capable of responding. Unlike preventive and detective controls, corrective controls are less technology and more process. They are the policy and procedure for responding to an attack against your web server or for dealing with an employee selling intellectual property to a competitor.

Combined, these controls create a complete security architecture whether protecting your home or business critical network and systems. Unfortunately, too many organizations focus 90% of their resources on preventive controls and 10% on detective and corrective. While this mix may make sense

for some organizations, for most it doesn't. The security and availability of your network is too important to rely only on preventive controls. These controls can be evaded by experienced attackers and are often poorly implemented.

Intrusion detection can provide balance to the equation, enabling organizations to detect misuse when it occurs and assist them in responding accordingly. So what is intrusion detection?

A Brief History

Prior to the mid 90's, intrusion detection was the process of reviewing audit and system logs. The audit log providing the who, what, and when of system access – who has authenticated, what did they access, when did it occur. The system log providing status and error messages from the operating system and applications. Combined, these sources of information provide enough information to detect approximately 90% of attacks – after the fact. However, ten years ago enabling the audit log required too much highly valued CPU cycles and was rarely turned on. Even if was, rarely was it reviewed – there just were not enough hours in the day to manually review thousands if not tens of thousands of audit records. As for system logs, good system administrators make it a practice of reviewing them on a regular basis. Unfortunately, most are not trained in how to find an attack amongst all the other activity and again, the review can occur hours or days following an attack. This model simply couldn't scale as client-server computing took root and more companies began connecting themselves to the Internet – automation was required.

Enter the first generation of intrusion detection systems, systems that could process thousands of log messages and identify those messages indicative of misuse. The government and associated research

institutions developed the majority of these early systems. They eventually proved the concept of host-based intrusion detection and the race to commercialization was on. In the mid 90's the first commercial products began to be introduced as well as a new detection approach referred to as network-based intrusion detection.

The introduction of network-based intrusion detection systems (NIDS) addressed two key issues with host-based intrusion detection systems. First, in order to analyze audit and system logs, the logs had to be forwarded to a dedicated processing system or a program had to be installed locally. Secondly, the analysis was limited to an individual system. This meant only those systems forwarding logs or having local software installed were protected. In large environments this impacted network bandwidth when forwarding logs or impacted performance when the program was run locally. Network-based intrusion detection sought to solve this by taking a different approach.

Instead of analyzing the results of misuse when evidenced in the audit and system log, NIDS looks for signs of misuse as it travels across the network. There are three principle benefits to this approach; first, misuse can be detected as it occurs instead of after. Second, misuse can be detected against any system on the network. Lastly, the system is passive; there is no requirement to install software on each system or forward logs. The benefits of NIDS made intrusion detection more cost effective and as a result commercial adoption began to occur.

As adoption began to rise, so did the demand for better systems. The capabilities of today's modern intrusion detection systems have advanced significantly beyond the point of only 5 years ago. In fact, today's intrusion detection systems are becoming tomorrow's intrusion prevention systems – but that is for another whitepaper. For now we will dig

further into the details of how host and network-based intrusion detection actually detects misuse.

General Approaches in Detecting Misuse

Although many terms are thrown out regarding how a product detects misuse, I believe they can all be described as either signature or anomaly based.

Signature-based detection, also referred to as pattern-based, looks for evidence known to be indicative of misuse. Whether it's looking for specific log entries or a specific payload in a data packet, the NIDS/HIDS is looking for something it knows about – a signature of misuse. These signatures are developed over time by research teams either in the public (e.g., Snort) or employed by commercial IDS vendors. For example, ISS has their x-force, NFR has their Rapid Response Team, Enterasys, Cisco, and others all have their respective R&D groups as well.

Key Advantage(s):

- Typically signature-based approaches result in fewer false alarms because they can be very specific about what it is they are looking for.
- Because the IDS is looking for something known, a lot of information regarding what the misuse is, the potential impact, and how to respond can be provided. This knowledge is extremely important in understanding what is occurring and effectively responding.

Key Disadvantage(s):

- Signature-based approaches can only detect misuse for which a signature exists. For a signature to exist, the form of misuse must be known about beforehand so it can be researched and programmatically identified. This means any new form of misuse will not be detected by a signature based system

until it is identified, analyzed, and then incorporated into the product. Depending on the circumstances, this could be hours, days, weeks, or even months.

Anomaly-based detection looks for signs that something is out of the ordinary that could indicate some form of misuse. Anomaly-based systems analyze current activity against a “baseline” of “normal” activity and look for deviations outside that which is considered normal. For example, an anomaly-based system might build a baseline of how and when computers communicate across the network. All future communications will then be compared against the “normal” baseline of communications to determine if anomalous activity is occurring. For example, the system might detect two hosts communicating with each other for the first time, or two systems communicating with each other at an abnormal time (e.g., 3:00 AM). Another example of an anomaly-based system would be one that models user activity on a local system – what time they typically log-in, what programs they use, the files they access, how fast they type, etc. A baseline of this type of activity can be built and then compared against future activity to detect changes in behavior.

Key Advantage(s):

- Because anomaly-based systems are capable of detecting misuse based on network and system behavior, the type of misuse does not need to be previously known. This allows for the detection of misuse a signature based system may not detect.

Key Disadvantage(s):

- Because behavior on a system or a network can vary widely, anomaly-based systems have the tendency to report a lot of false alarms. The art of effectively identifying “normal” activity vs. truly abnormal is extremely challenging.

However, some recent technologies have begun to manage the “noise” factor of anomaly-based detection and commercial adoption is beginning to rise.

- Unlike signature-based detection, no knowledge of the misuse exists beforehand. This means very little information exists to assist the user in understanding what the misuse is and how to respond. This requires a much more sophisticated user to analyze and understand the output.

Although almost all methods employed by intrusion detection systems can be classified as either signature or anomaly based, as you will see, the specific techniques vary widely. However, in general the pro’s and con’s of each approach tend to hold true.

Host-Based Intrusion Detection

As previously introduced, intrusion detection technologies are most often described as either network-based or host-based. Network-based intrusion detection systems detect misuse at the network level while host-based intrusion detection systems detect misuse at the system level. For example, a NIDS would detect an attack against your web-server as the attack data travels across the network while a HIDS would detect the same attack once it reached the web server and was processed. While NIDS has advantages, they also have weaknesses, weaknesses that are the strengths of host-based intrusion detection.

Modern host-based intrusion detection systems are agent based. The HIDS software is installed on the machine to be monitored becoming a highly integrated component much like anti-virus software. Because HIDS run on the machine itself, the level of analysis compared to NIDS is much deeper. They have the advantage of being able to see the results of action vs. the attempted action. For example, a NIDS might detect someone

attempting to access unauthorized information. However, it is most likely blind to whether the access actually occurred. HIDS can not only see the attempt, but also the result. This provides more actionable response as a failed attempt probably requires no immediate action while a successful attempt does. HIDS provide more detailed information as to the result of misuse than does NIDS – a very important strength.

The manner in which HIDS detect misuse has also changed over the years. Current products typically employ a number of different detection techniques, each of which are described in detail below:

Log Analysis

The need to more efficiently analyze system and audit logs was the catalyst that eventually bore commercial intrusion detection systems. Today, most commercial HIDS still employ some form of log analysis. Log analysis is a signature based approach in that the system is looking for specific log entries having known relevance in regards to misuse. Log analysis systems “hook into” audit and system logs analyzing every log entry recorded. Each log entry is compared against a list of pre-defined signatures. The signature is a pattern of data that uniquely identifies a specific log entry. When a log entry matches a signature, the log entry is typically processed further and then reported to the user or backend reporting system. The quantity and quality of signatures determine the effectiveness of log analysis systems. The quantity of signatures determines how many unique log entries the system can identify. The quality determines how effective each signature is in identifying the pertinent log entry. These two factors combined determine the overall effectiveness of the system. For example, a system with 10,000 signatures of low quality is likely less useful

than a system having 1000 high quality signatures.

File Integrity Monitoring

File integrity monitoring consists of monitoring sensitive files for inappropriate access. The basic types of file access are read, write, and delete. With file integrity monitoring, sensitive files can be identified for monitoring and notification issued whenever someone reads, writes (modifies), or deletes them. For example, on a file server containing confidential data, file integrity monitoring could be used to record whenever a file is read to ensure only authorized individuals are accessing the information. Another example would be to use file integrity monitoring on an external web server to notify the administrator whenever the content of the web site changes - extremely useful in detecting an attacker defacing your Internet presence. Tripwire first introduced this type of detection and it is now seen in a number of other systems.

System Call Analysis

System call analysis monitors interactions between application programs and the core of the operating system – the kernel. The kernel has a base set of programs that perform all low-level functions such as allocating memory or reading from a file. These low-level operating system functions are accessed by higher-level application programs through system calls. System calls provide the interface by which applications interact with the operating system to perform all basic functions.

Because system calls provide the only interface between application programs and the operating system, use of system calls can be monitored and analyzed to detect misuse. This is accomplished by inserting a program between system calls and the kernel. This program, often referred to as a “*shim*”, can then intercept and analyze all interactions

between an application and the operating system. This level of integration with the operating system provides an unequalled level of inspection capability as every request to the kernel can be analyzed.

Using system call analysis, two approaches are commonly used to detect misuse. The first approach is signature based and examines every application request against a list of known attacks. The second approach is anomaly based and examines every application request in the context of what normal kernel activity is.

System call analysis can be a very powerful form of intrusion detection given the level of integration with the monitored system. However, this integration comes with a cost. First is the potential performance impact of passing every system call through the HIDS before being processed by the kernel. Second is the fact that HIDS using this technique becomes an integrated component of the kernel and therefore, if the HIDS fails or has bugs, could impact the operation of the monitored system. However, this type of monitoring has been used in anti-virus and personal firewalls for years and the current commercial HIDS employing this technique have been doing so for some time now. Just make sure to perform appropriate testing prior to deploying this type of solution.

Network-based Intrusion Detection

Network-based intrusion detection systems come in the form of software or a fully integrated appliance. Whether delivered as a black box appliance or as software to be installed, almost all technologies share a common set of characteristics and use one or more general detection methods.

Common Characteristics of NIDS

Unlike host-based intrusion detection that monitors a single system for misuse, a single network intrusion detection system can monitor multiple systems – a key advantage of network compared to host-based intrusion detection systems. NIDS inspect activity as it travels across the network in the form of packets. There are many types of inspection performed but the basic architecture is the same regardless of detection technique or product.

NIDS are typically configured with more than one network interface card (NIC) – the hardware connecting your computer to the physical network (except in the case of wireless of course). One NIC serves as the “monitoring” interface, and the second NIC serves as the “management” interface. The monitoring interface collects network traffic and passes it to the NIDS program for analysis. This is accomplished by placing the monitoring NIC in “promiscuous mode”. Promiscuous mode is a special setting that configures the NIC to capture everything it sees on the physical wire to which it is connected. In a normal operating mode, a NIC only collects those packets destined for its IP address. Setting the interface to promiscuous mode allows for the capture of all network traffic visible to the NIC and therefore, the network intrusion detection system. This provides the capability for misuse detection across the network vs. a single system. Another characteristic of the monitoring interface is configuring it to be invisible to other computers on the network. This is done by not binding the interface to any layer 3 protocol (e.g., IP). In other words, the interface has no IP address and cannot be addressed by other systems or transmit any packets – this is commonly referred to as a “stealth” interface.

The management interface is configured more like a typical interface. It is assigned an

IP address and can be addressed like any other system. The function of the management interface is to provide remote access to the NIDS for administration and to provide event reporting from the NIDS to a central reporting system. In all cases, the NIDS itself needs to be secured much like a firewall to ensure it cannot be comprised. To add an additional layer of security, an advisable architecture is to establish an isolated management network that handles all NIDS administration and reporting traffic. This has many benefits besides being more secure but can add cost to the overall solution. The following diagram depicts a high-level NIDS deployment.

Although the basic architecture may be the same or similar across product, the manner in which today’s commercial products detect misuse vary greatly.

Signature-based detection

The most common technique used by NIDS to detect misuse is pattern matching, looking for a specific pattern of activity referred to as a signature. For NIDS, these patterns are found in the data of network packets. For instance, when an attacker targets a web server, the attack itself travels across the Internet in one or more IP packets visible to the NIDS. A signature-based NIDS examines each packet against a database of signatures where each signature is a specific string or code snippet that identifies the attack. This technique is extremely effective in identifying attacks having a very unique signature. Unfortunately, it also means attacks not having a truly unique signature are either not detected or detected too often leading to false negatives or false positives.

➤ **False negatives**

A false negative is when an attack occurs but the NIDS doesn’t detect it. Lack of an alert gives a false sense that no misuse has occurred. To step back to our physical world analogy, it’s when the

motion detector fails to detect the intruder because he has figured out how to evade it, disable it, or it's simply defective. There are many possible reasons for a NIDS false negative including evasion techniques and misconfiguration of the system. However, in the context of pattern matching, false negatives are when a signature for the attack hasn't been developed. This is most often due to the attack having never been "seen in the wild". In other words, the attack hasn't been seen by enough smart people to identify it is an attack and then develop a signature for it. When an attack is first identified, it is often referred to as a "zero-day exploit" meaning the counter has started on the number of days the attack is known. When zero-day exploits become known, the open community and vendor R&D groups race to develop a signature for the attack. Up until the signature is developed and the NIDS is updated, it is susceptible to false negatives.

➤ **False positives**

False positives are the other end of the spectrum, they report misuse when in fact, no misuse has occurred. In the context of signature-based detection, false positives occur when the signature matches what is normal activity. This can be the result of two scenarios, the signature is not specific enough in what it looks for or the signature is very specific but normal and malicious traffic match the same signature. False positives have been the chief criticism of intrusion detection systems in the past and continue to be a significant issue in the overall usability and cost of the system. However, signature-based systems have had the best results to-date in having the lowest false positive to true positive (actual misuse) ratio.

Perhaps the most well-known and pervasively used signature-based NIDS is Snort (www.snort.org). Snort is an open source NIDS that rivals many commercial products in terms of detection capability. There is an active community of open-source developers constantly improving snort. Recently some commercial companies have begun to take Snort commercial much like Red Hat has done for Linux.

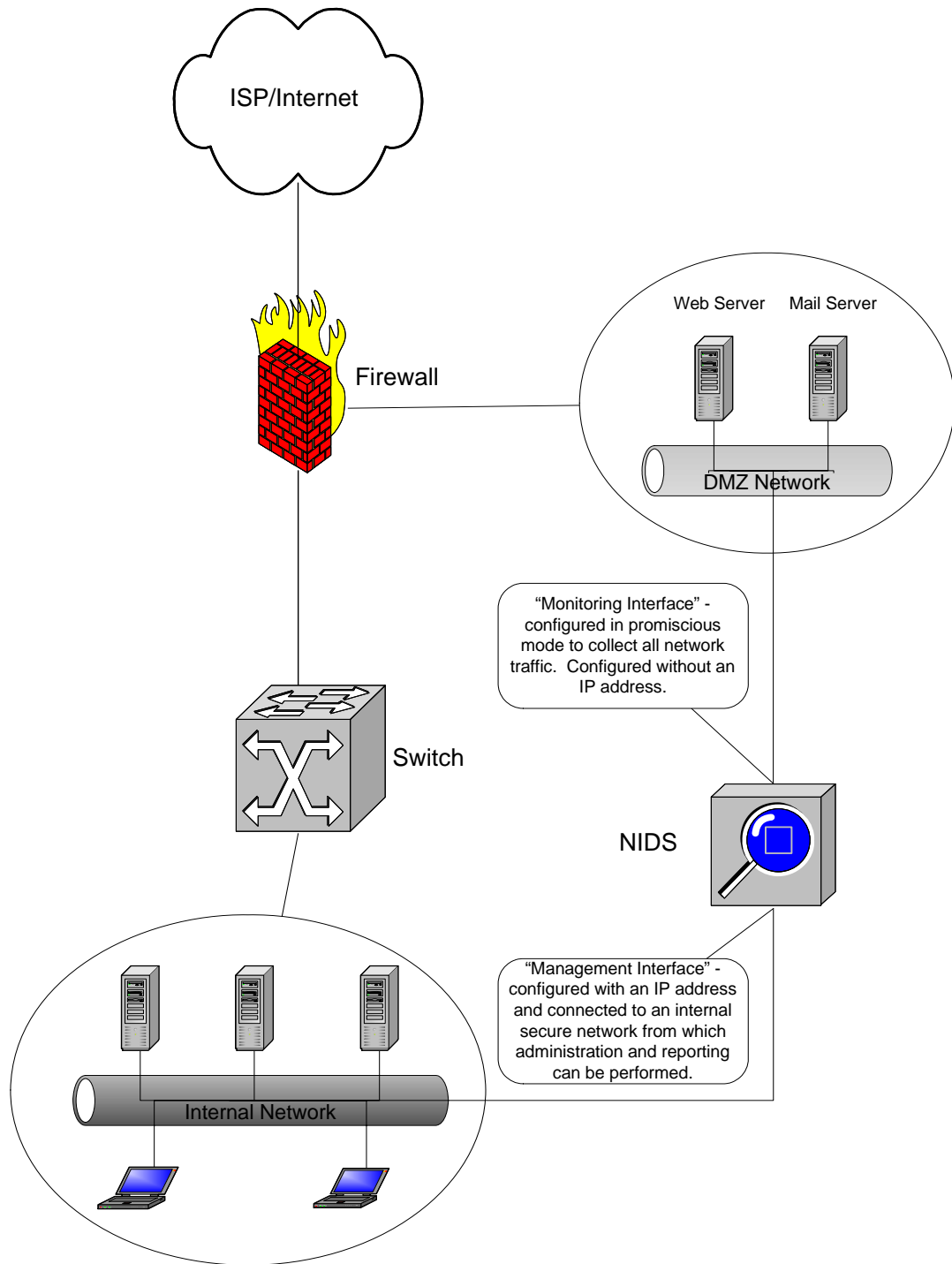
Protocol-based detection

Protocol-based detection, also referred to as specification or RFC based detection, looks at the problem a bit differently. Instead of analyzing the contents of each packet, the manner in which the packet is formed is analyzed. This manner is dictated by a protocol - IP, TCP, UDP, HTTP, SMTP, etc, etc, etc. Sounds complicated? It is. Perhaps an analogy to the real world will help.

When you drive from home to work every day you may not think about it, but you follow a protocol. The protocol determines what you can legally do from point a to point b. You must go on green, stop on red, and keep your speed below a specified level. This is a protocol, a set of rules that allows for the seamless interaction of cars over streets just as IP does for computers over a network. A protocol, whether it be the rules of the road or the specification of a network packet sets rules that ensure the system works. However, protocols can be ignored, you can stop on green, run a red, and speed to work. There may be consequences but the protocol itself can do nothing to prevent these actions. The same is true of network protocols, following them ensures everything works, not following them results in a breakdown of the system. Protocol-based NIDS detect violations of network protocols just as radar guns detect speeders.

In every network communication a variety of protocols must be followed to ensure

Figure 1



everything works. These protocols are often defined in Request for Comments (RFCs) that are developed by standards groups and then implemented by vendors in their software. The RFCs provide the design, vendors translate that design into software. Protocol-based detection analyzes each packet against the rules an RFC specifies on the assumption a violation of the RFC is indicative of misuse. Unfortunately, when an RFC is implemented there are many cases where it is not specific and/or allows for interpretation on the part of the vendor. This combined with the fact that some vendors don't comply 100% with the RFC results in many products having slight nuances in how they implement the protocol. Similar perhaps to state motor vehicle laws as compared to federal guidelines – some states follow, others don't. This introduces challenges in effectively performing protocol analysis.

To account for this and make protocol analysis more effective, signature-based techniques are combined with protocol analysis to look for specific protocol violations known to be indicative of misuse. For many protocols, there are specific actions an attacker may take to bend the protocol to his favor. Protocol analyzers can look for these specific protocol violations as well as general RFC violations of the protocol.

Anomaly-based detection

Anomaly based detection is probably the least defined detection method due its open-ended nature of detecting “anything not normal”. Given it is hard to define normal, it is equally as hard to define abnormal. However, the use of anomaly-based detection techniques have seen a rise in recent years as vendors have looked to augment signature-based and protocol-based techniques with an additional detection layer. The key advantage of anomaly-based

detection over the other two techniques is the ability to detect new attacks, or rather attacks for which no signature or known protocol violation exists. This is the panacea of anomaly-based detection – one the academic and research institutions pursued in the past and now is being seen in commercial form.

The most prevalent “real-world” use of anomaly-based detection is in analyzing network traffic. Patterns of network use can be identified, for example, who is talking to who, when, and how? What are typical traffic loads against key servers, typical packet sizes, typical download sizes, etc. Basically, any type of network activity that can be modeled over time can be used as a baseline with which to identify anomalous network traffic. The baseline is activity regarded as normal against which all future activity is compared. This is perhaps the biggest challenge with anomaly-based detection, developing a baseline to compare against that resembles your “normal” network state and is free itself of abnormal activity. To draw an analogy, this is like trying to determine the normal traffic flows in your city within a given time period when you know that during that period, people are speeding, running red lights, etc. How do you build a normal baseline of automobile traffic when that traffic always contains abnormalities – a challenge indeed.

This is not to say anomaly-based solutions have no merit. As hardware capabilities have increased, the ability to design more complex algorithms and store more historical data has allowed anomaly-based designers the ability to overcome limitations they faced in the past. Many NIDS today deploy anomaly-based techniques to provide another layer of detection and correlate that activity with the results of other detection techniques. However, anomaly-based detection is a buzz word thrown around a lot

so when a vendor makes a claim to be anomaly-based, be sure to get a clear explanation of exactly how they build their baseline and the types of anomalies they are looking for.

Conclusion

Hopefully you now understand intrusion detection better than you did before. It is a complicated and fascinating technology pressing the cyber detective against the cyber criminal. It's the same game played in any inner city only the players in the digital world manipulate bytes and have genius level IQs. You may not find it quite as intriguing as I, but hopefully you have a better appreciation for what intrusion detection is and how important a role it plays in defending your digital domain.

About the Author

Chris Petersen's unique combination of industry experience and technology vision has resulted in the industry's most comprehensive and relevant enterprise log management and analysis solution. His diverse set of experiences has lead to a rich understanding of customer problems and needs. As a Senior Consultant with Price Waterhouse (now PriceWaterhouseCoopers), he provided information assurance services to Fortune 500 clients and developed the Price Waterhouse Enterprise Security Architecture System. At Ernst & Young, Chris led an engineering group in developing one of the first managed security services and eSecurityOnline.com, a leading information assurance portal. Chris was among the first twenty employees at Counterpane Internet Security, where he made significant contributions to the technical and business aspects of Counterpane's pioneering managed security monitoring service. Prior to founding LogRhythm, Chris was responsible for product marketing at Enterasys Networks, helping to drive the Dragon Intrusion Detection System to a

market leading position. Chris has spoken at numerous conferences, been quoted in numerous publications and is a faculty member with the Institute for Applied Network Security. Chris has a degree in accounting/information systems from Colorado State University.

About LogRhythm, Inc.

LogRhythm was founded by Chris Petersen and Phillip Villella to address the unmet and growing need for a comprehensive log management and analysis solution.

LogRhythm's comprehensive log management and analysis software helps companies to efficiently comply with government regulations, secure their networks and optimize their IT infrastructure.

LogRhythm's log management system automates the collection, identification, centralization, archival and recovery of all log data. Its log analysis capabilities automate centralized analysis, correlation, and reporting while also providing real-time monitoring and alerting on critical issues. By automating the collection, organization, analysis and archival of all log data, LogRhythm enables enterprises to easily comply with log data retention regulations while simultaneously gaining valuable, timely and actionable insights into security, availability, performance and audit issues within their infrastructure.

LogRhythm is cross-platform, highly reliable and easily scalable across an enterprise. Unlike inadequate and inefficient home-grown scripts, LogRhythm provides a complete, highly efficient and easy to use solution for companies of all sizes that need to solve log collection, management and analysis challenges. With LogRhythm, companies can invest in a single, integrated solution that addresses the needs of all departments, whether their concern is security, compliance, audit or IT operational efficiency.