VIGILAR™

Is Intrusion Prevention Changing Information Security?

Revision Version 1.1, March 30, 2004

Written by: Eric Ahlm, CISSP - Vigilar Inc.

The Intrusion prevention space is one of the industry's hottest topics right now. The market has been validated by growing demand, manufacturer success and independent studies. But what can intrusion prevention do to better your company's security, or lower your security administration burden? Is intrusion prevention simply warmed over intrusion detection with the ability to stop unwanted traffic, or is there more to the technology? This white paper will discuss the differences between traditional intrusion detection and intrusion prevention for both host and networks, and endeavor to answer the question has intrusion prevention changed information security.

Table of Contents

Intrusion Detection versus Intrusion Prevention

Intrusion detection has long been a part of a classic layered security model for valuable assets beyond the information security world. The security model states that one endeavors to prevent attacks to assets, protect assets from ongoing attacks, and monitor assets for signs of attack. A real world example is how a bank protects its money from the would-be attacker. A preventative measure is the legislation the bank puts in place to thwart would-be attackers from taking action. A protective measure is the very thick steel door on the safe, and the monitoring is the cameras put in place to watch the vault for attack. The whole concept of monitoring revolves around incident response. When the security guard watching the bank's video monitor sees a thief picking away at the safe lock, he knows to call the authorities to respond to the incident.

In information security, the same model would have security administrators patching and hardening systems to prevent attacks, protecting assets with firewalls, and monitoring against attackers with intrusion detection. Although the model is a solid plan, many have found its execution to be flawed. Network based intrusion detection systems are well known for having a high number of false positives. With any security system, false positives lead to either increased monitoring manpower, or more commonly, deactivation of the monitoring system. Here is one of my favorite anecdotes to illustrate that point. Say one has a car alarm which is a monitoring system of sorts to detect car theft. The attacker decides at 2A.M. in the morning to kick the tire on the car, set off the alarm and run off. The alarm goes off, and the system monitor wakes up to check the incident out. The monitor sees no attacker, so resets the system and goes back to bed. The attacker waits 15 minutes, and trips the alarm again. After a couple of round of this, the system monitor is going to assume the system is flawed, and deactivate the alarm. At this point, the car thief steals the car and drives off undetected.

Some have decided because of the high probability of false positives in network based intrusion detection to move the detection closer to the assets. In other words, move the detection to the host, instead of the network. Although the accuracy is up there is a need for drastically increased response time. Say the bank decides to move the security camera inside the bank vault. There will no longer be false positives, but by the time the security administrator detects the thief inside the vault, it's too late to call the authorities with any chance of protecting against the attack.

So what is different with intrusion prevention? For both host based and network based intrusion approaches, the defining difference is that the tool used has the ability to take an automated response to the detected attack and take action to stop it and report on it, rather then just report on it. The best analogy continuing on the bank comparison is that the monitor to detect attacks against the safe now has a stun-gun feature to disable attackers! How exactly this is done will be covered it the following sections. One might ask the question at this point, "If intrusion detection had false positives, will intrusion prevention now make the same bad decisions, but this time stop valid traffic?" This paper will discuss the new methods used by intrusion prevention systems to avoid that very problem.

Traditional Network Based Intrusion Detection

Commercial based Networked intrusion detection systems, or NIDS as it's sometimes called has been around since the mid 1990's. The infosec world was introduced to the WheelGroup's Netranger and Internet Security System's Realsecure. The business issue was simple, to detect attacks on a trusted network and be able to respond to them in enough time to prevent loss.

For each trusted network segment, a sensor would have to be placed. In an enterprise environment, this could mean 40, 50 or more sensors each needing to be monitored, updated, and maintained. Early on performance was an issue, since networks would reach speeds greater than the sensors could examine packets. This could lead to multiple sensors per segment.

The first generation of Commercial NIDS was purely a signature-based model. A NIDS sensor was placed on each network segment needing monitoring and "sniffed" each network packet to check against the database of "known" attack signatures. This method is still used to some part today. When a new threat is released in to the wild, security analysts will review the network signature the exploit uses, and create a signature to detect that exploit that is (hopefully) unique from any other type of network traffic. Below is a partial packet capture from the Welchia worm. How the Welchia worm used ICMP (Ping) to find its next victim was unique and could be detected with signatures.

Partial Packet Decode of Welchia Worm. (The long string of "a"s was a dead giveaway that Welchia was running ping sweeps on a network)

```
11:47:47.576542 169.254.56.166 > 169.254.189.84: icmp: echo request
0x0000 4500 005c 599d 0000 8001 970c a9fe 38a6 E..\Y.........8.
0x0010 a9fe bd54 0800 fa51 0200 a658 aaaa aaaa ...T...Q...X....
0x0020 aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa ................
0x0030 aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa ................
0x0040 aaaa aaaa aaaa aaaa aaaa aaaa aaaa aaaa ................
0x0050 aaaa aaaa aaaa aaaa aaaa aaaa ............
```

The problem with pure signature based systems showed up shortly after their arrival. First was that sometimes the packet signature was not unique to other network traffic, and would alarm to valid traffic. Second, sometimes they would alert to attacks a security administrator does not care about. Do you need to know if someone is attempted a Windows rootkit attack on your Linux server? Maybe so, but it should not be a critical level event. Third is that sensor would not detect on attacks it has not seen before. In other words the detection was only as good as the last signature update. The response from the network signature companies has been good, but nonetheless, always after the exploit was released. The biggest issue with pure signature based systems showed up after years of use. Simply put, the rate of exploits in the wild became unmanageable. When NIDS first came into existence, there were less than 600 known exploits. The chart below shows the annual trend of vulnerabilities announced. Keep in mind that for every vulnerability found, there may be several exploits released, and that a NIDS must watch for traffic from each exploit, not the vulnerability.

Vulnerabilities reported from cert.org

Vulnerabilities reported

1995-1999

Year	1995	1996	1997	1998	1999
Vulnerabilities	171	345	311	262	417

2000-2003

Year	2000	2001	2002	2003
Vulnerabilities	1,090	2,437	4,129	3,784

Total vulnerabilities reported (1995-2003): **12,946**

With multiple exploits per vulnerability found, the race was overwhelming for signature based NIDS to keep up. The sensors that already had performance issues decoding all network traffic were more burdened by an ever growing signature database. This spawned a new approach to NIDS.

The second generation of NIDS used rules rather than signatures. A rules-based NIDS still analyzes each network packet, but now compares the packet signature to a set of rules rather than exploit signatures. These rules are set up to detect suspect conversations (from unknown sources to known assets), or unwanted conversations (detection of backdoor programs on the network). The detection has shifted from watching for exploits, to watching the results of what exploits might do if they were successful. For example, a rules-based system would not detect the packet signature of an exploit aimed at loading a back door program, but would detect the new service (i.e. the backdoor program) establishing

connections and alarm on that. A rules-based system can also be used to detect on network policy violations in addition to attacks. For example, a Peer-to-Peer file sharing application (like Kazaa) is not an exploit, but as a security administrator you may want to know if that traffic is flowing on your network. A policy rule could be created to do just that. Huge communities of security experts have taken the second generation of rules based detection to the next level by co-development of basic network security rules. The community also continues to expand the rules as new exploits are released that are not detected by existing rules. The best place to learn more about rules based detection is www.snort.org. Although rules detection is not foolproof*, it can detect attacks that it has not previously seen by nature of a policy violation. It can also detect a wide range of exploits written around a single vulnerability with a single rule.

**"Nothing is foolproof because fools are so ingenious",* Mark Twain.

The third generation of NIDS used protocol anomalies to detect attacks. The concept is a monitoring system that understands "good" use of valid protocols, and can detect on misuse. "Good" use is defined as a network packet that conforms to the RFCs in place for proper protocol use. RFCs, or Requests for Comments (http://www.faqs.org/rfcs/) is the standard by which all applications agree upon to communicate. By adhering to the standards, applications from different manufacturers can communicate. A protocol anomaly NIDS can identify attacks by seeing inappropriate use of allowed protocols on the network. The benefit is that the system can alarm on attacks that it has never seen before based on RFC non-compliance. The problem with these systems is that not all application developers adhere strictly to the standards, so the system may alarm on valid traffic that is communicating inappropriately. Here is an example of how a protocol anomaly system may detect an attack. It is very common for web exploits to attempt to add unwanted characters to the URL stream to gain extra privilege, or overflow the buffers of the web service. For example, the following link in a browser would cause a vulnerable web server to show the directory listing of the C:\ drive:

http://VulnerableSystem.com/scripts/..%c0%af../winnt/system32/cmd.exe?/dir+C:\

This attack is known as a transversal attack (the dot-dot-slash or **../** is a dead give away) and in no way conforms to the usage defended by RFC for HTTP usage. This attack was fixed a long time ago (win2k SP3), but is still seen in regularity. New attacks of this type that misuse HTTP protocol can easily be detected by a protocol anomaly system even if it has never seen the attack before and no signature exists.

One of the latest methods of NIDS uses pure statistical analysis to determine attacks. The system is "trained" or can "learn" normal communication patterns and alarm on anomalies. Let's take for example an average web server. Normally, outside persons will request a web page be served from the web server (inbound HTTP requests over port 80 to the web server). If, all of the sudden, the web server started accepting IRC (Internet Relay Chat over port 6666), one could quickly deduce that an attack had occurred and the web server was compromised. Although the accuracy is very high, the time in the attack life cycle is past critical, the system is no longer being attacked, it has been compromised. Modern systems of this type have evolved to give indicators of early warnings better than this example, but still rely on traffic flow analysis to determine probing or other early warnings. A major benefit of this type of system

above all other type of NIDS is its ability to detect non-traditional attacks against assets. Most NIDS can alarm on attacks from exploits or policy violations, but are blind to internal systems misuse. Say for example, a disgruntled employee decides to use a file transferring protocol (like FTP) to transfer the contents of the Research and Development database to an Internet server owned by the company's competitors. The disgruntled employee never used an exploit, nor did he likely violate a network usage policy, however, this security incident could cost a company more than any automated attack or worm ever could. A statistical anomaly detection system would see that this user normally did not transfer very large volumes of data to external sources using FTP, and would alarm to this "new" event on the network. This type of detection can be incredibly useful for watching for these types of insider attacks.

The NIDS category that is most prevalent today is a hybrid approach. A NIDS system may use some signatures to identify common attacks, and rules to put "watches" on critical systems. They may also use the "best of" protocol anomaly detection to catch the type of protocol misuse that is most common (instead of alarming on non-RFC compliance, alarm on known RFC misuse). These types of systems can help reduce false positives by creating an alarm based on multiple means of detection, and can provide the security admin with the type of monitoring that the security model intended.

Although these systems provide some real value, they still all rely on the "human" factor to watch alarms and respond appropriately to protect company assets.

Traditional Host Based Intrusion Detection

The concept of traditional host based detection is simple. The host based intrusion detection system, or HIDS, relies on something from the host changing, or generating an event log. The host-based system will then take that information and determine if an attack against the system is taking place and alarm if necessary.

The earliest HIDS used the concept of a "file watch" to determine an attack. The HIDS tool would be loaded on to a critical server and tuned to watch critical system files for modification. The concept is that if an exploit were successful, it would have to modify a system file to escalate privilege or further "root" the box. Once a critical file was modified, it would trip the alarm and notify the system administrator. Many systems of this type could also give an automated response option to have the HIDS tool copy a known good system file from another source back to the system with the modified system file. As effective as this method is, it is without a doubt a "last line" of defense.

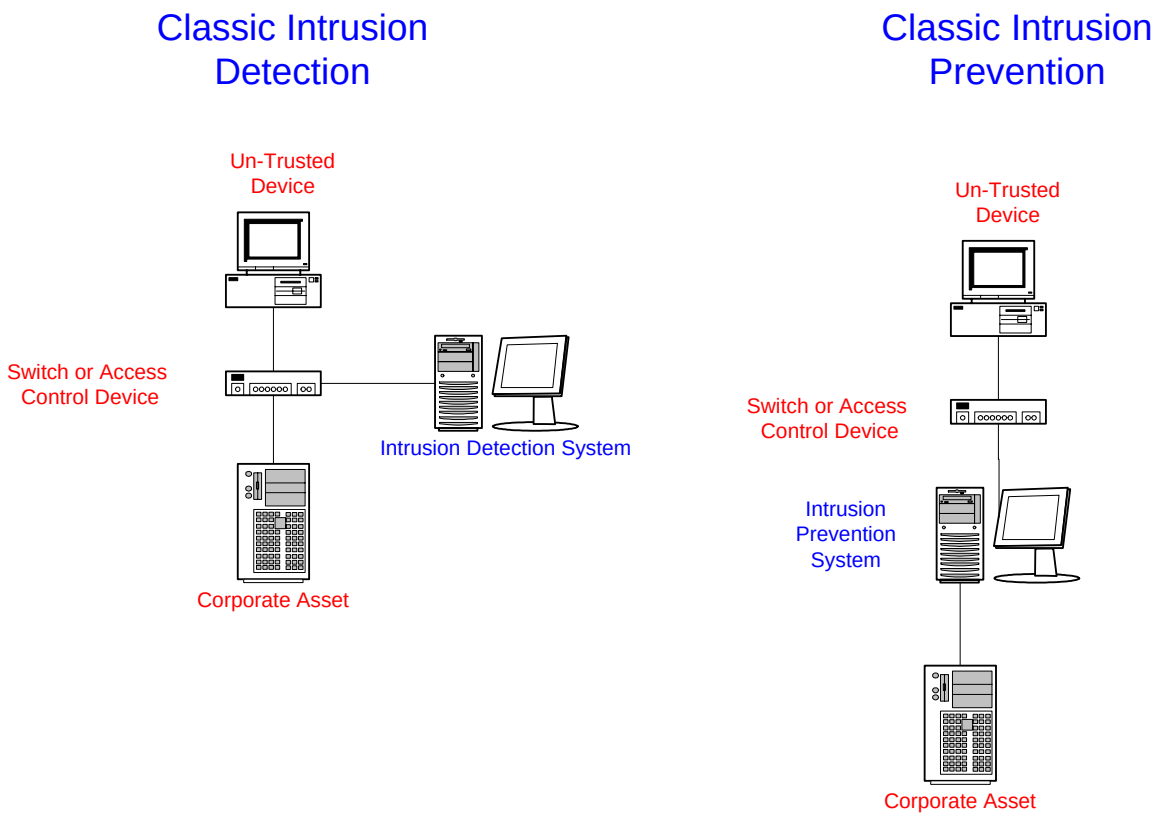
The next generation of HIDS allowed the security administrator to set strict policy on system and resource use. The HIDS agent could be tuned to watch the system registry and event logs for changes in the operating system from unprivileged users. The concept is the HIDS agent can watch for new programs being installed, or system values (or files) being modified and compare the new host action against a set of policies for use. If a policy was being broken, the tool could either alarm, or take an automated response, such as reverting the system to the state before the change. This is an effective method to enforce domain like security policies on a critical host. The draw back to these types of systems

was the configuration of policy. In order to have the alarm trigger, the system administrator would have to configure every desired behavior inside the HIDS management. The chance for policy mis-configuration was high, and therefore false positives was and still is an issue. HIDS is very effective at monitoring system event logs, correlating the events, and alarming on attacks that might otherwise be buried in the multitude of system logs.

A fundamental issue with HIDS is that they are always after the fact. An attack has to happen on the system (and usually be successful) before something will change, or generate a system log for the HIDS to alarm on. This can be compared to the earlier example of putting the security camera inside the bank vault. Yes it's accurate, but generally too late.

Network Based Intrusion Prevention

So what makes a network based intrusion prevention system (NIPS) different than a detection system (NIDS)? Let's start by defining the fundamental differences. The first thing to consider is the difference in network architecture.



As the diagram shows, NIDS is a single legged solution designed to strictly monitor network traffic and not make any decisions on whether or not to pass the traffic. NIPS on the other hand, is an inline device that can make decisions on

weather or not to pass traffic based on attack detection. This ability to gate unwanted traffic is the key differentiator.

Since NIPS in an inline solution, other network considerations arise. Questions about performance, network redesign, and availability are important. On most modern NIPS systems, they can detect at, or near, wire speeds up to several Gigabytes. They can also be placed inline and gate at OSI layer 2, commonly referred to as a bridge. This means that no network redesigns are necessary such as pointing all devices to a new route or gateway device. Modern NIPS also have the ability to fail closed relative to network traffic. These means if the NIPS appliance fails network traffic will continue to pass, but security has been lost. Secondary high availability units are also an option if loss of security is not an option.

How NIPS gates, or stops unwanted attack traffic is a greater consideration. Some early attempts at a preventative approach tied NIDS systems into firewalls. Based on detected attacks, the NIDS systems could add new firewall access control rules on the fly to block the attacker at the perimeter gateway. This was wildly unsuccessful for a number of reasons. At the time this approach was tried, NIDS were very inaccurate with a high number of false positives. Giving firewall control to a device this like this had little chance of increasing security, and a high probability of blocking valid production network traffic. How they did this is important. The systems of this type would add an access control rule to the firewall either blocking the entire IP from entering, or blocking a specific service from that IP (such as web traffic). Here is an example of why that is a bad method. Say your company has a major business partner that uses a common networking method called NAT (network address translation). NAT is very useful in hiding a large number of private computing devices (like users workstations) behind a single public IP address. Let's say for this example that the business partner is NAT'ing 5000 users behind a single public IP. If one of those users triggers the IDS system (say with an infected system or invalid use of protocol), the IDS will add a firewall rule to block that one attacking IP. The problem is behind that one IP is 5000 valid business users that have just been cut off from doing business with your company.

Modern NIPS takes a totally different approach stopping unwanted traffic. They only drop the unwanted packets from the offending sessions. This means given our example above, if that one user triggered the NIPS system, only the infectious traffic from that user would be dropped. The infectious user (as well as all the other users) would still be allowed to communicate with valid traffic. So one can say this is packet level detection and prevention, not access control prevention such as earlier systems. This is a one key reason modern NIPS systems are successful.

How a NIPS detects an attack is still important, and yes false positives are still a reality. Modern NIPS use a hybrid approach to detect attack traffic. A key difference however, it that they now use signatures that are based on vulnerabilities, not exploits. The chart from cert showed the growing number of vulnerabilities announced over the years. As mentioned earlier, there can be a large number of exploits released for each vulnerability found. By writing signatures on vulnerabilities, NIPS can add protection (hopeful) before actual exploits are released into the wild. Secondly, this helps keep the amount of data needed in the inspection engine considerably lower.

One could argue that Intrusion Prevention changes the classical incident response job duties. In the classical security monitoring and response roll, a Network Intrusion Detection System alarms a security administrator about an attack, and that administrator must respond to the attack manually to try and stop it. With NIPS, the system still alarms to attacks, but the alarm notification is in the context of "such and such traffic was detected trying to attack this asset, and it was blocked". With alarms of this type, is a human response needed? In an environment where security is paramount, the answer is likely yes, human response is needed to verify the automated response, and if also gather forensic data in a timely fashion so that it can be used to protect against future attacks, or identify a greater threat. In most environments where security is a concern, not a priority, no human interaction is really needed. A diligent response to the automated system would be to run reports once a day or week to validate the automated responses. This would free up security administrators to do other tasks besides watching a classical IDS alarm monitor.

Host Based Intrusion Prevention

The Host based Intrusion Prevention System (HIPS) market is the newest of the bunch. The security model of "last layer" of protection still applies since the HIPS agent runs on the host or operating system of the asset that is to be protected. The new twist is that the HIPS tool can detect attacks on the host, and stop the offending process before it executes.

The method of attack detection has also changed from the traditional HIDS model. HIPS no longer requires that a service generates an event log or system file changes before it can take action. The actual method of detection varies depending on manufacturer, but the common method is a rules-based approach to attack detection. The HIPS tool has a predefined list of "allowed/disallowed behavior" rules that ship with the product. These rules know how an operating system or application should behave. If the application starts to "misbehave" then a rule is triggered, and the offending process is killed at the kernel level, before it can cause harm.

The theory behind a rules-based HIPS is that vulnerabilities and exploits always change, and at an alarmingly high rate. However, the actions that those exploits do are fairly constant. For example, in 1988 the Moris Worm (the Internet's first worm) did a buffer overflow to the Finger service, which prompted that service to spawn a command shell that executed code to find it's next victim. In 2001, SQL Slammer did a buffer overflow to the SQL service that spawned a command shell to execute code that found its next victim. From this, a common sense rule one could derive is that no services (like SQL or Finger) should be allowed to spawn command shells that accept connections or try to make outgoing connections. That is the heart of how a typical HIPS works. Another example is email worms. A common sense rule would say that the email application (like Outlook) is not allowed to download content (like files) that execute automatically (like worms) and then spawn command shells that do things like access the email client's address book. A rule like that will stop every email worm or virus from doing harm with a signature of the worm.

Other HIPS systems use an observational approach. The theory is that an agent will run on the host and observe all system calls, registry entries and services communications. After the observational period, the agent can be set in enforce

mode and any calls outside of the observed behavior are killed at the kernel level. Although this method can be very effective, it takes a classic "strict deny unless otherwise allowed" model. These types of systems can provide the greatest security at the cost of higher administration of valid applications that make inappropriate system calls.

Yet another method is the hybrid approach. A hybrid HIPS may use a combination of rules, application behavior and signatures to detect an attack. The major benefit to these types of systems is the ability to absolutely identify attacks by name that it has seen before (or that a signature exists). Rules or behavioral based systems can only show the actions that were suspicious and stopped. It is still up to the administrator to cross reference the blocked behavior with known attacks to try to identify the attack. Of course, hybrid systems only offer this benefit for older attacks, but as any IDS/IPS system admin can tell you, the old attacks are not dead and still propagate.

What Works Today?

All of the methods discussed in this paper can add extra protection against attacks on assets, if used properly. The real question is which method can provide the greatest balance between security and administration burden for my company's needs?

Host-based Intrusion Prevention provides the greatest security return of any of the other methods mentioned. It may not be the cheapest to purchase or deploy, but the benefits go beyond host layer protection. Going back to the original model of prevent future attacks, protect against ongoing attacks, and monitor for new attacks, a good HIPS system can add value across all three layers. The most notable is across the prevention layers. For hosts, the most common prevention method is patching of operating systems and applications. In a large organization, it may take a large team to test and deploy new patches. And in every case, the time to deploy the patches is at the application developer's convince, not your company's. When a patch is announced and released a company must respond with great haste, and the start time is always "right now". There is no luxury of waiting if you wish to prevent an attack. One could argue that if you have a system that is not vulnerable to an exploit because of a HIPS system, do you need to patch it? Most companies have decided yes, continue to patch, it is a good security practice, but now you can patch when it is convenient, not when the fire bell rings for a new vulnerability. Typically, companies can restructure their patch management to once a quarter with a quality HIPS product. This benefit of patch management control is why HIPS makes it to the top of the list.

From a protection standpoint, a system with a quality HIPS product is protected against a new attack before a vulnerability is announced and a patch is issued, and without and update of any kind from the HIPS manufacturer. This is sometime termed "zero day" protection.

A quality HIPS product can also provide classic host monitoring such as system calls, privilege monitoring, and file watches. Because HIPS can do all of this with

no security updates, it wins my vote as adding the most security return on investment.

NIPS systems provide real value to a company's overall security strategy. They have a good chance of being ahead of the attack wave (since they patch on vulnerabilities, not exploits), but still rely on update to be truly effective. The major benefit that will continue to be the number one reason companies buy NIPS is their ability to stop the massive propagation of worms inside a companies perimeter firewall. The latest worms have found their way inside the perimeter firewall (through email, home users on VPNs and business partners) and caused real harm to a company in the form of system outages and emergency containment teams. A quality NIPS can stop that issue if deployed at the core of a network.

Pros and Cons

HIPS (Host-Based Intrusion Prevention Systems)

Pros

- Provides "zero day" protection against attacks no one has seen before
- Requires little to no security updates in an annual period, helping to reduce the cost of ownership
- Prevents attacks from executing on a host at a kernel level rather than detect the outcome of a successful attack
- Can reduce work load from other security teams such as patch management
- Can be tuned to add application specific protection to "home grown" applications that traditional products will not detect

Cons

- The cost of the total system can be expensive since an agent is needed for all critical servers and/or workstations
- The deployment time to reach every server/desktop can be long
- The product requires tuning after the initial installation to be a functional security tool
- Can stop legitimate applications from running if not tuned properly
- New applications may need to be tested against the HIPS before they are deployed
- HIPS don't identify the attacks that they stop by name or clean up infections.

NIPS (Network Intrusion Prevention Systems)

Pros

- Can stop the propagation of worms if deployed correctly without stopping production traffic
- It will protect against new attacks before exploit code is released (in most cases)
- Will reduce the cost of incident response (since most incidents are responded to automatically)

Cons

- The cost of deploying NIPS at the core of a network can be vary expensive
- Since NIPS are an inline device, they create a single point of failure. Although there are methods to deal with system failure, the most common approach is to add redundant units since all network traffic passes through the NIPS
- NIPS still rely on security updates to be truly effective

NIDS (Network-Based Intrusion Detection Systems)

Pros

- Can add the best visibility into network events that raise security concerns above and beyond exploit code
- Anomaly based systems can provide attack detection even on systems that use encryption (they do not see the exploits, they see the resulting abnormal traffic flow as a result of a successful attack)
- Watching traffic flow with a rules based system can help enforce company network usage policy
- May be all that is needed to meet an audit requirement

Cons

- The cost of the "human" factor is high to monitor events and respond to incidents
- Unless an incident response plan is designed and staffed, an IDS provides little to no security value
- A successful deployment will involve extensive tuning of an IDS to minimize false positives

HIDS (Host-Based Intrusion Detection Systems)

Pros

- Can detect valid system usage that violates company security policy
- Can alert on system changes such as critical file modification
- Some automated responses can return a comprised system to a state before the attack was successful

Cons

- The cost of deployment and management is high since every host has to be touched and policy developed
- Most commercial HIDS manufacturers do not make a desktop product, so coverage is only for servers
- The detection is generally "after the fact" or late in the response curve. A successful detection comes from a successful attack attempt

Incident Response Considerations

Classical incident response is often the most overlooked consideration in choosing an intrusion detection system. Companies that fail to map out their incident response plan prior to purchasing an IDS system are often in for what the industry terms, "shelf-ware", or software that will never be used. The National Institute of Standards and Technology (NIST) has an excellent framework document that can be used to plan out an incident response initiative that is appropriate for your company's security needs. A few of the key points will be laid out here for discussion on how it relates to detection systems.

The Basics of Incident Response

The process of incident response has one purpose, protect the company assets against threats from risks of attack. This is done by first identifying the assets, not only by name, but by worth. Once that is done, a formal process should be created to prepare to defend against any threats that could do harm to the assets. This data is typically found by doing a vulnerability assessment. Next a detection system(s) to watch the assets for attack, and a formalized reaction plan an administrator can use to take the correct steps to protect against detected attacks is needed. Last is the constant tuning of this process for improvement as system change and threats grow.

Basic Process Flow for Incident Response

If done right an incident response plan should be able to categorize attacks in to set categories of attack type so that the containment plan(s) can take proportional measures. Typical attack types include:

- DoS (Denial of Service)
- Malicious code (Worms)
- Unauthorized access
- Inappropriate usage
- Multiple components

An incident response plan should also contain the answers to common forensic questions such as:

- Who attacked you?
- When did it happen?
- How did they do the attack?

- How widespread is this incident?
- Did this happen because you have poor security practices?
- What steps are you taking to determine what happened?
- What is the impact of this incident?
- What is the estimated monetary cost of this incident?

How Does Incident Response Differ from Detection Systems to Prevention Systems?

When evaluating an IDS or IPS system, it is important to consider how the product will fit into your incident response (or possible be your only tool used for incident response). Will the tool be able to categorize events how you care to see them? Or will it be able to quickly present the answers to forensics questions?

Classic intrusion detection systems tend to map better to traditional incident response plans and provide greater forensic data on events. However, the cost for this greater data is increased manpower for containment and cleanup when an event is correctly detected. On the whole, IDS can only be one part of an incident response plan. Containment and eradication must be done by other means.

An IPS system on the other hand can take over two major parts of incident response, detection and containment/eradication. One could argue that IPS is changing information security for this very reason. When building a business case around IPS consider the impact to an existing incident response plan. Can it be streamlined? Can some resources dedicated to response be redeployed elsewhere for a cost saving to the security team? Those that adapt their incident response plans to systems that can take automated responses will no doubt give their companies a competitive advantage.

Enterprise Considerations

Products in the IPS space are definitely targeted to the enterprise customer. Host based systems will scale to greater than 100,000 users and network systems can handle multi-gigabyte throughput. Some of the considerations that remain are:

- Modification of incident response plans
- Architectural placement of NIPS to gain maximum benefit
- Political considerations of a host agent on a system that security does not own, or an inline network device on a network that security does not own.
- Cost of deployment and hardware. Host based deployment can take weeks or months. Hardware for network prevention systems have to be at the core for maximum value. Plan on two (or more) very high speed units for performance and availability.
- Modification of help desk process (for host agents)

Small and Medium Business Considerations

Although newer HIPS and NIPS are aimed at larger companies there is value that can be had for the small to medium business. The biggest benefit comes in the way IPS can take automated responses. Traditionally, small to medium businesses do not have full time staff for incident response, but have the same needs as larger companies. With an IPS system, smaller companies can get the major benefits of an incident response team without the team. This may not be considered “best practice”, but it certainly is diligent and practical for a smaller company.

Some enterprise HIPS manufacturers have reached out to the small to medium market and created bundles that make it cost effective to protect just a few systems. This may be the cheapest route to go if the only concern is a few critical hosts. Also some HIPS vendors have reached out to the desktop market and can be a cost effective replacement to personal firewalls on laptops and workstations.

Conclusion

Is intrusion prevention changing information security? In many ways it is doing just that. It allows companies to rethink traditional incident response plans. The ones that find the most cost effective use IPS over IDS will set themselves and their companies ahead of the curve. It is also placing the control of system patching back into the hands of the customer, not the application developer. From a security standpoint, the only reason to patch a system (or application) is to prevent it from being attacked. If a HIPS agent can do that to a system, why patch it? Patching is diligent and should continue, but will be a more cost effective process to the company that can patch once a quarter versus once (or twice!) a week.

There is still a "cooking" process for the IPS market to be accepted by the main stream business consumer. Third party reports like the one from NSS (http://www.nss.co.uk/download_form.htm) have validated the market for network based intrusion prevention systems and recent acquisitions from Cisco (http://newsroom.cisco.com/dlls/corp_012403.html) And Microsoft (http://www.nwfusion.com/news/2004/0301microsoftrsa.html) in the HIPS space show that some of the larger security players are moving in this direction.

IPS systems have already changed the way information security works. The only thing that remains is how wide spread the change will be. Will this cause traditional IDS vendors to rethink or fold from the market? Will new hybrid highly accurate IDS systems close the gap on customer demand and have the greatest value?

References

Guy Bruneau, “The History and Evolution of Intrusion Detection”
http://www.sans.org/rr/papers/30/344.pdf

NIST, “Computer Security Incident Handling Guide” special publication 800-61
http://csrc.nist.gov/publications/nistpubs/

CERT, statistics 1988-2003
http://www.cert.org/stats/