The Information Technology Security Arms Race

Dr. Steven Hofmeyr
Sana Security

Increasingly, new attack technologies and tools are overwhelming existing information technology defenses. This ongoing arms race requires new technologies for the defender. In this article, I describe how an intrusion prevention system (IPS) fills this need, and I dissect the sometimes confusing world of IPS, describing the various technologies available, and where and how to deploy them.

Security for information technology (IT) is subject to an ongoing arms race between the attackers and the defenders. As new IT systems are developed and deployed, attackers find new weaknesses in these systems and new ways of exploiting the weaknesses. Defenders must keep innovating to keep up with the attackers; without ongoing innovation, IT systems will become crippled by attacks.

New Attack Technologies

Recently, attackers have devised a set of new tools to make it easier to attack systems and evade defenses. A good example is a group of tools known as *binary differs* that determine differences in binary code [1]. Binary differs are typically used to determine the difference between a patched version of an application and an unpatched version. This enables the attacker to determine what vulnerability was patched, and how.

Armed with this information, an attacker can rapidly develop an exploit for that vulnerability, often within a matter of hours. Consequently, as soon as a new patch is announced, an attacker can have an exploit for that patched vulnerability within a matter of hours. This is a problem because defenders can rarely, if ever, patch that fast, even with automated patch systems. Patches have to be tested before being deployed and that can take days, even weeks. With the advent of binary differs, the defender will lose the patching race every time. Patching is no longer a viable defense strategy.

Another kind of tool is the automated attack framework. With these frameworks, an attacker with little or no technical expertise can easily launch a variety of attacks. The best example is the free open source tool Metasploit [2], which allows an attacker to scan a system for vulnerabilities, and then gives the attacker the choice of various attack modules to click on to launch an attack. In addition to choosing the type of attack, the attacker also gets to choose the type of payload, which can be any one of a variety of malicious software (malware for short) such as a Trojan horse or a root kit. Such automated attack frameworks make it very easy for anyone to launch attacks on systems and to encourage the rapid dissemination of attack information.

Not only is it much easier to launch attacks, but the payloads of those attacks are becoming increasingly sophisticated. We are seeing a proliferation of malware designed to do nasty things to the victim such as stealing information, spreading rapidly, and clogging networks, and serving as bot[1] networks that can be used to launch denial-of-service attacks or function as spam relays.

> *"Before harm is done, an IPS [intrusion prevention system] will proactively detect attacks and prevent them, providing a powerful new layer of defense."*

Often this malware will be tailored to a particular attack, for example, a Trojan horse may be specifically designed to steal information from one organization, to be used only in that circumstance. This means that Trojan horse will be something new and not detected by signatures in traditional antivirus or antispyware tools. If a signature is developed (assuming the Trojan horse is ever discovered), it will be useless because the Trojan horse will never be used again.

It is a trivial matter to develop customized malware: in the simplest case, the attacker can use Morphine [3], a hosted service that will obfuscate any piece of malware for a small fee (about $36), ensuring that the malware is not detectable by any of the standard signature-based antivirus systems.

New Technologies for Shielding Vulnerabilities

With the new technologies attackers are developing, we can no longer rely on reactive technologies such as patching and static signature scanning. These technologies are too slow to prevent widespread damage, and too dependent on human expertise to be able to scale to the complexity and growing size of IT systems today. We need a new approach.

We can move forward in the arms race by using the proactive technology in an intrusion prevention system (IPS). Before harm is done, an IPS will proactively detect attacks and prevent them, providing a powerful new layer of defense. An IPS can be used to shield vulnerable systems, giving time for administrators to fully test and deploy patches at their own convenience, and buying time for human operators to better understand the threat. Further, an IPS reduces the need for human expertise and saves on expensive and hard-to-scale human resources. Properly deployed, IPS can protect against a wide variety of threats, including surreptitious malware and fast-spreading destructive worms and viruses.

The world of IPS technology can be very confusing. There are a host of different technologies available, and it can be very difficult for administrators to determine which are the most suitable for their needs. To add to the confusion, IPS can mean different things to different people. In this article, I discuss the relatively new technologies (within the last few years) that are commonly acknowledged to comprise an IPS. I do not include technologies such as firewalls and signature-based antivirus systems that have been around for many years and are failing to secure IT systems. The intent of this article is to clarify the IPS landscape; to this end, there are three aspects that need to be considered: (1) where to deploy IPS, (2) what kind of IPS technology to deploy, and (3) how to deploy the chosen IPS system. I will address each of these in turn.

Where to Deploy IPS

There are basically two places to deploy IPS: on the network or on the host computer (see Figure 1). The network IPS has several advantages: It is usually a single device or appliance that is both easy to manage and easy to deploy. All the security administrator need do is drop a box onto the network segment; usually no permission is required from the application owners because there will be no conflicts with software installed on the hosts. Furthermore, a network IPS provides broad coverage if placed at an appropriate choke-point: A single appliance can protect a whole segment with multiple hosts.

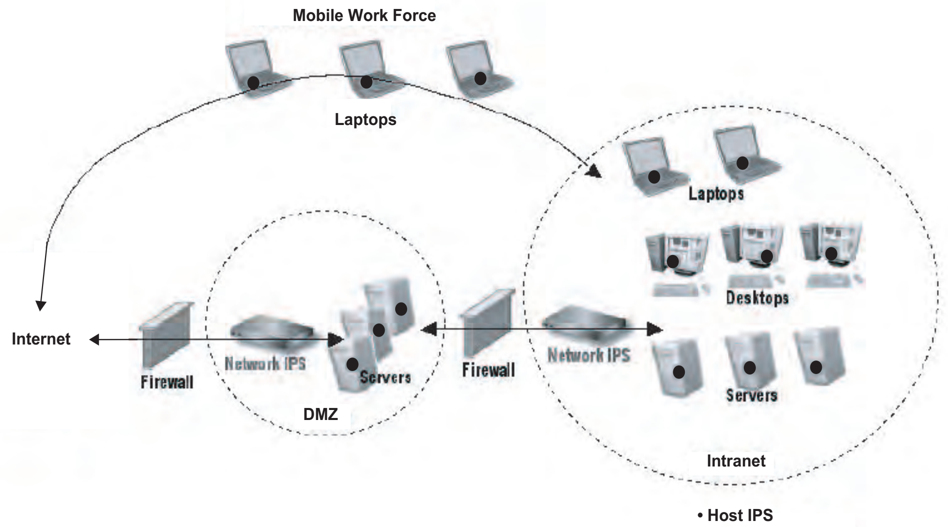
However, there are limitations to the network IPS. The broad coverage can also be detrimental because failure of a single appliance will cut off traffic to a whole subnet, a consequence of the fact that the IPS has to be inline to be able to drop malicious traffic. There is also a performance tradeoff because the more hosts a single appliance protects, the more traffic it will have to process; a network IPS that does sophisticated traffic analyses can rapidly become a bottleneck, unable to cope with high traffic volumes. When deploying network IPS, an administrator should be well aware of this tradeoff.

By contrast, IPS on the host does not suffer from the same problems. Each host will have its own IPS software so the security processing is distributed across all machines and performance is no longer an issue. Further, a host IPS tends to be more robust, because failure of one system will only have a small affect on the overall performance of hosts in the network.

There is another powerful driver toward using host IPS: de-perimeterization. Network IPS requires a clear notion of a network perimeter, and unfortunately the perimeter is collapsing with the advent of distributed applications – such as Web services – and more business being done over the Internet necessitating closer links with partners, suppliers, etc. This trend is so powerful that a high-level industry organization, the Jericho Forum, has been created to promote de-perimeterization [4]. The loss of the perimeter is exacerbated by the increasingly mobile work force. Users that work from outside the corporate or government network can easily pick up malware infections and bring those into the secure environment, infecting all vulnerable hosts behind the firewall. This is another compelling reason for the rapid adoption of host IPS.

In typical deployments, however, an organization will use both network and host-level IPS. This layered approach generally gives the most comprehensive security, although organizations deploying multiple layers should be aware that the more layers in place, the more chance there is of false positives, and the more difficult it is to manage the system. Any security architecture using IPS will also include network-level defenses to protect against network-level threats such as denial-of-service attacks and eavesdropping, but these are not generally considered part of IPS, and are not discussed in detail in this article.

Figure 1: *Typical IPS Deployment*



What to Deploy: IPS Detection Technologies

Although IPS is designed to prevent attacks, and not just detect them, it is still reliant on its underlying detection technology: Only that which is detected can be prevented. Attacks are detected and then prevented by an IPS. For example, a network IPS that drops packets[2] from an attack has first detected the packets and then prevented the attack by dropping the packets. Similarly, a host IPS that prevents applications from making system calls has detected the system calls being attempted, and prevented them from being executed, hence preventing the attack. The discussion of IPS technology is greatly clarified by separating out the detection from the prevention aspects. In this section, only detection technologies are discussed; in the section, “What to Deploy: IPS Prevention Technologies,” prevention technologies are discussed.

There are a variety of detection technologies available, each with its advantages and disadvantages. Many of these have long been in development and used in intrusion detection systems, and are little changed although they are used in an IPS. Often the best solution is a layered approach in which multiple technologies are used to complement each other’s strengths and cover each other’s weaknesses.

Signature-Based Detection

Signature-based detection relies on human experts knowing and understanding what particular exploits look like so the experts can encode signatures for those exploits. Typically, signature-based technology is used to scan static data such as network packets. Signatures are *exploit-focused*, meaning they will not protect a vulnerability, but only particular exploits of that vulnerability. For example, there are many ways to write a buffer overflow, but typically a signature will only look for one way of doing so; if the attacker changes the code in the exploit, the signature will not detect it.

Generally, there are many different ways in which a vulnerability can be exploited, and a signature will only protect against one of those. Because of this, signatures only protect against yesterday’s attacks – those that we already know about. Signatures will not protect against zero-day (unknown) threats, or against modified or mutated malware. In addition, signatures require extensive human expertise and constant updating, an approach that does not scale well. Fundamentally, signatures are one of the poorest choices for detection in IPS because of the human overhead, and the fact that this approach fails continually. For these reasons, signatures are rarely used in IPS.

Expert-Based Detection

Expert-based detection relies on human experts defining a set of rules that dictate what behaviors are normal, and hence, allow for applications and operating systems. The expert-based approach relies on humans understanding the applications and systems to be protected, but requires little or no knowledge of attacks. This approach can be very effective at protecting against zero-day threats because these generally cause deviations in application or system behavior. However, this approach requires that the people defining the rules know a great deal about the applications and system to be protected, which can be problematic for complex and custom applications.

When an application is complex, expert-based detection ends up being detuned to reduce false positives: The rules become increasingly generalized to the point where they offer little or no protection at all. For example, an IPS might have a rule preventing applications from loading drivers into the kernel. However, some applications may legitimately need to load drivers, which would result in a false positive. A common response is to turn off the rule altogether because of the complexity of determining which applications should be allowed to load drivers. Relaxing the rule will allow any application to load drivers with consequent security risks.

Generally, the expert-based approach works best when applied to well known, relatively simple applications with well-known behavior that does not deviate significantly from one system to the next, or from one use to the next.

Specification-Based Detection

Specification-based detection gets away from dependence on human expertise by deriving a set of rules governing normal (allowed) behavior by either statically scanning protocols, or application binary or source code. This has the advantage of avoiding the overhead of human involvement and human bias and error, which can plague approaches such as expert-based detection. Although this approach can be powerful, it is limited in that it is often too generalized.

When protecting applications, the protection will usually only prevent injected code, and not detect (and hence be able to prevent) the large number of attacks that exploit other types of vulnerabilities such as misconfigurations. In the case of protocol-based network protection, it is limited in that it cannot detect flaws in the protocol itself, and often networking components and applications will not implement protocols strictly according to the specifications, resulting in many false positives.

Autodidactic Detection

Autodidactic detection attempts to automatically derive a normal model like the specification-based approach, but uses learning or auto-configuration to refine the normal profile so that the approach does not suffer from excessive generalization. The normal model is derived from monitoring systems during normal usage and learning the normal model of network traffic, or host behaviors. This learning can be done in a production environment or a quality assurance (QA) laboratory.

> *"Where an organization is uncertain, they should follow a process whereby they first test the performance of the IPS. On the host, this means measuring the impact on running applications and memory/disk usage, and in the network, this generally means measuring network throughput and latency."*

Autodidactic systems tend to be scalable and offer very good protection for all kinds of applications and networks because of their ability to automatically learn all the nuances of complex behavior. However, there are several limitations to this approach. First, only stable behaviors or network patterns can be learned, so in highly variable environments or with highly variable usage patterns the system can prove to be inaccurate. Second, learning takes time, during which the system can be vulnerable to attack. This can be offset by doing the learning in another environment such as a QA lab or a test lab, but how well that works depends on how closely the lab environment represents the real environment.

Innate Defense Detection

Innate defense detection is most similar to signature-based in that it protects against specific attacks, but it differs from signature-based in that it is *vulnerability-focused*: It detects a whole class of attacks rather than instances of a particular class. A good example is technology for detecting buffer overflows: There are many different attacks that exploit buffer overflows, necessitating many different signatures, but an innate defense for buffer overflows will stop every kind of buffer overflow without requiring any knowledge of specific attacks.

Generally, an innate defense will detect one class of threats, and do so effectively, with high accuracy and little or no overhead in terms of tuning or configuration. This approach is very important for protecting systems by default on a large scale since this kind of technology can easily be distributed with the operating system. It can be used stand-alone to incrementally improve protection, but because the protection provided by innate defenses is not comprehensive, it is best used as part of a layered solution.

The major shortfall of this approach is that not all threats are amenable to prevention through innate defenses; there may not be any way of developing a generic method of detecting all attack instances within a given threat class.

What to Deploy: IPS Prevention Technologies

The hardest aspect of prevention is that it should be immediate (proactive) whenever possible. This can be problematic if the system has to initially gather information to determine if an attack is actually happening; too much delay in the response will result in damage. There is often a trade-off: The more information is used to determine if an attack is happening, the more accurate the IPS, but the more potential there is for the attack to cause harm. Another important aspect of prevention is that it can cause damage when reacting to false positives because it can block legitimate behavior or data. One way to minimize the potential harm is to have responses that are as fine-grained as possible, for example, to block a particular system call, rather than kill a process.

However, fine-grained responses can only go so far in ameliorating the damage done by responses to false positives. The problem is that currently all prevention tends to be all or nothing: Either an action is allowed, or it is blocked. For example, host IPS tends to block actions such as file accesses, network accesses, process starts, etc., and network IPS tends to drop packets, reset connections, and block particular Internet protocol addresses. To move forward, we need *benign*, recoverable responses. The goal is responses that can stop attacks, but allow the system to recover from false positives.

For example, databases have mechanisms for rolling back transactions; we can expect to see similar concepts in the future in IPS. Another method of benign response is using delays to slow down the rate at which attacks propagate. Research has shown that this can be an effective method of blocking attacks [5], and can also be very useful in slowing down virus and worm propagation [6]. Prevention technologies in IPS today are extremely useful and powerful, but we expect to see great improvements in the future.

How to Deploy IPS

Typically, the first step in deploying IPS will be testing and evaluation. The extent to which an organization does testing depends on many factors, including the resources available for testing and how comfortable an organization is with the reported and claimed functioning of IPS. Where an organization is uncertain, they should follow a process whereby they first test the performance of the IPS. On the host, this means measuring the impact on running applications and memory/disk usage, and in the network, this generally means measuring network throughput and latency. Stability can be an even more important measure: Does the IPS crash and block network traffic or bring down the host? A good IPS should never impact performance and stability unreasonably, although bear in mind that network IPSs are usually designed to support various speeds, and an IPS that processes traffic faster tends to be a lot more expensive.

Another important factor to test is accuracy. Does the IPS stop attacks effectively without high false alarm rates? One way to test this is in a lab by running the IPS on vulnerable systems and actively launching attacks against them. Although this gives some idea of accuracy, it is limited in that it gives very little idea of usability, scalability, and false positive rates in the real world. For example, it will often be a simple matter to configure an expert-based IPS in a lab, protecting standard applications – the accuracy may appear to be very high. However, deploying such technology out in the production environment may lead to many false alarms with a subsequent detuning and loss of accuracy of the IPS. The importance of evaluation in the production environment cannot be overstated.

Once the IPS is tested, the next phase is usually a limited deployment phase, using the IPS to protect a few systems such as those under high threat in the Demilitarized Zone[3], or those that are of little importance (hence false positives are acceptable). When the IPS has proven itself in the limited deployment phase, it can then be deployed across the whole organization. This process will obviously differ for network and host IPS.

Once deployed, the organization enters the maintenance phase; most IPS will have to be tuned whenever there are changes in the organization such as additions of new networks or machines, or reconfigurations of applications. An organization should plan for these changes knowing that they will have an effect on the IPS deployment. As stated before, some IPS technologies are much more adaptable than others (for example, the autodidactic approach) and are much easier to deal with during the maintenance phase.

> *"Once deployed, the organization enters the maintenance phase; most IPS will have to be tuned whenever there are changes in the organization such as additions of new networks or machines, or reconfigurations of applications."*

The Benefits of IPS

IPS is starting to be widely deployed in many different market sectors across many different organizations. There are three markets that are seeing immediate benefit from IPS: the financial sector; the government, including the military; and the health-care industry. All of these sectors are under increasing attack and feeling the pressure of new government regulations such as Sarbanes-Oxley [7], the California Senate Information Disclosure Bill 1386 [8], and the Health Insurance Portability and Accountability Act of 1996 (HIPAA) [9].

For example, military organizations are using the Internet and commercial off-the-shelf software to realize efficiency gains, but are consequently at risk from attacks that target platforms such as Microsoft Windows. It is vital to the security of the country that these organizations maintain high standards of protection; to this end, IPS is an essential part of the defenses. We are also seeing the potential for IPS in the mobile battlefield where unprotected mobile computers such as laptops would be prime targets for attack, especially if they are not used and not updated when in storage and then brought out rapidly for battlefield deployment.

Financial organizations are also among the early adopters of IPS. They find IPS particularly useful for securing unpatched applications. For example, many financial organizations mandate at least three weeks' testing of any new patch because faulty patches can bring down mission-critical servers. However, three weeks of exposure for vulnerable servers connected to the Internet will almost certainly result in compromise, so these organizations turn to IPS to enable them to adequately test patches while still ensuring their servers are protected.

In the health-care industry, regulations such as HIPAA require health-care providers to ensure the confidentiality of patient data. Deploying intrusion detection systems and other reactive technologies that only determine when an attack has happened after the fact is not sufficient because valuable patient data will already be stolen. Hence, the health-care industry is realizing critical benefits through the ability to stop information leaks before any harm is done.

Summary

IT security is an ongoing arms race. Recently, attackers have been gaining the upper hand with a new set of attack tools and techniques. IPS regains the initiative for defenders, providing a shield for unpatched vulnerabilities. This buys time to test and deploy patches, reduces human resource cost, and reduces security breaches and the associated costs.

But IPS can be complex. An organization should know where to deploy IPS, whether on the host or network (ideally, both), and the tradeoffs inherent in such a decision. Further, an organization should understand what sorts of technology are available and what are most suitable for its environment. In general, the best approach is a layered one that uses multiple technologies. Finally, an organization should plan for a phase of testing and evaluation, and should know how to go about rolling out the IPS.

The technologies described in this article all exist in commercial products, of which there are many. When considering deploying IPS, an organization should search for vendors in the IPS arena and solicit information from a set of vendors to ascertain exactly what they do. This article is intended to be a useful guideline in cutting through the marketing language and enabling users to understand exactly what a vendor's products are likely to achieve.

Implementing IPS will take effort and money, without doubt, but IPS is essential in today's threat environment. Without improved security measures, our IT systems will soon become worse than useless, and the costs of failed security will far outweigh the costs of IPS.◆

References

1. Flake, Halvar. "More Fun With Graphs." Blackhat Federal 2003, Tyson's Corner, VA, 1-2 Oct. 2003 <http://cansecwest.com/csw04/csw04-Halvar.ppt>.
2. Metasploit. 26 July 2005 <www.metaspoloit.com>.
3. Anti-Detection Service. 11 July 2005 <http://hxdef.czweb.org/antidetection.php>.
4. The Open Group. The Jericho Forum. 11 July 2005 <www.opengroup.org/jericho>.
5. Somayaji, A. "Operating System Security and Stability Through Process Homeostasis." Doctoral Diss. University of New Mexico, 2002.
6. Williamson, M., J. Twycross, J. Griffin, and A. Norman. "Virus Throttling." Technical Paper HPL-2003-69. Hewlett Packard Labs, 2003.
7. Sarbanes-Oxley. "Financial and Accounting Disclosure Information." Huntington Beach, CA: Sarbanes-Oxley <www.sarbanes-oxley.com>.
8. Sen. Peace, Assembly Member Simitian. "SB 1386." California Senate, 26 Sept. 2002 <http://info.sen.ca.gov/pub/01-02/bill/sen/sb_1351-1400/sb_1386_bill_20020926_chaptered.html>.
9. U.S. Department of Health and Human Services. "Office for Civil Rights – HIPAA." Washington: HHS, 1996 <www.hhs.gov/ocr/hipaa>.

Notes

[1] A *bot* network is a network of compromised computers that are remotely controlled by an attacker. Each computer runs *bot* software that enables an attacker to access the computer and control it remotely.

[2] This means not forwarding suspicious packets of network data.

[3] This is the part of the network that contains Internet facing servers, and is usually separated out from the main intranet, forming a buffer zone between the intranet and the Internet.

About the Author

Steven Hofmeyr, Ph.D., is chief scientist at Sana Security, which he founded in 2000. Sana Security is a market leader in intrusion prevention with its host-based products widely deployed throughout industry and government. Hofmeyer has also carried out research at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology (MIT) and the Santa Fe Institute for Complexity Studies. He has authored and co-authored many published papers on computer security, immunology, and adaptive computation. He has been an invited participant to several U.S. government workshops on future directions for technology such as the Joint Engineering Team Roadmap Workshop. In 2003, MIT's *Technology Review* named him as one of the top 100 young innovators under 35, and in 2004, he was named one of the 12 innovators of the year by *InfoWorld*. Hofmeyr has a doctorate in computer science from the University of New Mexico.

Sana Security
2121 El Camino Real
STE 700
San Mateo, CA 94403
Phone: (650) 292-7152
E-mail: steve@sanasecurity.com

More Online from CrossTalk

CrossTalk is pleased to bring you additional articles with full text at <www.hill.af.mil/crosstalk/2005/10/index.html>.

Security Issues in Garbage Collection

Dr. Chia-Tien Dan Lo, Dr. Witawas Srisa-an,
and Dr. J. Morris Chang
University of Texas at San Antonio

This article examines Java security models, describing security issues in garbage collection (GC), metrics used to predict program behaviors, and their relations. Heap memory attacks are introduced and classified into slow death and fast death categories. These are potential scenarios if GC is under attack. Experimental results show that a compromised system may result in GC being invoked more times than its normal counterpart. Furthermore, presented here is a run-time monitoring system that can detect anomalous program behaviors using the collected memory metrics. This can be a run-time throttle that controls program behaviors, and a postmortem diagnosis technique in case of heap memory attacks.

Attacks and Countermeasures

Zaid Dwaikat
Systems and Software Consortium, Inc.

Security attacks on information systems have become a standard occurrence directed against all components of a system, including people, networks, and applications. Attacks have gotten more complex while the knowledge needed to execute such attacks has decreased. Attackers look for the weakest links in each component; using sophisticated techniques and freely available tools, they exploit potential vulnerabilities wreaking havoc on information systems. To better defend systems, it is necessary to understand how they function and, more importantly, how attackers use vulnerabilities to compromise them. Information systems today are distributed, complex, and extensible. This article provides an overview of the most common attacks: attacks on people, networks, applications, and passwords.