Global Information Assurance Certification Paper

Intrusion Prevention: Does it Measure up to the Hype?

By

Brian C. Rudzonis

SANS GSEC Practical v1.4b

Abstract

One of the newest topics given much press in the technology trade magazines has been the so-called “intrusion prevention” products. The pages that follow will describe the technology called intrusion prevention and give a clear understanding of how it relates to the general field of intrusion detection. Products representing niches within the intrusion prevention category will be presented to reinforce the basic understanding of the technology. Next, the discussion will focus on the potential benefits and drawbacks of using intrusion prevention tools. The paper will then conclude with an analysis of the technology and suggest possible situations for its use so readers may decide whether or not it measures up to the rave reviews that can be found in almost every technology magazine in recent months.

Just by hearing the term “intrusion prevention” one cannot help but be lured into learning about how it can solve the usual problems with hackers and malicious code. One does not need to look far to learn more. Almost every information technology trade magazine is littered with articles singing the praises of intrusion prevention. *Information Security Magazine* even made intrusion prevention the cover story for October 2002. These articles even have interviews with security professionals who have recently discovered these intrusion prevention tools and cannot figure out how they were able to perform their jobs without them. So the natural conclusion is the technology must be wonderful. But is it true?

First of all, intrusion prevention is a sub-category of the larger category of intrusion detection tools. Intrusion detection can be described as “the art of detecting inappropriate, incorrect, or anomalous activity.” (http://www.sans.org/newlook/resources/IDFAQ/what_is_ID.htm). The key to intrusion detection is the word “detect”. Intrusion prevention, on the other hand, seeks to prevent actions that can be described as inappropriate, incorrect, or anomalous.

The intrusion prevention line of products actually dates back about three years. The first company to use the phrase was called Click-Net. Their intrusion prevention product was called Entercept. Today, the product and the company are both called Entercept. Their unique approach to security was to evaluate system calls against a database of attacks patterns and decide if the system call should be permitted or denied (Briney, p. 1). The actual blocking of suspicious system calls thereby prevented the malicious action. A classic intrusion detection system would have detected the attack (presuming it matched a signature within its signature database) and reported it to the proper authority. The attack would have been successful in this case. One can see just how powerful the intrusion prevention technology could become. Thus, it led to many other companies marketing their own line of intrusion prevention technology.

Intrusion prevention products have been developed to protect a variety of resources within a network. There are host-based products, like Entercept. Trusted operating systems can also be considered intrusion prevention products. There are also web server shields and even web application firewalls. Gateway intrusion detection and prevention products round out the classes of products.

Trusted operating systems provide greater security over the entire operating system. Important concepts are enforced such as least-privileged. Even resources such as network interfaces are given extra security protection. The overall picture is locked down environment where administrative access is controlled tightly and auditing capability is extended. The classification of trusted operating systems as intrusion prevention tools comes in because all the resources of the operating system, such as processes, files, and administrative tools are isolated from each other and prevent many malicious actions by their very design. Mainstream operating systems rely upon knowledgeable and security conscious administrators to tighten security. Trusted operating systems are tightened out of the box and are designed to be tightened to a degree not even possible by a standard Unix or Microsoft Windows installation. Sun

Microsystems markets their Trusted Solaris as an option for trusted operating systems. According to Sun's description of Trusted Solaris 8, it implements a multi-level file system, segregates users into internal and external groups, and strictly enforces control over what users can see and do (http://www.sun.com). With a feature-rich security tool set as contained within Trusted Solaris, it is clearly seen that many malicious actions can be prevented which are normally exploited on a standard Unix platform. However, the drawbacks of using this type of operating system are the higher cost, the increased complexity of the operating system, and the tight security controls are sure to increase the difficulty of a mass deployment. The highest returns from a trusted operating system would be on the most mission critical systems or the ones that contain an organization's most valuable assets.

Web server shields are another sub-class of the intrusion prevention category of security tools. By their very name, it is not hard to figure out they function to protect web servers. They usually integrate with the web server and monitor all the requests to that server. The goal is to prevent many of the common web server exploits plaguing the web community, like buffer overflows and the execution of arbitrary code through URL requests. An example of a web server shield is eEye Digital Security's SecureIIS™ Application Firewall (http://www.eeye.com). This particular web shield wraps around Microsoft's Internet Information Server (IIS). It actively blocks anomalous requests from being executed on the web server, which effectively prevents the attack from succeeding. This again is intrusion prevention at work. Another key point to the intrusion prevention tool classification is the ability to detect attacks that are yet to be known. This can occur because the tool prevents the attack from succeeding based on the actual function calls rather than a signature of the traffic. This concept will be described in more detail in later pages when discussing the strengths of such tools.

Web application firewalls are another class of intrusion prevention products. These firewalls are more commonly found on the perimeter of the network and monitor http traffic and users' web sessions. According to Pete Lindstrom in an October 2002 *Information Security Magazine* article, web application firewalls protect against attacks such as cookie poisoning; URL masking, encoding, or manipulation; and protecting against database attacks through the application on the web server (Lindstrom, pp. 1-3). Although eEye Digital Security's SecureIIS refers to itself as an application firewall, for the purposes of this paper it is considered a web server shield since it integrates with the web server. An example of a web application firewall is Sanctum's AppShield. The key to their technology is being able to prevent attacks against custom-written web applications. An application written by a developer for use within an organization may contain any number of vulnerabilities. While these vulnerabilities are not widely known throughout the Internet community, they still pose a potential threat due to the possible use of poor coding techniques, a lack of testing, or poor overall design. Sanctum has remedied this with their AppShield product (http://www.sanctuminc.com). Again, it is important to note their technology can work on well-known attacks or attacks which are yet to be

developed. The product does not require updating signatures or tweaking rules. This is yet another advantage which will be discussed later.

Another example of an intrusion prevention category of products is actually an evolutionary step within the network intrusion detection realm. This category is a network intrusion prevention system. Traditional intrusion detection monitors traffic and generates alerts based on a match of the traffic pattern to a signature database. The key here, again, is the alert that is generated. No action is taken other than to bring the action to the attention of the security team or the administrators. Companies have taken their products a step further and added the capability to alter firewall rule sets in order prevent the malicious traffic from entering the internal network. This leads to the network intrusion prevention category of tools. In order to accomplish the task of dropping packets and network traffic, these tools often need to be put inline with traffic entering the network. Contrast this to traditional network intrusion detection tools, which passively listen to traffic or even sample traffic somewhere along the data path or at selectively chosen points in a network. Unlike the other types of intrusion prevention tools, many network intrusion prevention tools depend on, at least in part, the use of signatures to detect potential malicious traffic. This is because they are based on the original idea of network intrusion detection. Internet Security Systems' RealSecure™ Guard falls into the category of a network intrusion prevention tool (http://www.iss.net). It is placed directly on an external link and analyzes traffic for anomalous patterns. Upon detection of potential malicious traffic, Guard simply blocks the traffic.

Earlier, the concept of intrusion prevention was introduced with the host-based tools. Host-based intrusion prevention tools intercept system calls and decide whether or not to pass them onto the operating system kernel. Malicious calls are blocked and logged. The founding father of the intrusion prevention class of security tools, Entercept is the classic example of host-based intrusion prevention. Entercept provides a robust capability by using both the classic signatures and the use of behavioral rules to prevent damage to systems (http://www.entercept.com). Once again, there is the common theme of having the ability to prevent attacks which are yet unknown.

To this point, the discussion has focused on discovering the exact nature of intrusion prevention tools and reinforcing their basic functionality that leads to their name and separate classification. This is, of course, the ability to take active steps to block attempted attacks on a network, and the ability to prevent attacks that are as of yet unknown or untried. The discussion will now transition to point out the strengths and weaknesses of this technology. In making any business decision one needs to objectively evaluate benefits and drawbacks rather than be sold on the hype in the permeated throughout the press.

Like many new technologies, intrusion prevention is not without its advantages. It can actively block actions from taking place, thereby preventing hackers from successfully assaulting networks. As seen earlier, these tools can work on a system call level and thereby block attacks that have not even been discovered. As such, it can reduce the need for the continuous path cycle many organizations seem to be in as flaws in software are discovered on a daily basis.

Many intrusion prevention tools do not depend on only looking at traffic coming in through external interfaces to be successful, thus providing some protection from malicious insiders. By blocking malicious actions, these products can stop the rampant spread of many types of viruses and worms and even prevent a network from becoming a platform for attacks against other networks. When using the principle of defense in depth, these tools can point out weaknesses in perimeter defenses and classic intrusion detection systems. These systems are also less administratively intense, as signature-based systems require updates as new signatures are released, or the creation of homegrown signatures which is difficult.

Discussed earlier, intrusion prevention tools will actively take action to block attacks before they happen. This is an extremely important point and cannot be understated. Consider two situations. In the first, a network is protected by a traditional intrusion detection system that monitors any traffic successfully reaching the internal side of the firewall. Since, like many firewalls, there are ports opened up in order to do business with the outside world, malicious traffic does make it through on occasion. The occasion in this case happens to be the day after Thanksgiving, the well-known beginning of the Christmas shopping season. A hacker probes the network and finds a few holes and potential exploits. The intrusion detection system picks up the probes coming through the firewall and automatically notifies the lead administrator. However, the administrator called in sick and is out shopping. The hacker, meanwhile, is able to go through a list of possible exploits and do all kinds of unthinkable damage. Does it really matter if it is taking the web site down which processes orders, launching a successful denial of service attack to prevent customers from placing orders, or deleting vital data? The point is, the damage has been done, and even if the administrator was able to react in minutes, what could have been done? Taking down the servers or blocking access via the firewall would be a denial of service anyway. Now consider the second situation. Here we have an intrusion prevention tool. We can have a network intrusion prevention tool, a web server shield, a web application shield, or a host-based intrusion prevention tool. The same hacker finds the same potential vulnerabilities and launches attacks. Any number of intrusion prevention tools prevents these attacks from taking place AND alerts the administrator at the same time. No damage has been done and the company saves itself the embarrassment and potentially bankruptcy-saving financial damage. If the example seems over-simplified, just think of how prevention of one single successful attack can save much money and reputation. This is a definite huge advantage of intrusion prevention over intrusion detection and contributes to the rave reviews received by the press.

Another distinguishing factor of intrusion prevention tools is also one of its advantages. This is the ability to detect attacks that have yet to be attempted or have yet to be patched. Learned from earlier discussion, this ability comes from looking at system calls to the operating system, web server, or web application. Previous attacks have been analyzed to determine exactly what is being passed on to the system. Intrusion detection analyzes traffic patterns and depends on

matching these traffic patterns to signatures found in a database. However, many of these attacks have very different traffic signatures. At their core, they may result in similar calls to the operating system. If a new attack is designed for which there is no signature, intrusion detection will not even pick up on it. If the new attack is based on the same type of operating system call as other known attacks, the new attack will still be blocked. This is a very powerful advantage, as attackers are always varying their techniques to devise ways to evade intrusion detection systems. And, of course, the intrusion detection system will only detect and alert to the attack, where the intrusion prevention system will prevent it from occurring.

Another huge advantage of these tools is the avoidance of being caught up in constant patch deployment. Almost daily there are new vulnerabilities released regarding some type of application or operating system. If an organization runs a mixed environment of operating systems and applications, this constant patching can create a nightmare in staffing levels and in the potential instability of conducting business on an ever-changing baseline. Just how do intrusion prevention tools ease patch deployment? As described in the previous paragraph, these tools can detect and prevent attacks not yet known. If a system contains a critical server for which an exploit is found and it is not patched, it is vulnerable. The problem is this server may end up being patched several times a year. Multiply that by how many servers there are in an organization. Enter intrusion prevention. The new vulnerability may just be a new spin on an old problem. It results in the same system call being made to the operating system. It will prevent the attack from being successful even without the patch. The result is some organizations can reorganize their staff to concentrate on other projects and there will be more stability with the systems. Patching can be moved to a quarterly or biannual basis.

An advantage of host-based intrusion prevention, trusted operating systems, web server shields, and web application firewalls are the ability to protect systems from any malicious attack, whether they come from external entities or insiders. Some of the most damaging attacks come from insiders who are familiar with the infrastructure. Many organizations spend most of their time concentrating their efforts on external connections. Even much of the intrusion detection equipment is focusing on the external links. This leaves the internal networks vulnerable to the person who was given notice his/her job was being terminated or someone who did not get the big promotion. If nobody is looking or attempting to prevent a problem originating from the inside, the damage could be much worse than what a hacker could do. Intrusion prevention does not discriminate between internal and external system calls or traffic (unless it is the category of network intrusion prevention). The systems are shielded from everyone alike.

One of the basic security principles is to deploy a layered approach to protection. This is, of course, called defense in depth. Firewalls can protect external connections, but due to the need to communicate with the outside world, there are inevitably ports opened up through which exploits can occur. Firewalls also do not protect against every vector of attack. Electronic mail with malicious

code, users inadvertently introducing malicious code, modem connections, and users who are already inside the firewall cannot be protected by protection on external connections. Intrusion detection focused on monitoring external connections also will miss many of these potential problems. There are even circumstances where intrusion detection is monitoring key internal segments and the system is either blinded by a denial of service attack, the software is not configured correctly, or the signatures have not been updated. Intrusion prevention provides an additional layer to plug the holes left open by other mechanisms. They can even point out potential weaknesses with intrusion detection sensors if only the intrusion prevention software logged an attack. This can result in the strengthening of the other layers of protection if, for example, the host-based intrusion prevention tool is continuously detecting attacks that get by the firewall and intrusion detection system. Administrators can correct or update the software and the overall security posture of the system has been strengthened.

One final important advantage of intrusion prevention tools is they require less maintenance than their intrusion detection cousins. Because they are based on behavioral patterns instead of known traffic patterns, they do not have to be updated as frequently even if new attacks are discovered. As long as the attacks are based on known behavioral patterns, the software does not need the update and will detect and prevent the new attack.

With all the advantages of intrusion prevention tools, one could see how the press would create the hype through their respective technology periodicals. However, there are two sides to every story, and intrusion prevention is no different in this respect. Some of the advantages even become disadvantages. For example, the very action of blocking system calls can prevent systems from properly operating, especially if the systems are very dynamic in nature. Anything this good probably costs a lot of money, and many of these tools will impact even a large budget. Even though many as of yet unknown attacks will be blocked, this assumes the new attacks follow the same behavioral patterns as known attacks. Some of these devices are also meant to be deployed inline with traffic entering a system. This is a natural point to create a performance bottleneck in the system. Not necessarily a weakness, but when patches are deployed to fix certain issues, the question needs to be asked whether or not those particular actions need to be blocked in the future. As with any relatively new technology, the playing field is in continuous flux. The chosen vendor for a particular product may not be around in the future. Or the vendor's product may be integrated into another line of products. This could change the architecture of the system because the product to which it is integrated might be a competitor to the vendor of the current firewall or network intrusion detection system. In order to stay current, the firewall and network intrusion detection system would have to be replaced. This causes an increase in expenses and labor. The final weakness is the same across any technology – the human factor. Any system is only as good as the person designing and maintaining it.

Remember one of the defining actions of an intrusion prevention product – it actively blocks actions it determines to be malicious. What if the action was

actually a valid action by your application? It would cause the application to malfunction or fail. One might think applications should not be making system calls that are interpreted as malicious. However, not all programmers produce a clean product – hence the multitude of patches available for any given product. What if the intrusion prevention tool was configured to correct the problem of poor coding by ignoring the system call? Now the system is vulnerable to real exploits and defeats the purpose of having the intrusion prevention tool in the first place. Naturally, configuring the system to ignore a single type of system call will not render the product useless, but it simply makes it less effective.

Take the concept of blocking valid system calls a step further. This particular example takes a system that is very dynamic. Either there are many different platforms or applications, or the system is continuously being updated. New software or updated software introduces new system calls. Some of these new system calls may be interpreted by the intrusion prevention software as malicious. In a dynamic environment, one must take into account the possibility of the intrusion prevention software eventually registering false positives and impacting operations. A static environment, on the other hand, would achieve stability after a period of operation and have a lesser possibility of disruptions due to the intrusion prevention software.

Another downside to the intrusion prevention movement is the cost. Many of these tools cost a lot more than their intrusion detection cousins. These tools are more advanced and represent a more significant investment in research and development. Simply deploying host-based intrusion prevention to every server and desktop would probably not result in a very good return on the large amount of money that would have to be spent. It also represents another piece of the infrastructure to maintain, which leads to more labor expended to keep it functioning.

Although the possibility of preventing as of yet unknown attacks is enough to make anyone run out and purchase an intrusion prevention system, how can it possibly detect every single unknown attack? Not all attacks are based on the same tried and tested buffer overflows, for example. The basis for blocking attacks comes from the historical analysis of previous attacks. As intrusion detection analyzes previous traffic patterns, intrusion prevention analyzes the system calls that are at the core of the attacks. If there is a weakness to a particular portion of an operating system or application that has not yet been identified, the intrusion prevention tool may not be able to block it if the tool does not know the behavioral pattern behind the attack.

Some of these tools or devices can be potential bottlenecks in the performance of a system. Network intrusion prevention tools need to look at every system call coming into the system. If the intrusion prevention tool cannot keep up with the traffic, users will notice the delay. Some applications are also low-latency dependent and cannot tolerate a performance bottleneck. The addition of software on a system can also create an increased load on a processor, as the host-based intrusion prevention software is analyzing every call made to the operating system. This can also result in a decrease in performance.

Another potential issue arises when patches are installed to bring operating systems and applications up to date. If patches fix potential vulnerabilities, then does the intrusion prevention software need to evaluate the system calls if they do not result in a successful attack? One might think it does not create much of an impact. However, the event will still be logged and administrators and security personnel might be responding or investigating every time the alarm goes off for a blocked action that is patched anyway. It also takes more processing power to block the attack that will not succeed anyway. To many organizations it might represent an inconsequential circumstance, or it might not. It comes down to a decision based on each individual situation.

As with any new technology, there are often a myriad of startup companies offering great solutions. They depend on marketing a single product. Organizations purchasing a product from this type of company could be left unsupported if the company declares bankruptcy. Other possible situations might involve the company being taken over by a larger company or making a deal with a larger company which results in the product being integrated into another line of products. If an organization's other security solutions are from competing vendors, it could force the organization to go through a major change in their security structure. Lack of change would eventually result in the failure to support the intrusion prevention product. A lack of support means the software would become out of date and as time goes on, the number of attacks it prevents might decrease. This is not a unique issue to intrusion prevention tools, but it tends to be more prevalent in newer technologies.

Another issue not unique to intrusion prevention is a human factor. A system is only as strong as the person performing the configuration. A great tool can become useless if installed properly. And in the case of intrusion prevention, an improperly configured tool can become deadly to one's business. Remember that a denial of service could result from valid system calls becoming blocked. And performance bottlenecks could result from misconfigured tools as well.

Now the reader should be aware of the positives and negatives of using various types of intrusion prevention tools. Given these factors, there are obviously situations where intrusion prevention excels. The single most important factor to an organization is the return on investment (ROI) of a security tool. After all, why would an organization purchase and install a product if it did not benefit the business? Therefore, logic dictates it would be best to determine the circumstances for which an intrusion prevention tool would give the best ROI for an organization. Since all organizations are different, there is no single answer.

The proper analysis is to perform a risk assessment. Determine the assets that need to be protected. Are there web servers that are vital to the business? Is there proprietary information or code that cannot be compromised? Is the information vital to national defense? These are just some of the questions to identify the critical parts of the system. Now determine where the threats come from. Is the system connected to the Internet? Are there remote access servers? Does the organization depend on a lot of temporary help, which could result in many people gaining access to a system over a period of time? Are the

security policies loosely based (if they even exist), which results in users having much freedom on the system? Now match the risks with the threats and determine if there is a vulnerability that can make an attack successful. Also take into account past incidents and the cost to the organization. If attacks are costing a certain amount of money and a particular intrusion prevention solution can reduce this dollar amount by a certain percentage, the cost of the product and deployment might pay off. If the organization depends on the Internet for business, then it makes sense to protect the connection to the Internet and the web server. If the users have a lot of freedom on the system, then host-based security might make sense. If there is the potential of an insider threat, then it might take an approach with multiple tools. If an organization has been in a never-ending patch cycle, then host-based intrusion prevention can slow down the cycle to a more reasonable frequency. Every situation is unique and requires its own analysis. The goal is to get the most results out of each dollar spent and spend those dollars in the right place.

Now it should be clear that intrusion prevention clearly has a place in the crowded marketplace of various intrusion detection tools. It is very important for one to be familiar with the positives and negatives and the risks that are to be mitigated. Armed with these tools, one can make a very informed decision. Perform a careful risk analysis, come out with the key points to protect, calculate the total cost of deployment, and determine the cost savings. If the figure calculated meets your business goals, it is time to start planning. If not, then do not fall prey to the rave reviews and become one of the types of people who information technology professionals loathe – the decision maker who picks up a trade magazine, sees a fancy advertisement or product comparison, and begins the planning right away.

References

"AppShield: Web Application Firewall."
URL: http://www.sanctuminc.com/solutions/appshield/index.html. (November 1, 2002).

Bobbitt, Mike. "Inhospitable Hosts." Information Security Magazine. October 2002. URL: http://www.infosecuritymag.com/2002/oct/cover.shtml. (October 24, 2002).

Briere, D. & Bacco, C. "The Bleeding Edge: Intrusion Prevention Systems Complete Security." Network World Fusion. October 15, 2002. URL: http://www.nwfusion.com/edge/columnists/2002/1015bleed.html. (October 24, 2002).

Briney, Andy. "What Isn't Intrusion Prevention?" Information Security Magazine. April 2002. URL: http://www.infosecuritymag.com/2002/apr/note.shtml. (October 24, 2002).

Cummings, Joanne. "From Intrusion Detection to Intrusion Prevention." Network World Fusion. September 23, 2002.
URL: http://www.nwfusion.com/buzz/2002/intruder.html. (October 24, 2002).

Desmond, Paul. "IDS Tools Smarten Up." Network World Fusion. September 9, 2002. URL: http://www.nwfusion.com/research/2002/0909feat.html. (October 24, 2002).

"Entercept: Intrusion prevention for enterprise servers."
URL: http://www.entercept.com/products/entercept/index.asp. (November 1, 2002).

Lindstrom, Pete. "Guide to Intrusion Prevention." Information Security Magazine. October 2002.
URL: http://www.infosecuritymag.com/2002/oct/sidebar.shtml. (October 24, 2002).

"RealSecure® Guard." URL:
http://www.iss.net/products_services/enterprise_protection/rsnetwork/guard.php. (November 2, 2002).

Saita, Anne. "Paying for Protection." Information Security Magazine. October 2002. URL: http://www.infosecuritymag.com/2002/oct/casestudy.shtml. (October 24, 2002).

"SecureIIS™ Application Firewall: Proactive Web Server Security." URL://http://eeye.com/html/products/secureIIS/index.html. (October 28, 2002).

Smith, Tom. "Intrusion-Prevention Vendors Promise End-to-End Protection." Internet Week. September 9, 2002. URL: http://www.internetweek.com/story/INW20020909S0007. (November 1, 2002).

"Trusted Solaris™ 8 Operating Environment: Features and Benefits." URL: http://www.sun.com/software/solaris/trustedsolaris/features.html. (October 28, 2002).