

Defining the Rules for Preemptive Host Protection: Internet Security Systems' Multi-Layered Strategy

By Joshua Corman

Copyright© 2004 Internet Security Systems, Inc. All rights reserved worldwide

 **INTERNET | SECURITY | SYSTEMS®**
Ahead of the threat.

Introduction

Protecting desktop and server — or “host” — systems has rapidly become a high priority for organizations that want to ensure uptime and the availability of day-to-day business applications. In 2003, the average cost of a virus disaster’s impact rose approximately 23 percent, to \$99,900¹, a figure that’s worsened for eight consecutive years. This whitepaper serves to identify common problems associated with effectively protecting host systems and defines the components of a comprehensive solution offering a superior level of host protection. The first section describes the nature of modern threats to the host and the threat vectors used to attack host systems. Next, the whitepaper analyzes each individual technology available to combat host threats. Finally, it aims to establish that modern threats require a multi-layered host protection solution. Hopefully, through stronger information and understanding in the marketplace, organizations will be better equipped to thwart the spectrum of modern threats to host systems.

Understanding Modern Threats to the Host

When researching threats to host systems, it is important to understand the primary phases of a successful attack. In one popular model, attacks on the host are broken into three phases — penetration, launch and propagation — as shown in Figure 1.

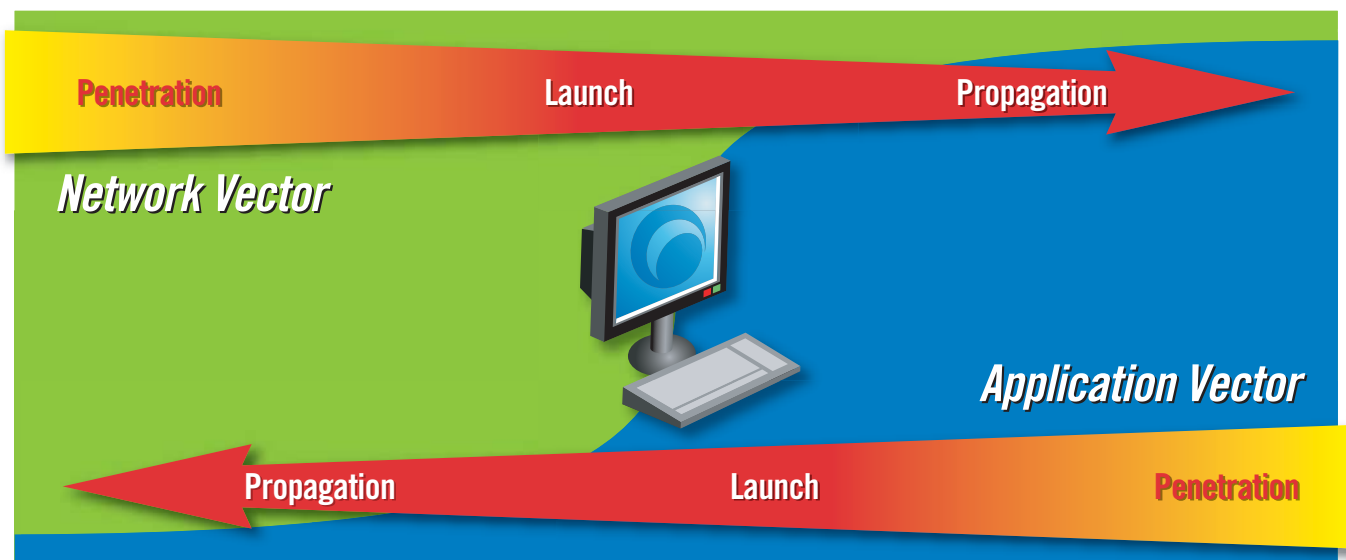


Figure 1: Phases of an Attack on the Host

Penetration	The compromise of a host which allows further malicious activity to occur. Penetration can occur through e-mail, Web browsers, remote buffer overflow or various other methods.
Launch	The execution of the attack’s malicious payload. Launch methods can range from a user double-click to remote memory buffer overflow.
Propagation	Post-compromise activity intended to replicate, retrieve other components, transmit data or enable remote control.

Table 1: Definitions of Host Attack Phases

¹ICSA Labs Virus Prevalence Survey 2003

Protecting hosts from threats used to be much simpler. Because hosts are now so interconnected, they have become susceptible to many more types of attacks that threaten real-time business.

Attacks target host systems using one of two major threat vectors: the network vector and the application vector, as illustrated in Figure 2. Similar to the spread of disease in biological pathology, attacks are carried by vectors to their targets.



Figure 2: Threat Vectors Used in Recent Attacks

The Network Threat Vector

Network-based attacks utilize malicious network traffic to remotely compromise their target systems. Unlike other threats, network-based attacks can penetrate, launch and propagate *without human intervention*. Network-based attacks on the host predominantly exploit vulnerabilities in protocols and network-aware processes. These vulnerabilities are typically the result of programming errors which provide opportunities for a *buffer overflow*. Exploit types include, but are not limited to: direct hacking and theft, network-based worms, denial of service (DoS) attacks, and the installation of remote access backdoors and robot (bot) footholds for future use by the hacker.

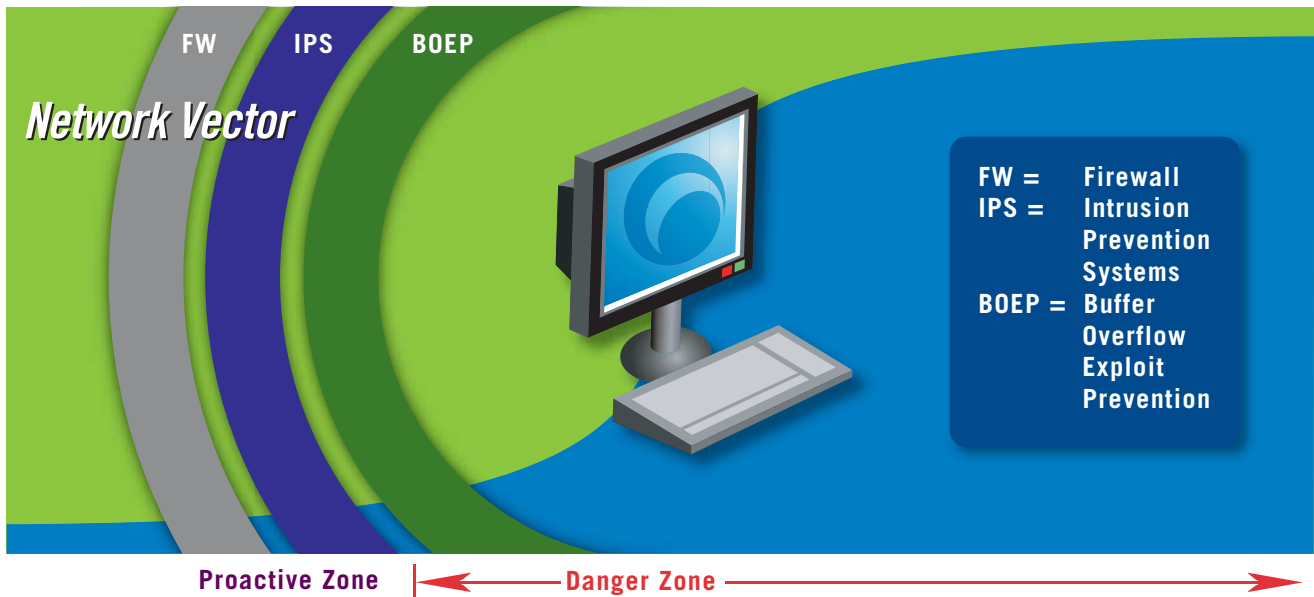


Figure 3: Protection Technology for Network-based Attacks

Example: The Sasser worm infects hosts *from the network* by exploiting vulnerabilities in the Microsoft Local Security Authority Subsystem Service (LSASS)². After penetration of the host via exploitation of the LSASS vulnerability, the worm's payload is transferred to the compromised system and launched remotely, *without user involvement*. Once a system is infected, the malicious executable associated with the Sasser worm intends to propagate by scanning other network hosts for the vulnerable LSASS service.

The Application Threat Vector

Application-based attacks utilize executable files to attack and compromise target systems. Unlike network-based attacks, executable files typically require some form of user involvement to launch an attack. E-mail, Web downloads, removable media and peer-to-peer applications are all common sources of application-based attacks. Attack types include, but are not limited to: viruses, e-mail worms, Trojan horse applications and spyware.

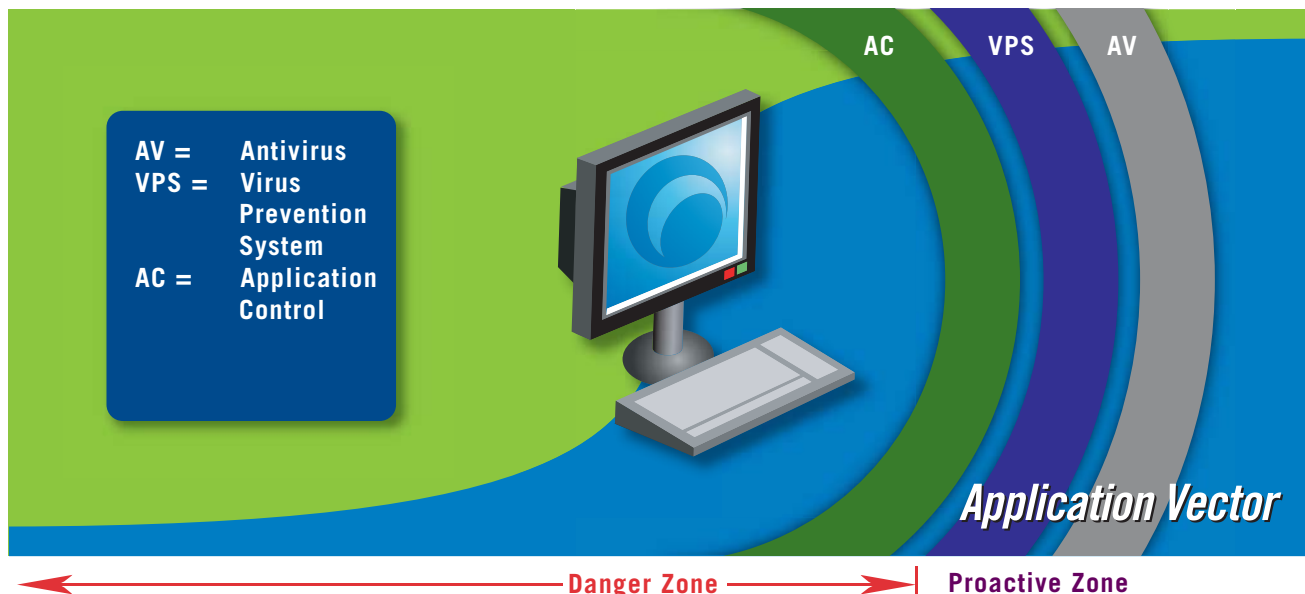


Figure 4: Protection Technology for Application-based Attacks

Example: MyDoom worm variants are sent to users via e-mail, and contain a semi-randomly-named executable attachment. Upon *user execution* of the attachment, the worm launches, installs itself onto the system and begins propagating via e-mail to other e-mail addresses found on the compromised host. Some variants of the MyDoom worm launch distributed denial of service (DDoS) attacks against specific targets. The first MyDoom worm targeted Microsoft Windows Update. The MyDoom.M variant inadvertently affected the popular Google search engine in an attempt to find e-mail addresses by using search queries.

The market's failure to distinguish between the two primary threat vectors has resulted in confusion over which technologies can most effectively *prevent* a particular attack on the host — the focus of the next section.

Examining Host Protection Technologies

While there are several protection technologies available to defend hosts against threats (See Figure 5 on the next page), including personal firewalls and antivirus systems, no single approach provides a comprehensive silver bullet for host security. Each protection technology possesses inherent strengths and weaknesses based on its architecture and implementation. Additionally, as protection technologies have evolved, so have the threats. Many protection technologies now find themselves the target of attacks and evasion techniques.

²<http://xforce.iss.net/xforce/alerts/id/172>

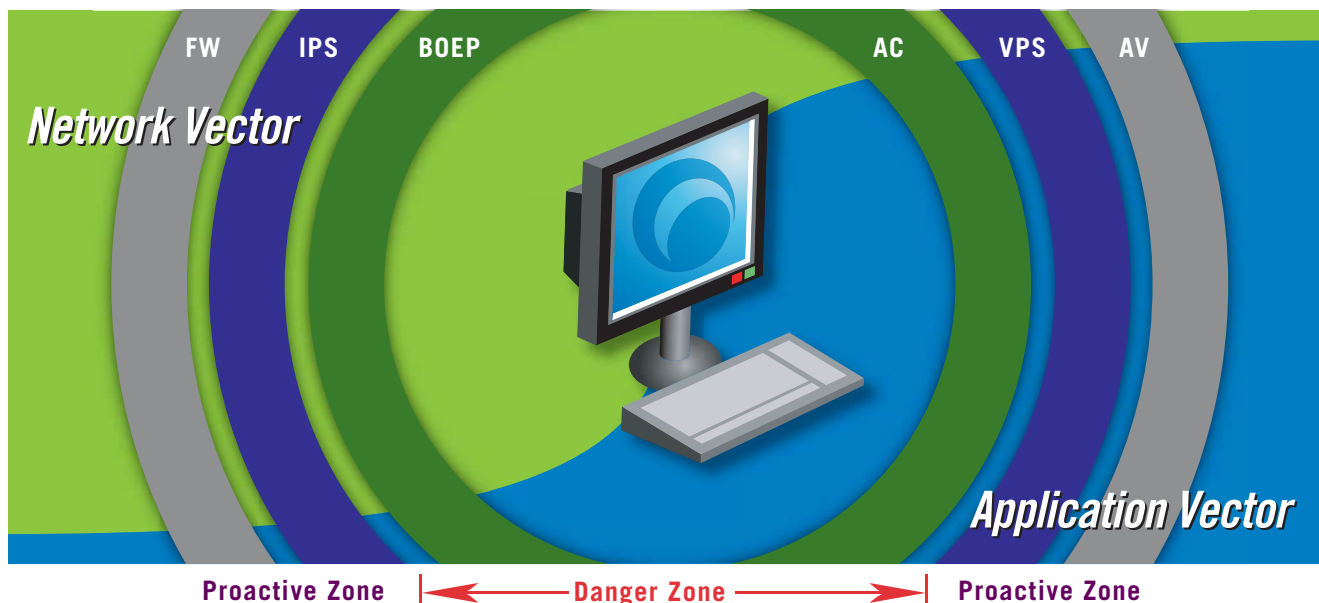


Figure 5: Host Protection Technologies as They Relate to Attack Vectors

Network Attack Prevention

Three main technologies work to defend host systems against network-based attacks, including personal firewalls, intrusion prevention systems and buffer overflow exploit prevention. A subset of network-based attacks can utilize file executables to further propagate from the host. In such cases, application-based prevention technologies may provide detection post-launch and prevent attack propagation.

Personal Firewalls

Personal firewalls (PFW) represent first-generation technology sometimes known as *distributed firewall* technology or *managed personal firewall* technology. Personal firewalls are the most commonly understood and deployed form of host protection, and defend against attacks using network threat vectors in the pre-launch phase before they affect the system. Through overall security policy choices, a personal firewall can *reduce*, but not *eliminate*, risk exposure introduced by interneting hosts. By blocking access to ports, single IP addresses or ranges of IP addresses, protocols and services not needed for legitimate business, personal firewall technology can prevent attacks targeting those resources.

Example: If your business security policy prohibits employees from using FTP (File Transfer Protocol), you could implement firewall rules by port or protocol to block FTP traffic. This type of business policy decision would make the host system immune to network attacks which targeted FTP. However, allowing FTP traffic for legitimate business use would leave the host open to attacks.

Analogy: Using a personal firewall is like locking down the doors and windows of a building for physical security purposes. From time to time, people and information still need to get in and out for legitimate business purposes. When the doors and windows are unlocked for allowable business communication, the openings can't help allowing some bad in with the good. Additional technologies must therefore inspect all traffic permitted by the firewall to ensure that malicious attacks don't slip past and infect the host.

Although personal firewall technology is the most mature and feature-rich of the network protection layers, its value against evolving threats is rapidly diminishing. As observed in 2003 and 2004, attacks now target core network services and applications which cannot be protected via firewall rules without a subsequent disruption to business connectivity and mission-critical applications like e-mail and Web services.

Intrusion Detection and Prevention Systems

Intrusion detection systems (IDS) provide deep packet inspection capabilities that examine the traffic allowed through by personal firewall rules and alert the user to an attack on the host system. Host IDS technology quickly evolved into second-generation *intrusion prevention systems (IPS)* through in-line TCP/IP driver-blocking capabilities that actually block attacks before they can penetrate the system. IPS technology is more advanced than personal firewalls, with the ability to identify *good* traffic from *malicious* traffic in real time, and the capability to block any attacks that may elude the firewall — ensuring business continuity for critical host resources.

Example: The Microsoft Remote Procedure Call (MS-RPC) is a service that must be accessible on most networked hosts. It is used by Outlook Exchange and several third-party and custom-built applications. For this reason, attacks targeting MS-RPC cannot be stopped with personal firewall technology without blocking access to the service. By performing decodes and protocol analysis on the MS-RPC service, IPS technology identifies and blocks traffic that would have exploited a vulnerability in the service. The MS-Blaster worm, a well-known threat using MS-RPC, can be effectively mitigated with IPS technology. Although blocking TCP port 135 at the perimeter firewall will protect networks to some degree, attacks originating inside the network from sources like roaming laptops can easily bypass perimeter firewalls. An IPS technology assures uptime for business services while defending against attacks.

Specific IPS techniques vary greatly, but are generally categorized into either *signature-based* methods or *protocol analysis-based* methods. Signature-based techniques are effective at stopping known exploits, but are often too reactive. Sophisticated IPS offerings combine protocol recognition and analysis techniques to check for *any* exploit of a *known vulnerability*. Most of the well-known attacks released in 2003 were stopped by preemptive IPS technology employing a combination of prevention techniques. As the window of time decreases between vulnerability disclosure and the release of rapidly propagating, highly infectious worms, signature-based IPS techniques tend to be of less value. In contrast, vulnerability-based protocol analysis proves very effective against modern, fast-moving attacks, and since it's based on shielding vulnerabilities, often provides protection even before attacks are released. Preemptive protection from an IPS also requires a multi-layered approach consisting of performance, protection and research. In real-world applications, a multi-method IPS engine is required to balance accuracy for known exploits and vulnerabilities.³

Analogy: An IPS is like an airport security checkpoint. A variety of technologies look for multiple types of threats, including checking bags and people for weapons and chemical residues, and utilizing facial recognition software to identify wanted individuals. Still, to prevent all attacks, you need some idea of what to look for.

Even vulnerability-based IPS technology cannot always prevent the exploit of *unknown vulnerabilities*, defined as previously undisclosed weak spots in software/systems that can become the target of attack. Layered use of personal firewall and IPS technology reduces exposure to attacks on unknown vulnerabilities, but an additional layer of protection helps consistently prevent attacks targeting unknown vulnerabilities on the host.

Buffer Overflow Exploit Prevention

One of the newest host protection technologies available is *buffer overflow exploit prevention*, also known as *memory protection*. This “last line of defense” technology protects hosts from buffer overflow attacks against known and unknown vulnerabilities. According to the National Institute of Standards and Technology, 24 percent of all attacks in 2003 were intended to exploit buffer overflows.⁴ From 1997-99, around 55 percent of CERT advisories involved buffer overflows. As a high-level rule, code should never be executed from writable areas of system memory. By watching the use of Stack and Heap system memory, buffer overflow exploit prevention identifies if a buffer overflow has succeeded and attempts to thwart its executable payload.

Example: Hackers with political or financial motives rarely disclose their techniques until they are no longer of personal use. The vulnerabilities they discover often go undisclosed for a significant amount of time, introducing unknown risk exposure for organizations. Although sophisticated IPS technology offers some protection for unknown vulnerabilities, it does not eliminate risks entirely. Buffer overflow exploit prevention reinforces good application execution and generally prevents unknown buffer overflow exploitation.

³Please reference “Defining the Rules of Preemptive Protection, The ISS Intrusion Prevention System” whitepaper for more information on various IPS prevention methods.

⁴<http://icat.nist.gov/icat.cfm?function=statistics>

Although sometimes marketed as providing more protection, buffer overflow exploit prevention operates in the post-penetration space and is not an ideal way to prevent an attack as it allows some degree of collateral damage. Since the overflow itself succeeds, the targeted process memory can be corrupted and result in a denial of service (DoS). To recover from the compromise, the targeted application may require restarting. If the compromise affects a core service like MS-RPC or LSASS, recovery requires rebooting the operating system. In the case of Microsoft Windows XP-based operating systems, the reboot will occur automatically, resulting in a denial of service.

Analogy: Buffer overflow exploit prevention is like setting a circuit breaker. It stops the house from burning down, but will knock out the power. When the malicious code tries to execute on the host, it trips the switch to protect the rest of the system, but causes an interruption of service to that area. Thwarting the attack in live execution space incurs collateral damage to the targeted service or application. Until the root cause or vulnerability is addressed, the circuit breaker will trip often.

Buffer overflow exploit prevention technology should be viewed as a last line of defense in a layered approach that includes personal firewall and IPS. A personal firewall will block known and unknown attacks against the ports and services you don't need. IPS will filter out known and unknown attacks against *known* vulnerabilities. At a cost, buffer overflow exploit prevention will provide the necessary insurance for overflows against unknown vulnerabilities.

Application Attack Prevention

There are three main technologies available to defend against application-based attacks — antivirus (AV), virus prevention systems (VPS) and application control (AC). A subset of application-based attacks can utilize the network in the course of the attack. In such cases, the network attack prevention technologies may provide detection post-launch and prevent propagation.

Antivirus:

The most common, first-generation protection technique for defending hosts against the application threat vectors is *antivirus* technology. Antivirus technology mitigates *known* threats in the pre-launch phase of the attack's lifecycle. Despite the technology's maturity, AV has been unable to escape its inherent, reactive, signature-based nature. According to the 2003 Virus Prevalence Survey from ICSA Labs, approximately 2.7 million outbreaks occurred per month despite nearly 100 percent desktop AV coverage.⁵

Example: When the first Netsky variant appeared in the wild and a sample was harvested, it took a day or so for AV vendors to develop a virus signature for their products. The customers that received the Netsky.A signature had to immediately update, test and distribute it thoroughly throughout their network to remain protected from the outbreak. However, when the next Netsky variant is unleashed, all systems will be prone to infection and the AV signature update cycle will occur all over again.

Traditional signature-based AV products are extremely effective at detection and prevention of *known* viruses, worms and some Trojans. An AV product will block nearly everything *tomorrow*, but will catch almost nothing today. According to AV-Test.org, the top two AV vendors typically take more than 25 hours to respond to day-zero threats.⁶

⁵ ICSA Labs 9th Annual Computer Virus Prevalence Survey

⁶ <http://www.avtest.org/>

Average Vendor Response Time

Data Source: Datamation and AV-Test.org
http://itmanagement.earthweb.com/columns/executive_tech/article.php/3316511

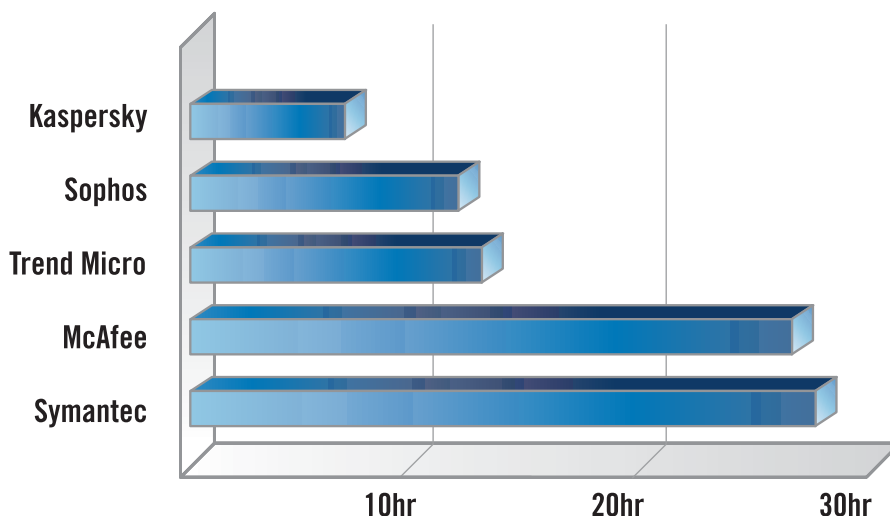


Figure 6: Average Antivirus Vendor Response Time

Analogy: Traditional AV is like a most wanted list or a criminal database that identifies known offenders, but cannot single out new ones until they've been captured, fingerprinted and photographed — and the information shared far and wide. Even superficial variations in known viruses make it possible to evade antivirus systems.

AV remains effective against known viruses and worms, but is almost powerless to stop day-zero attacks. Preemptive protection involves additional technologies to stop unknown threats.

Virus Prevention Systems:

Virus prevention systems (VPS) address day-zero, application-based threats which evade traditional antivirus technology. AV technology has proven too reactive to effectively stop day-zero, file-based attacks. Next-generation VPS technology complements traditional AV by using pre-execution behavioral analysis to detect and block malicious code. With a focus on execution behavior, the VPS preemptively detects and prevents entire families of malicious code based on what they do. In addition, the VPS offers day-zero virus prevention without updates.

Rather than writing a signature for a single strain of a virus or worm, a behavioral system like VPS examines the activities of an executable file and detects whole families of malicious code based on code actions or techniques. Although there are new malicious viruses written all the time, virus authors use a limited number of techniques to create viruses and these techniques exhibit common unhealthy behaviors. Once the VPS knows a technique, it will always detect it. For example, healthy code doesn't self-replicate. If VPS technology identifies a known technique for replication, it will then prevent the attack even if it is previously unknown.

Example: Whereas MyDoom.A, MyDoom.M, and MyDoom.XYZ may be different enough to evade AV signatures, the techniques they employ are all common to tens, hundreds or even thousands of other previously written viruses or worms. By targeting what the code does, new and unknown malicious code can be prevented by the virus prevention system — day-zero before the code is ever unleashed.

Bagle Variants and Response times

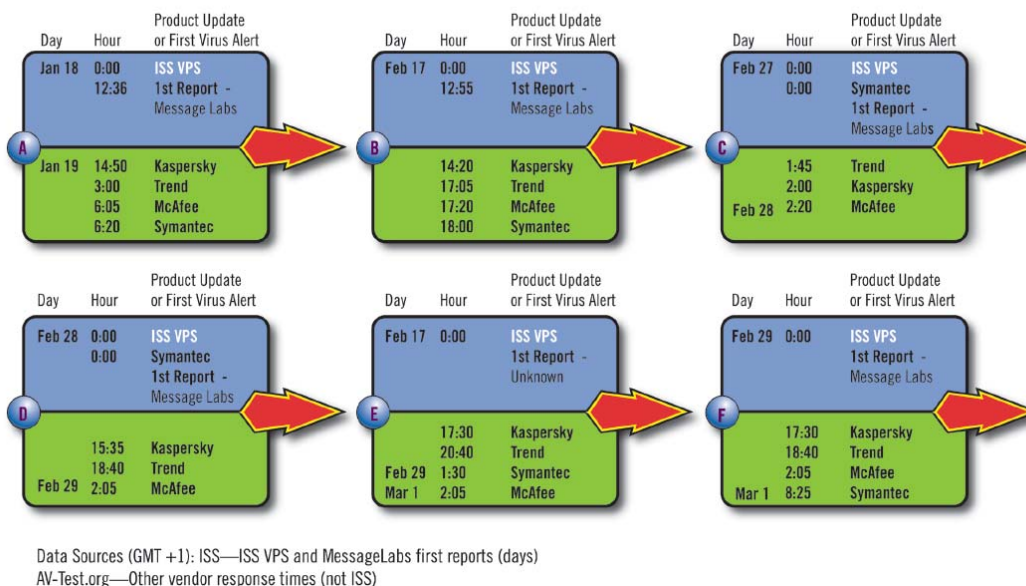


Figure 7: The Virus Prevention System's Protection from Bagle Variants A-F Compared with Traditional Antivirus Response Times

To prevent an attack from occurring, inspection of an executable's behavior must occur *outside* of live execution space. The term "sandboxing" refers to inspecting code during its execution on a host. Early sandboxing technologies could detect unknown viruses, but produced unacceptable false-positive rates. Preliminary sandboxing technologies also tried to detect attacks in live execution space. Allowing code to run in live executable space on the host introduces risk to the system. For sandboxing technology to detect attacks, it often allows them to execute long enough for some form of system alteration to occur.

Virus prevention systems have improved upon the intent of sandboxing technology. VPS technology detects attacks before execution in live space by running the code in a virtual environment to examine its entire execution safely. Then the VPS decides whether to allow the executable to continue to live execution space.

Analogy: The virus prevention system running code in pre-execution space is almost like working in a consequence-free virtual reality. Sci-fi TV show *Star Trek: The Next Generation* illustrated the concept of virtual reality with the Holodeck, and the film *The Matrix* also explored the idea of a virtual world. In both productions, characters interact in a realistic environment that simply doesn't exist. Scenarios are played out, but nothing actually occurs and there are no consequences. VPS works much the same way, allowing code to execute in a virtual environment to understand its intent without placing the host system in any real danger. The code believes it is running on the real system, and observing its actions in the simulated reality enables the VPS to identify and avoid anything which would have harmed the real system.

In testing conducted within Internet Security Systems (ISS) by the X-Force® security intelligence team, VPS technology has proven to detect and prevent approximately 98 percent of unknown and known viruses, worms, and other malware and is a powerful component of a multi-layered host protection strategy.

Application Control:

Application control rounds out the protection technologies designed to prevent application-based attacks on the host. Application control technology protects hosts from threats before or during the launch phase of an attack. Much like personal firewall technology, application control reduces a host's attack surface through policy decisions and static rules. Acting as another last-line defense, application control is a useful protection technology operating in the execution space that mitigates remaining threats after other protection methods have been employed and exhausted.

Example: For static environments, an administrator could create very restrictive application control rules designating which applications can run and what they can do while they run. If a user were to unknowingly try to execute MyDoom.exe, then the restrictive rules could prohibit it. However, each rule that tightens security may also prohibit new or modified healthy application use from taking place. In typical, dynamic host environments, a rigorous change management process is often necessary when utilizing this technology.

As with all of the host protection technologies, implementations of application control vary. Common to all implementations, however, is the high management cost associated with application control. Operating system updates and patches often require testing and tuning before deployment to ensure application control policy doesn't restrict legitimate applications from running. *Baselining*, — defined as allowing all applications currently on a system at the set point to execute — can reduce, but not eliminate the management cost. Baselining also carries the risk of allowing a previously unnoticed, but malicious, application to continue running. Therefore, baselining is most appropriate for systems requiring few updates such as ATMs, cash registers or other static systems where repeating the baselining process is uncommon.

Newer application control technology implementations offer rule-based behavioral blocking. Through the use of Windows application programming interface (API) shims and application access control lists (ACLs), application control can thwart some new and unknown day-zero attacks. However, this approach is plagued by the same false-positive issues that hurt early sandboxing technology. Rules-based, behavioral application control technology allows profiling and customization of individual rules, but the practice involves a high cost of setup and ownership and ultimately forces trade-offs between protection and legitimate business use.

Another promising area of application control technology involves *application compliance*. Application compliance provides the ability to enforce corporate policy on AV compliance, operating system patch level and restricted applications (such as peer-to-peer applications) before network access is granted. Enforcing compliance improves the overall health of the system, and in turn, the health of the entire computer network.

Analogy: Application control is like child-proofing a house. You can block off rooms and lock cabinets, but that is also invasive to adults. You can hide or remove all of your dangerous items and tools, but that too restricts appropriate use of those tools. As the child grows and changes, so too must you modify and adapt your preventative measures.

In many ways application control technology is similar to personal firewall technology, based on the ability to realize some protection from threats by using acceptable policy decisions. The more that AV and VPS can be leveraged, the less necessary it is to create and maintain strict or fragile application control rules. Application control can be a useful technology in a multi-layered host protection strategy but is never recommended as the sole method of protecting a dynamic host system.

Summary:

Organizations must understand the nature of today's wide variety of Internet threats in order to apply preemptive protection that stops threats *before* they impact servers and desktops. Recognizing the two major attack vectors targeting host systems makes it easier to understand how different technologies protect hosts from various forms of attack. It is also crucial to understand what each protection technology can and cannot do. Technologies which *prevent* attacks in the early phases of their life-cycle result in significantly less negative impact than those which protect after the penetration or launch of an attack.

As each host protection technology possesses strengths and weaknesses, selecting just one technology for comprehensive host protection results in too much risk to the host environment. Any single technology represents a singular point of failure. Employing the different protection technologies in concert brings risk exposure to threats down to acceptable levels. In addition, combining multiple host protection technologies into a single host protection solution significantly reduces management costs.

A complementary, multi-layered host protection approach remains superior to any single technology offering. Where one protection technology may be weak, another can be strong, offering several lines of defense and greater overall host protection. When developing a host protection strategy, only a comprehensive solution can keep you ahead of the next threat.

About the Author

Joshua Corman serves as technical product manager of host solutions for Internet Security Systems, Inc. (ISS). With more than eight years of experience in security and networking software development, Corman is responsible for the technical vision and direction of host protection solutions and is regarded as the expert on technical subject matter for ISS' host protection products. Corman is also charged with competitive analysis and field support resource development.

Corman was the product manager at vCIS Technology, Inc., when ISS acquired the company in 2002 for its preemptive behavioral inspection technology. Before joining vCIS, Corman worked with the SPECTRUM Network Management Platform for Cabletron Systems. He served as software QA manager and Computer-Human Interactions – user centered design specialist.

Corman received a BA in Philosophy, Phi Beta Kappa, Summa Cum Laude, from the University of New Hampshire.

About Internet Security Systems

Internet Security Systems, Inc. (ISS), the trusted expert to global enterprises and world governments, provides products and services that protect against Internet threats. An established world leader in security since 1994, Internet Security Systems delivers proven cost efficiencies and reduces regulatory and business risk across the enterprise for more than 11,000 customers worldwide. Internet Security Systems is traded publicly on the Nasdaq (ISSX), and is one of the most widely recognized and valued information security brands in the world.

GLOBAL HEADQUARTERS

6303 Barfield Road
Atlanta, GA 30328
United States
Phone: (404) 236-2600
e-mail: sales@iss.net

REGIONAL HEADQUARTERS

Australia and New Zealand

Internet Security Systems Pty Ltd.
Level 6, 15 Astor Terrace
Spring Hill Queensland 4000
Australia
Phone: +61 (0)7 3838 1555
Fax: +61 (0)7 3832 4756
e-mail: aus-info@iss.net

Asia Pacific

Internet Security Systems K. K.
JR Tokyu Meguro Bldg. 3-1-1
Kami-Osaki, Shinagawa-ku
Tokyo 141-0021
Japan
Phone: +81 (3) 5740-4050
Fax: +81 (3) 5487-0711
e-mail: jp-sales@iss.net

Europe, Middle East and Africa

Ringlaan 39 bus 5
1853 Strombeek-Bever
Belgium
Phone: +32 (2) 479 67 97
Fax: +32 (2) 479 75 18
e-mail: isseur@iss.net

Latin America

6303 Barfield Road
Atlanta, GA 30328
United States
Phone: (404) 236-2709
Fax: (509) 756-5406
e-mail: isslatam@iss.net

Copyright© 2004 Internet Security Systems, Inc. All rights reserved worldwide.

Internet Security Systems, Proventia and SiteProtector are trademarks, and the Internet Security Systems logo and X-Force registered trademarks, of Internet Security Systems, Inc. Other marks and trade names mentioned are the property of their owners, as indicated. All marks are the property of their respective owner and used in an editorial context without intent of infringement. Specifications and content are subject to change without notice.

MC-HIPS-WP-904