

# iSolation Server 1.0

*Stopping new and unknown viruses from entering the network*



*June 2004*

**avinto**The logo for Avinto, consisting of the word "avinto" in a bold, lowercase, sans-serif font. The final letter 'o' is replaced by a stylized icon: a black circle containing a white 'i' shape, with a small orange circle above it.

## Disclaimer

This document is provided 'as is' without any express or implied warranty relating to the sale and/or use of Avinti products. While all information in this document is believed to be correct as of the time of publication, this document is for reference purposes only and may not reflect the actual state or progress of the products described. All brand names and product names are trade names or trademarks of their respective holders. Avinti, Inc. makes no implication of association with other vendors or products mentioned in this document.

## Trademarks

Avinti and iSolation Server are trademarks of Avinti, Inc. in the United States and/or other countries. All other companies and products mentioned herein are recognized as the continuing owners of their respective trademarks.

© 2003, Avinti, Inc.

# Table of Contents

<b>Overview</b>	A Long Time Problem	1
	E-mail: The Number One Vector for New Viruses	2
	iSolation Server stops new viruses from entering the network	3
<b>The Problem</b>	Overview	4
	E-mail—Open Channel into Otherwise Secure Networks	5
	Virus Signature Systems—Window of Vulnerability	6
	Server or Client Software?	7
	Scalability	8
	False Positives, False Negatives	9
	Productivity	10
<b>The Solution</b>	How iSolation Server solves the problem of unknown viruses	11
	How iSolation Server Works	12
	Key Internal Components	13
	System Performance, Throughput, and Latency	15
<b>Conclusion</b>		18
	<b>About Avinti</b>	19

# Overview

## *A Long-Time Problem*

Traditionally, antivirus software has been a reactive system—only after a new virus emerges is a pattern made that will allow the antivirus system to detect the *next* occurrence of the same virus. New and unknown viruses are not detected by pattern-matching antivirus systems because patterns do not exist for them yet. Since modern viruses are very fast moving, worldwide propagation only takes a few minutes. By the time that new patterns are available and installed, a new virus will have already infiltrated and infected many networks. At the very least, new and unknown viruses cause increased network traffic that creates annoying bandwidth concerns, but they can also be programmed to carry extremely destructive and malicious payloads that can cause irreparable harm to user data and network systems.

Pattern-matching antivirus software has an acute window of vulnerability problem: the process of discovering a new virus, studying how the virus works and how to detect it, creating a new detection pattern for the virus, making the new pattern generally available, and then finally installing the on-site update takes time—a long time. During this window of vulnerability (which in the best case can be just a few hours, but is usually measured in days), network users and systems are open to attack from fast-moving outbreak viruses. Just recently, variants of the MyDoom, Bagle, and Netsky viruses carried on e-mail were being created and released into the wild multiple times a day. No pattern-matching system could keep up with the frenzy of defending against these viruses. The impact caused by these viruses has been extreme.

In order to minimize this window of vulnerability and keep viruses out of their networks, many system administrators spend considerable effort in constantly updating their virus signature files. Some administrators try to keep viruses from their networks by stripping all e-mail attachments and by blocking certain file types from entering the network, resulting in user complaints due to decreased work productivity and inability to use e-mail. Blocking executables and stripping attachments from e-mail encourages network users to circumvent these practices by obtaining such executables through non-standard methods (like creating user accounts on public e-mail servers to get around the standard e-mail system, or copying software from removable media). Such practices only worsen the virus problem by allowing potentially harmful executables into the network through non-standard means.

Simply blocking attachments or executables based on their file extension or other attributes is also an ineffective method to prevent viruses from entering a network. Virus writers can simply change file names or employ social engineering techniques to coerce users into inadvertently running malicious code.

Traditional virus pattern scanning software is very good at detecting known viruses and cleaning infected systems that have already been attacked. The task remains, however, to prevent new and unknown viruses from entering a network via e-mail while still allowing users full access to productive tools and e-mail attachments available through the full use of e-mail systems.

## *E-mail: The Number One Vector for New Viruses*

E-mail continues to be the greatest source for network viral infections. ICSA Labs' latest findings indicate e-mail attachments account for 86% of all viral infections.

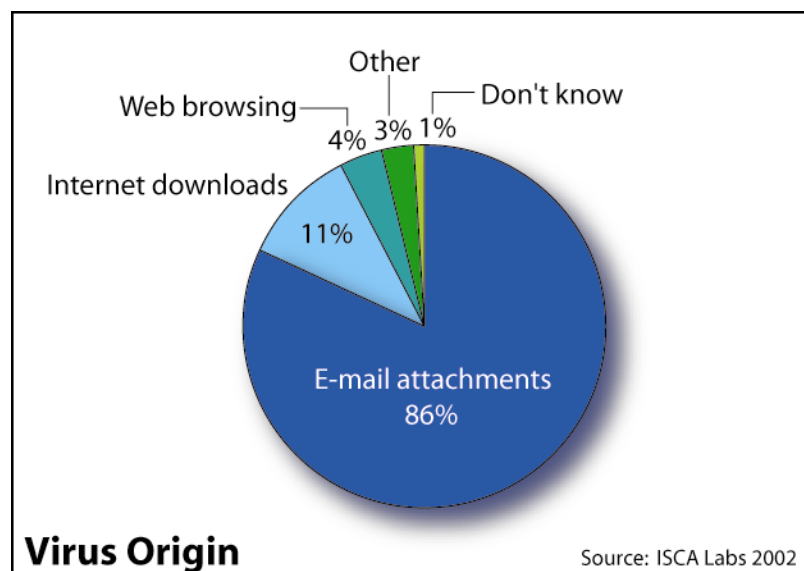


Figure 1. Virus origin. (Source: ICSA Labs 2002)

Organizations depend on e-mail as a mission-critical application to enhance the productivity of employees by improving communications with customers, co-workers, vendors, and suppliers. E-mail enables Web-based customer support, allows customers to obtain critical updates and new products almost instantaneously, and substantially improves business goals and productivity. Using e-mail, employees share productivity tools and get instant access to vital information. Without this information, business suffers.

Because e-mail is so heavily used, virus writers use it as a favorite medium to introduce viruses into organizations. Knowing this, some system administrators strip attachments from e-mail or block executables from entering the system. Others attempt to strip attachments based on their file name extension (.EXE or .PIF, for example). These brute-force methods may be helpful in preventing some viruses, but most often they are counter-productive and easily circumvented. For example, outside users can easily rename \*.EXE and \*.SCR files to \*.EX1 or \*.SC1, bypassing the system administrator's filter. In addition, e-mail may be formatted using HTML or have other active content that can be potentially harmful or carry new viruses into the network. Using social engineering techniques, virus writers have cleverly disguised their destructive viruses and duped many users into inadvertently opening attachments and executing viruses in their e-mail. Some e-mail messages can even carry and propagate executable payloads without an attachment. The proliferation of new viruses by the use of e-mail has caused some system administrators to severely restrict or even prohibit the use of e-mail in an effort to stop new viruses.

## *iSolation Server stops new viruses from entering the network*

Avinti's iSolation Server™ is a system that prevents e-mail containing executables that exhibit viral behavior from entering the network—including new and unknown viruses. Designed to work with any SMTP-based e-mail server, iSolation Server functions as an SMTP gateway; intercepts e-mail at the network edge; executes and monitors the e-mail and related messages in a separate, isolated environment; and quarantines any e-mail that exhibits malicious behavior. E-mail that does not violate set policies is forwarded to the e-mail server. Using iSolation Server, system administrators can allow the full use of e-mail, including attachments, executables, and active content without worrying that the e-mail system is being used to transport malicious viruses into their network.

Using Simple Mail Transport Protocol (SMTP) interception techniques, iSolation Server intercepts e-mail before it is sent to the e-mail server (Exchange, for example) and detects whether or not there are any executable or 'active' components present in the e-mail body and/or attachments. Even before executable e-mail is sent to the execution module that is isolated from the network, a series of filters are used to determine e-mail viability and whether or not the particular e-mail message should be directly forwarded to the e-mail server, deleted, or sent to the execution module for observation. Each execution module is a virtual representation of the intended recipient's environment, and is used to execute the e-mail and all attachments. Execution monitors built into the virtual execution modules enable iSolation Server to watch and record all execution events. As the e-mail and its associated attachments execute, the execution monitors observe what actions the program takes, and record any resultant effect. E-mail exhibiting viral or malicious intent is quarantined, while clean e-mail and attachments are forwarded on to the regular e-mail system where it is delivered to the intended recipients.

iSolation Server does not replace traditional virus scanning but rather works in conjunction with virus scanners to provide an advanced level of security and protection. Traditional virus scanners, by nature, are designed to detect known viruses. The combination of traditional virus scanning technology with iSolation Server, however, provides layered protection and enhanced security for the network.

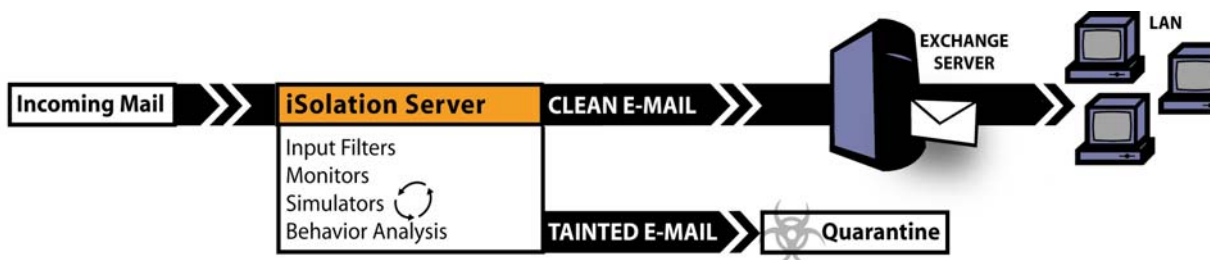


Figure 2. iSolation Server Overview

# The Problem

## Overview

Relying on traditional virus scanning technology as the sole source of network protection from viruses poses two issues:

**False Positives.** Because the number of viruses is increasing (and consequently the number of virus signature patterns), the potential for false positives reported in scanning for known virus signatures also increases. These factors also add to the time and processing power required to complete virus scans.

**False Negatives.** More importantly, the bigger problem with traditional pattern matching antivirus techniques is false negatives: new or unknown viruses will not match up with any of the signatures in the database file, allowing these viruses to enter the network where they can cause much havoc and destruction.

Virus writers often make enhancements and modifications to existing viruses, altering their appearance and changing their byte patterns to elude virus scanning software. In August 2003, for example, a variant of the SoBig e-mail virus flooded the Internet and damaged e-mail systems worldwide.

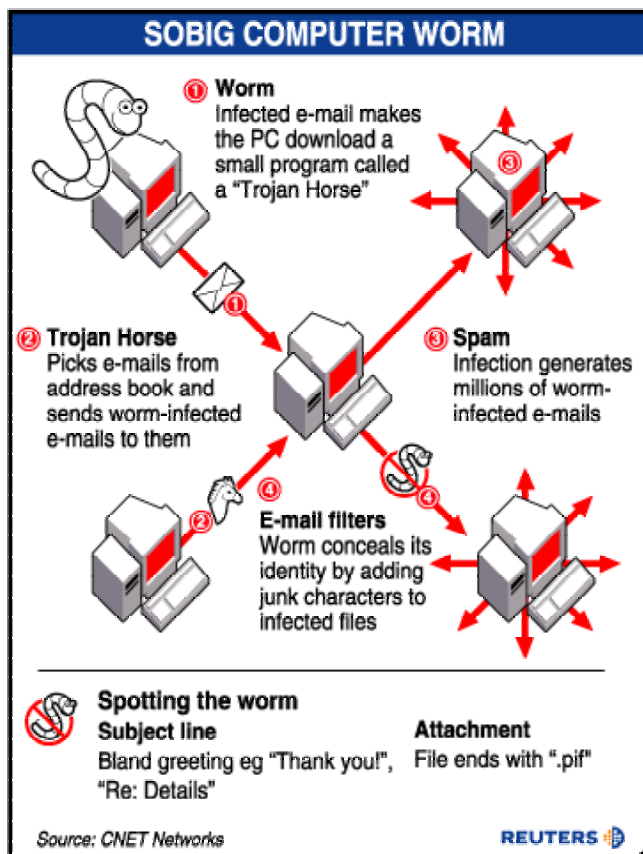


Figure 3: SoBig Worm Attack Method

Antivirus software did not detect the SoBig.F virus because it was a variant of an earlier version; virus signatures in existing database files could not detect the new variant. Although major antivirus software companies developed a new signature for SoBig.F in near record time, the fast-moving virus infected millions of PCs worldwide, causing massive Internet and e-mail outages.

More recently, in February 2004 more than 50 variants of two new e-mail viruses (Netsky and Bagle) were written and released into the wild causing huge disruption to services and damage across the entire Internet.

Instead of relying on signatures of known viruses, Avinti's iSolation Server catches new viruses carried by e-mail by monitoring the behavior of the virus *before* it is allowed into the network. iSolation Server detects behaviors like those exhibited by SoBig, MyDoom, Bagle, Netsky, Mimail and other viruses and traps the virus before it can damage the network.

### ***E-mail—Open Channel into Otherwise Secure Networks***

E-mail has become one of the most widely used systems in the modern business world. Everyone depends on its use—and everyone is affected by its abuse. Modern networks depend on firewalls and other intrusion prevention devices to prevent unauthorized access and entry. If used correctly, these devices are highly effective at preventing various types of malicious attacks on networks. However, even within otherwise secure networks, e-mail provides an entry point for attackers that depend on social engineering techniques and unsuspecting e-mail users to launch destructive exploits. Since e-mail is a mission-critical application for many businesses and most e-mail users are non-technical, unscrupulous attackers use it to transmit their criminal activities.

Because e-mail is the leading carrier of viral threats into networks, many system administrators limit either (1) e-mail use among employees or (2) the number or types of attachments that are permitted into the e-mail system. Limiting employee use of e-mail not only stifles business productivity, it actually creates a backdoor environment wherein employees can circumvent the standard e-mail system by opening accounts on public e-mail systems or by bringing potentially harmful executables in using personal media.

## *Virus Signature Systems—Window of Vulnerability*

Since traditional virus scanning systems rely on a database of known virus signatures, they cannot detect new or unknown viruses. Network and security administrators must constantly verify that virus signature files are updated with the latest patterns in order to protect their systems. New viruses, however, have no matching pattern in signature databases because they have not yet been detected. Such viruses are called 'outbreak' or 'day-zero' viruses because they infiltrate and infect networks immediately—well before new pattern definitions are available. Outbreak viruses use their anonymity to replicate rapidly and infect hundreds of thousands of machines.

In order for antivirus software companies to develop new virus patterns, new viruses must first be contained and studied. Although multiple 'honeypot' systems currently attract new viruses and contain them for study, many organizations still catch these same viruses *after* they have infected their systems and later send them to antivirus companies for evaluation. After antivirus companies receive a new virus for study, they create a new signature that must be added to the existing database of known viruses in order to trap the new virus the *next* time it hits.

The window of vulnerability in which networks can be infected by outbreak viruses follows this timeline:

- Release of new virus into the wild
- Detection and study of new virus
- Development of new virus signature pattern
- Addition of signature pattern to database of known viruses
- Distribution and installation of new virus signature database in users systems

Even in the best case, this process is measured in terms of days—during which a new outbreak virus runs wild.

When system administrators become aware of new viruses, they often disable outside access to their networks to prevent delivery or use of e-mail until the window of vulnerability has passed. Although network administrators recognize that these methods stifle employee productivity and burden network personnel, they allow them because they provide a better option than dealing with the aftermath of a virus attack.

Even when utilizing pattern-matching antivirus systems that automatically search for and update virus definitions, networks are vulnerable to new outbreak viruses. Popular antivirus companies employ teams that work around-the-clock solely to catch, identify, and develop definitions for new outbreak viruses. Even in the best scenarios, this effort takes time—time wherein outbreak viruses wreak great damage on unprotected networks.

Instead of matching file contents and network data against a database of known virus signatures, iSolation Server monitors executable behavior in an isolated environment, traps malicious or viral executables, and prevents them from entering the network. iSolation Server defends networks from new outbreak viruses carried on e-mail—without a window of vulnerability problem.

## *Server or Client Software?*

Of particular concern to many system administrators is the task of having to install, configure, and maintain software on client machines in the network. Many antivirus and intrusion detection software systems require that software be installed on all client machines to be protected. In many cases, the software requires hands-on installation and configuration for each individual workstation. Individual attention at each workstation creates a complex and time-consuming network management situation for the system administrator. Furthermore, many of these same systems must be un-installed and then re-installed whenever other system software is to be added or modified, thus adding to the network administrator's workload.

Since viruses often deliver their payload on end-user workstations, many vendors of antivirus and security software find that software must be installed on every end-user system in the network. This type of scenario dramatically affects workstation performance and usability, and creates an environment that makes any changes, enhancements, or modifications to the workstation very difficult. iSolation Server eliminates this problem by simulating client workstation environments on a server-level device and enforcing security policies that take place long before executables are ever allowed into the network or onto workstations. No workstation configuration or workstation software needs to be installed with iSolation Server—iSolation Server operates as a server-based solution.

## Scalability

iIsolation Server 1.0 is designed to protect standard SMTP e-mail systems, and currently works with small to medium sized e-mail implementations. iIsolation Server is designed to be highly scalable; it can scale vertically by making use of multiple processors on the same machine, or horizontally by running its processing over multiple physical machines. For functional as well as practical reasons, iIsolation Server requires dedicated hardware that is separate from the e-mail server hardware. In its simplest form, iIsolation Server 1.0 running on a single machine will support small to medium e-mail environments of up to 1000 clients; subsequent versions will scale to support much larger environments by making use of multiple machines. High throughput and heavily used e-mail systems can also be supported well by running iIsolation Server on high performance multiprocessor-based machines. In order to achieve optimal performance, each e-mail server should be paired with a dedicated iIsolation Server.

By running on dedicated hardware, iIsolation Server does not burden the e-mail server with unnecessary processing. The design of iIsolation Server prescribes that the simulation and evaluation engines be executed on separate hardware from the rest of the e-mail system in order to effectively isolate and protect the network from potential malicious attacks.

Several major components comprise iIsolation Server, including the following engines:

- E-mail interception
- E-mail filtering
- Threat potential analysis
- Simulator management
- Simulators
- Analysis
- Output filters



Figure 4. iIsolation Server engines

These components form a comprehensive system that intercepts and analyzes e-mail traffic and sends suspect executable e-mail to an appropriate protected simulator environment where the executable runs in a simulated, virtual environment that mimics a recipient workstation environment. During the simulation, iIsolation Server monitors the execution of the suspect e-mail and compares it to a set of policies. The system determines whether the executable violates set policies, sends policy-violating e-mail to quarantine, and forwards clean e-mail to the Exchange server for transmission to the intended recipient.

### iIsolation Server 1.x

iIsolation Server 1.x runs on a single machine. Higher performance hardware will yield higher throughput and performance benchmarks.

## **iSolation Server 2.x**

Each of iSolation Server's component engines are designed to run on a single machine (separate from the e-mail server) or on multiple processors across multiple machines. The system is multi-threaded and utilizes available multi-processors and high performance hardware to co-exist with and protect small to large networks. iSolation Server's flexible architecture provides optimal scalability and system performance by allowing the major components to run together on the same physical system or optionally on separate physical systems for high performance and higher throughput e-mail systems. This architecture will be implemented in iSolation Server 2.x.

## ***False Positives, False Negatives***

### **False Positives**

A false positive is simply a false alarm. A reported side effect of some behavior based antivirus and intrusion detection system is that they tend to be very prone to false positives. Some error-prone 'sandboxing' systems detect when certain operating system Application Programming Interface (API) routines are called and erroneously report a false positive without verifying exactly what the calling program was doing (good, bad, or indifferent). Instead of allowing the suspect program to proceed after detecting that the monitored API has been called, these sandboxing systems simply halt the program and flag it as being malevolent without actually determining what it does. This type of system is extremely prone to false positives—they declare false positives when in reality there are none.

Conversely, the iSolation Server places suspect code in a protected, isolated environment and allows it to execute. Not only does it monitor the called APIs, it also allows the program to execute and notes all events associated with it. Allowing suspect code to execute lets iSolation Server monitor the behavior of the code. Instead of guessing the future actions of the code, iSolation Server observes precisely what suspect code actually does within the security of a protected environment. Consequently, the system is much more precise in judging what suspect code does. False positives are drastically reduced, while programs that violate set policies are halted and quarantined from the normal e-mail system.

## **False Negatives**

A false negative is a miss. This occurs when the software program fails to detect a virus attack. A false negative can bring grave consequences because it means a malevolent program has bypassed network security and has infiltrated the network. By executing and observing behavior of a suspect program, iSolation Server can precisely determine whether or not executables violate set policies.

Both pattern-matching scanners and sandboxing behavior analysis systems are susceptible to false positives and false negatives. Armed with a database of virus signatures, pattern scanners can incorrectly flag a file as viral if it contains a coincidental byte pattern that matches a known virus signature. However, the greater threat with pattern-scanners is their inherent susceptibility to false negatives because they cannot detect new and unknown viruses – this poses an even greater threat to the network. API-monitoring sandbox systems are more error-prone to false positives because they trap innocent programs that happen to call an API that could be used for malicious purposes.

Monitoring program execution allows for more precise and exact analysis. By allowing suspect executable code to execute in a closely guarded environment, iSolation Server significantly reduces the possibility of false positives and minimizes false negatives. iSolation Server can ascertain exactly what a suspect executable does in the actual system by first monitoring the executable's every instruction and call in the virtual environment.

## ***Productivity***

Organizations depend on e-mail for internal communication, collaboration and scheduling, as well as maintaining relationships and communication with customers. Heavily used for document sharing and file transferal among employees and customers, e-mail is the preferred means of transmitting information of all sorts among the vast majority of businesses worldwide.

Because of the pervasive use of e-mail, virus writers use it more than any other medium to spread malicious code and malware. Through the use of social engineering, unscrupulous hackers get innocent e-mail users to open and execute computer viruses. Recognizing that e-mail is used as a transmission vector for viruses, some businesses have elected to curtail its use; other organizations have either limited or eliminated e-mail attachments in an effort to reduce vulnerability to viruses carried in e-mail. These practices severely hamper employee productivity. Employees, discovering that attachment-enabled e-mail increases their productivity, frequently circumvent restrictive company policies. Denied the full use of e-mail, employees often open public e-mail accounts to surreptitiously send and receive e-mail against company policy, often with attachments and executables that have disastrous consequences for the company network. Allowing the full use of e-mail with iSolation Server monitoring for malicious executables allows high employee productivity while providing a greater level of security.

Halting employee use of e-mail because viruses are spread through e-mail is like asking people to stop driving cars because they cause accidents<sup>1</sup>. Both are counter-productive, unreasonable requests. iSolation Server gives e-mail system administrators peace of mind knowing that executables and attachments are being monitored and checked for policy enforcement before being allowed into the network. Through the prudent use of available technology, iSolation Server allows employees to maximize their work productivity by allowing e-mail attachments into the company network.

---

<sup>1</sup> Jon Swartz, "More workers get shut out of e-mail," *USA Today*, 8 Sept. 2003, sec. B, p. 1

## The Solution—a closer look at iSolation Server

### *How iSolation Server solves the problem of new and unknown viruses*

New and unknown viruses do not match up with any of the signatures in traditional pattern-matching antivirus database files, allowing these viruses to enter the network and cause much havoc and destruction before they are recognized and cleaned up. Besides affecting individual workstations on the network, some new and unknown viruses can severely damage the delicate core systems that run an enterprise network. Rebuilding or restoring individual workstations after a viral attack can be a very time consuming endeavor, and rebuilding core networking systems is a very tedious and intricate task. Instead of having to rebuild and restore corrupted and damaged systems, it would be much better to catch and trap new and unknown viruses before they can enter a network. The best way to do this would be in an environment that was separate and isolated from the rest of the enterprise network so as not to encumber the core network systems with any additional task, and also to keep potential viruses away from the network and the systems connected to it.

Malicious code is easily sent as an attachment to an e-mail, or embedded within the e-mail itself. Destructive programs can be executed in many ways using e-mail, including auto-execution with the e-mail viewer or by an inadvertent click of the mouse. In fact, e-mail is the easiest and most used mechanism to transport and introduce viruses throughout the world today. But since e-mail is so productive, useful, and necessary it would not make sense to limit or stop using it. What is needed is a system that will intercept e-mail and send it to an isolated environment where all executable programs and their consequences can be monitored. In this system, innocent programs and code would be forwarded to the regular e-mail system, while malicious and destructive programs would be trapped or quarantined.

iSolation Server does exactly this. It is an effective solution to new and unknown (outbreak) viruses. It allows the open use of e-mail within organizations and permits the productive use of attachments. Before unproven programs are allowed into a network, iSolation Server intercepts the e-mail that contains these programs and sends them to a protected, virtual environment that is isolated from the regular production network. Inside the virtual environment, the program is executed and performs all of the functions that it normally would. Each action the program takes is carefully monitored and noted – and compared against a set of rules that determine whether or not the program is malicious in nature. If a program is determined to be malicious or viral it is quarantined so that e-mail or security administrators can examine it more closely. Other innocent e-mail programs and attachments are forwarded on to the e-mail server.

The virtual environment within iSolation Server is able to run different environments so that all target environments in the network can be simulated and protected. After each execution of suspect code the virtual environment is rebuilt to avoid any legacy remnants of suspect code that may be lingering, and is ready to analyze the next program.

## How iIsolation Server Works

iIsolation Server is designed and built to work with e-mail systems to filter out viruses carried on e-mail and prevent them from entering the network. iIsolation Server is based on the concept of monitoring executable behavior instead of looking for an embedded digital pattern or signature that identifies a virus. But unlike other behavior-based security systems that merely hook or trap API calls, iIsolation Server executes suspect programs—allowing them to run within an isolated environment where all program actions can be monitored. The isolated environment simulates real target environments and is built so that every program instruction can be monitored. Programs that exhibit malicious behavior or intent are quarantined, while innocent programs are forwarded to the regular e-mail system. This concept prevents not only known viruses, but also new and/or unknown viruses from entering the network.

iIsolation Server runs at the network server level, preventing viruses from entering the e-mail system network. Since viruses are contained at the edge of the network, no additional software or screening is necessary at the workstation level; this makes network management and implementation much easier.

### The Process

Using several patent-pending techniques, iIsolation Server protects users from malicious viruses that can cause much harm and destruction to their systems. As e-mail enters the network from any SMTP transfer agent, it is intercepted by iIsolation Server and routed through a series of filtering services that scan the e-mail and attachments for possible viral indications. After passing through the initial filters, iIsolation Server determines if an e-mail needs to be simulated and assigns the e-mail to an execution manager that determines which simulation environment should be used in order to most closely simulate the intended target system. After being passed into an appropriate simulation environment, the program is executed and monitored. The program is allowed to execute, and perform all actions that it normally would. All executable actions are monitored for viral behavior such as: file system access and activity, self-replication, system timer events, address book lookup, modifications and access of the system registry database, disk access, interrupt table use, and other program behavior. If the e-mail exhibits any harmful intent or behavior, it is routed to a containment chamber where further examination may be performed. If no harmful behavior is observed, the e-mail and attachments are forwarded to their intended destination.

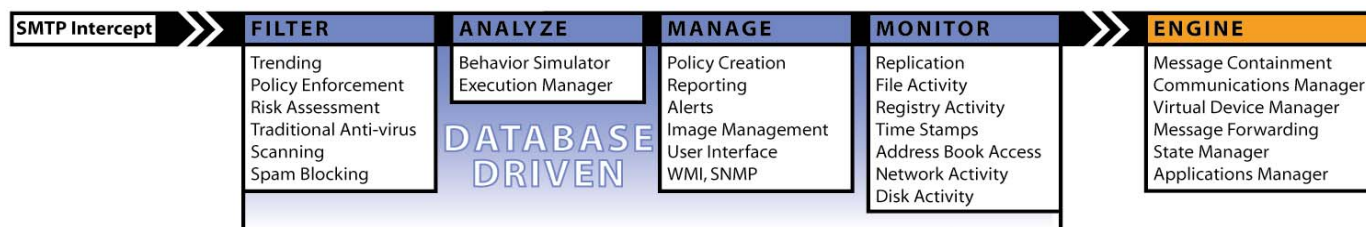


Figure 4. iIsolation Server's internal processes

iSolation Server 1.0 has the following characteristics:

- Designed and built for standard SMTP e-mail messaging environments.
- Intercepts SMTP e-mail at the Mail Transfer Agent (MTA) server level.
- Works in conjunction with other antivirus products and services.
- Provides a scalable and configurable architecture..
- Isolates and contains viruses from entering the network.
- Prevents new and unknown viruses from causing damage to networks.
- Requires no additional workstation software or support.
- Has an ASP-based user interface.
- Executes e-mail and attachments to look for viral behavior.
- Requires no updated signature files.

### *Key Internal Components of iSolation Server*

#### **Intercept Mechanism**

iSolation Server intercepts the e-mail stream using SMTP interception techniques associated with the Internet Information Services (IIS) component of Windows 2000/2003/XP servers. This allows iSolation Server to obtain e-mail records and process them before forwarding them on to the Microsoft Exchange server.

## Virtual Machine

The virtual machine within iSolation Server simulates standard PC hardware, making it possible to install, configure, and run a complete operating system within the confines of the virtual machine. All components of the virtual hardware and the operating system running thereon are contained within a virtual operating environment that can be programmatically manipulated, altered, and managed. This allows iSolation Server to execute suspect e-mails with their associated attachments and programs within an environment completely isolated from the actual network, and monitor the behaviors of the executables within the confines of the virtual machine.

A virtual machine is a self-contained operating environment behaving as if on a separate computer, similar to Java applets running in a Java virtual machine. The Java virtual machine interprets Java code, allowing it to function within the confines of the Java virtual machine and prevents any access to the host operating system—this is precisely how the virtual machine works in iSolation Server. It encapsulates executables within a virtual framework and allows them to execute without actually affecting (or harming) the real system.

Generally, virtual machines have two benefits:

1. **System Independence.** Applications run the same in a given virtual machine without regard to the real hardware and software underlying the system.
2. **Security.** Because the virtual machine has no contact with the real operating system, very little possibility exists that a program running in the virtual machine will damage files, applications, or users of the real system.

There are also two drawbacks to virtual machines:

1. **Access.** The virtual system, by its very nature, lies apart from the underlying operating system and has less access to its functions.
2. **Speed.** Because machine instructions must be simulated or interpreted in a virtual machine, program instructions take many times the amount of actual machine cycles to execute. Hence, program execution is drastically slower.

## Isolation Details

To more effectively protect the network from malicious threats and attacks, iSolation Server runs on dedicated hardware separate from the standard networking and e-mail hardware already installed. This isolates the examination and quarantining of suspect executables from the rest of the system. iSolation Server monitors and executes suspect executables in a virtual machine—providing a logical barrier from the real machine. This barrier prevents any actions taking place inside the virtual machine from affecting the real machine or network. iSolation Server contains and isolates viral behavior within the virtual machine and stops it from migrating to the real world.

## Quarantine Area

iSolation Server detects programs that exhibit pre-determined illegal parameters that define unacceptable behavior. Any executable exhibiting such behavior is marked for containment and forwarded to a quarantine area to be further examined by a system administrator or deleted from the system. Executables not exhibiting such behavior are forwarded to the regular e-mail system where they are sent to the intended recipients.

## *iSolation Server Performance, Throughput, Latency, and Requirements*

### **Performance**

The overall performance of iSolation Server is predicated by multiple parameters, including the following:

- Number of e-mail messages received per minute by the e-mail system
- Percentage of e-mails containing executable elements in the body
- Percentage of e-mails containing executable attachments
- Number of e-mails needing to be simulated
- Capability of the hardware where iSolation Server is running
- Number of processors
- Amount of memory
- Speed of network channels
- Number of separate physical machines used to run iSolation Server

Designed to operate with typical e-mail systems, iSolation Server needs to operate on a machine equivalent in processor and memory to the machine running the e-mail server. The machine running iSolation Server, however, does not need to be equivalent in disk storage capacity to the e-mail server because it runs none of the e-mail associated stores.

Some of the heaviest processing of iSolation Server is performed in running its virtual machines. Virtual machines may utilize 100% of available processing power on the machine where they are running. However, because of the multi-threaded characteristics of iSolation Server, it takes advantage of the capabilities of multi-processor machines to operate at peak performance.

The multi-threaded design of iSolation Server also allows for high system throughput. When executable e-mail attachments are not being executed and monitored, the system is intercepting the e-mail stream and looking for other executables. E-mail messages containing only clear text are rapidly forwarded on to the normal e-mail system, as are other messages that do not contain executable attachments. In this manner, high e-mail throughput is attained while executable content is siphoned from the e-mail stream and sent to the iSolation Server simulator engine to be processed.

Executables examined by iSolation Server arrive in the end-user's mailbox later than they would have without having passed through iSolation Server, but these e-mail messages have been examined and monitored for illegal actions and behavior.

True isolation and security comes at the price of speed in examining suspect executables. This is similar to the analogy of scanning carry-on baggage at the airport: Travelers must take the extra time necessary for their bags to be x-rayed, examined, and tested. Everyone, however, feels more secure that everyone else's baggage has been checked, too. This imposes a necessary latency on the overall system, but generally travelers are comfortable with the extra latency knowing that they have better chances of arriving safely at their destination.

## **Throughput and Latency**

iSolation Server performs the following general tasks:

- Intercepts e-mail.
- Determines if executable portions exist in the e-mail or in any attachments.
- Forwards non-executable e-mail to the normal e-mail system.
- Determines the extent of executable code associated with each e-mail/attachment.
- Runs e-mail through several input filters
- Determines most suitable operating environment in which to run the suspect e-mail.
- Passes the suspect e-mail and/or attachment into a protected operating environment.
- Executes the e-mail/attachment.
- Monitors and notes execution elements.
- Determines if observed execution details violate approved behavior.
- Forwards innocent e-mail to the normal e-mail system.
- Quarantines or stops suspicious or malevolent e-mail from being delivered, reporting such actions to the system administrator.
- Loads a new copy of the protected operating environment so that a new e-mail can be tested.

The overhead associated with executing all e-mail and attachments is substantial, requiring significant processor time and computer resources. For security as well as performance reasons, iSolation Server runs on its own dedicated machine (or series of machines in larger environments in subsequent releases). The system is designed to operate as quickly as possible while still affording maximum e-mail throughput. Multi-threaded techniques are employed that allow the system to perform multiple tasks in parallel. Performance enhancements are attained when using multi-processor machines to run iSolation Server—in large-scale environments, multiple machines will be used to run iSolation Server.

In order to maximize system throughput and performance, iSolation Server does not attempt to analyze non-executable e-mail such as clear text and other non-executable e-mail messages. The system sends such messages directly to the e-mail server.

Some executables only take a few milliseconds to run and be analyzed, while others take much longer. iSolation Server is programmed to quarantine executables that fail to exhibit any executable behavior within a default time period, even if they do not display malicious behavior.

Executing and examining e-mail and attachments before forwarding to the e-mail server induces latency—the required amount of time to successfully manipulate and examine all behavioral aspects of the executable—into the e-mail stream. Latency is estimated at several seconds for normal executables, with larger and more complex files and executables taking up to several minutes. Non-executable e-mail components will only incur the amount of time for the system's decision to forward the e-mail to the regular e-mail server. Because the system is multi-threaded and is working on multiple e-mail messages simultaneously, it does not stop the flow of e-mail into the e-mail server.

## **System Requirements**

iSolation Server runs on Microsoft Windows 2000 servers and is designed to work with standard SMTP based e-mail servers. iSolation Server must be installed on dedicated hardware that is network connected to the e-mail server.

The preferred system requirements for the machine running iSolation Server are:

- Intel server class machine
- Dual Xeon processors—2.4 GHz or above
- Minimum 2GB memory
- At least 1 100 Mb Ethernet NIC
- CD-ROM
- Minimum disk requirement: 30 GB

Minimum system requirements for the machine running iSolation Server are:

- Pentium 4—1 GHz
- 512 MB system memory
- 100 Mb Ethernet NIC
- CD-ROM
- 30 GB disk

## Conclusion

Traditional antivirus software systems are becoming increasingly susceptible to false negatives because they allow new and unknown viruses into networks before virus signatures can be updated and installed. New viruses spread rapidly and are increasingly complex—worsening the window of vulnerability in which networks are exposed to viral threats. More sophisticated means are necessary to stem the tide of new and unknown viruses.

E-mail is essential for high productivity in business, yet it is the primary vector for spreading viral infections. Instead of limiting the use of e-mail or crippling the functionality of e-mail by disallowing attachments, new technology is needed that will protect the users of e-mail systems from new and unknown viruses.

iSolation Server protects e-mail systems from new and unknown viruses by examining executable behavior in an isolated environment *before* the e-mail is delivered to the normal e-mail system. E-mail or attachments exhibiting aberrations from acceptable behavior are quarantined by iSolation Server, while clean e-mail is forwarded to the regular e-mail system. Avinti's patented system allows system administrators to permit full use of e-mail and e-mail attachments.

## About Avinti

Avinti, Inc. provides “day-zero” virus outbreak containment software that bridges the security gap between virus outbreak and signature update distribution. Avinti provides a server-based solution—the iIsolation Server—that executes suspect e-mail and attachments in a virtual machine and quarantines messages demonstrating malicious behavior. Avinti’s technology complements traditional virus scanning technology as part of a layered solution to quarantine potentially dangerous e-mail and protect networks from new and unknown viruses.

### *Contact Information*

For more information, visit <http://www.avinti.com/>

Avinti, Inc.  
1875 S. State St. #T-500  
Orem, UT 84097  
Phone: 801-762-0707  
Fax: 801-762-0808  
E-mail: [info@avinti.com](mailto:info@avinti.com)