

No Time for Zero-Day Solutions ***John Muir, Managing Partner***

Executive Summary

Innovations in virus construction and propagation have created a “zero-day” threat from email attachments that can wreak significant damage without warning. A variety of zero-day security solutions have been developed to overcome the “window of vulnerability” associated with the conventional signature-based solutions. However, these zero-day solutions can succeed only to the extent that they correctly anticipate viral behaviors or sources, and so are not truly effective. In contrast, Avinti’s Isolation Server is a next generation “time-zero” security product that empirically tests suspect content in a safe, configurable virtual machine with no reliance on advance warning and no assumptions about virus behavior or sources.

Introduction

In the world of information security, nothing is as sinister as an attack that arrives without warning and creates havoc with no time to respond. Initially dubbed “zero-day” threats to distinguish them from the slower moving menaces of days past, assaults of this type are obviously more dangerous. However, given the increasing sophistication and virulence of automated attacks propagated across the Internet, where attacks can have huge impacts within hours, it is now more appropriate to think in terms of “zero-time” threats.

There are a variety of ways in which a zero-time attack might be mounted including blended-threat worms that arrive through undefended network ports or hidden malware unwittingly down-loaded from websites. But the vast majority of zero-time attacks, an estimated 80% or more, are propagated by means of executable code embedded in email attachments. Although viruses have been distributed by email attachments for years, zero-time viruses are more dangerous because they combine speed, stealth and surprise to achieve maximum destruction:

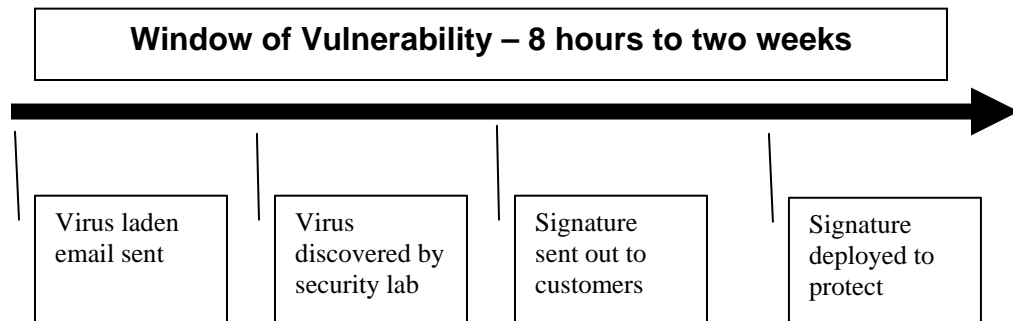
Viruses seek to overwhelm large numbers of PCs before a defense can be mounted. Sophisticated replication capabilities can swiftly propel modern viruses through the Internet in just a few hours. Alternatively, a virus can spread gradually in stealth mode and not go “active” until a certain date or trigger event occurs. In this manner the virus can slip past defense systems that look for unusual amounts of network traffic.

Regardless of how swiftly the virus spreads, once it becomes active, substantial damage can occur before the victims are even aware of the attack. The resulting damage can range from denial of service, data corruption, data loss, or the covert implanting of spyware, key loggers and other dangerous malware.

Window of Vulnerability

First generation anti-virus solutions utilize a large file of known virus “signatures” to filter out messages that contain viruses. The fundamental problem is that the signature process requires time. Even after anti-virus labs detect a virus and determine how to defuse it, the virus signature and associated fixes need to be distributed and implemented by customers to prevent further infection. The period of time between the first discovery of a virus and the implementation of signature recognition is called the

“window of vulnerability” and can last from 8 hours in the best case to several days or weeks in the worst case. Often the window of vulnerability remains open far longer than the theoretical minimum time for deployment because system administrators might be dealing with multiple viruses simultaneously, or may be distracted by other security issues, or simply are lax in their performance. For all these reasons, many systems remain unprotected long after warnings have been posted on the Internet.



Given the inability of signature-based virus solutions to provide full protection, some organizations have taken drastic measures such as blocking all email attachments or quarantining messages with attachments until the assumed window of vulnerability has been passed. While these measures are indeed effective for virus prevention, the costs in terms of productivity and user satisfaction are very high. Clearly a more sophisticated technology is required.

Overview of “zero-day” solutions for email security

To close the window of vulnerability, a variety of “zero-day” solutions have been developed. These second generation products are designed specifically to defend against unknown threats. Instead of trying to recognize virus “signatures”, zero-day solutions are designed to recognize and prevent viral behavior. This approach makes sense because ultimately the goal is not so much to identify viruses, but to prevent damage.

Zero-day virus solutions can be implemented either at the Internet-network gateway or on individual PCs. Cautious organizations may implement both types in order to create a stronger, “layered” defense. Network gateway implementations reduce the work factor for administrators and PCs alike, and prevent contagion from getting inside the network. However gateway implementations must be scalable and fault-tolerant to avoid creating network logjams and potentially introducing a single point of failure.

In contrast, PC-based security solutions are a last line of defense that catches anything missed at the network level. However PC hosted solutions require full deployment and continual updates, may have an impact on PC performance, and may present users with security decisions they are not equipped to handle.

Types of zero-day solutions for email security

The following table describes the three alternative techniques that zero-day virus solutions utilize to prevent the spread of infected files: heuristics, behavior based analysis, and reputation filters. The table also includes the two “self-help” methods used by organizations to combat viruses.

Heuristics is a method that looks for virus-like properties in email attachments that can subsequently be blocked or quarantined. Although heuristics succeeds in catching some viruses, overall success has not been satisfactory. In addition to missing viruses that present previously unknown attributes, heuristic systems frequently generate a large number of false positive warnings that consume time and may mask real problems.

Behavior-based detection makes assumptions about how suspect code performs. Incoming attachments are captured and analyzed by a software emulator that attempts to determine the intended actions of the attachment. Unfortunately emulation is not the same as actual behavior so cleverly packaged viruses can still slip by. Equally significant, behavior-based systems tend to create substantial numbers of false positive alarms.

Source filters take an entirely different approach. Rather than trying to determine whether executable code is viral, a source filter simply screens out email emanating from sources that have been associated with viruses in the past. Because large numbers of emails are blocked from the outset without the need to analyze content, this technology is attractive because it saves computation resources rather than consuming them. The downside is that source filters introduce a new window of vulnerability starting from when a site begins to send out infected traffic until the point in time when the site is identified and its location sent out to the source filter. Also, many viruses emanate from “bot” machines that have been suborned by other machines to disguise the ultimate source of the virus. Consequently, a source filter is very likely to miss dangerous viruses.

Comparison of Conventional Methods for Detecting and Preventing Viruses

Security Technology	Definition	Pros	Cons
Heuristics	Rule of thumb methods to detect virus-like properties in attachments	<ul style="list-style-type: none">• Works some of the time	<ul style="list-style-type: none">• Inaccurate - high risk of missing dangerous viruses• Can only deal with virus attributes that have been previously experience• Many false positives
Behavior-based detection	Similar to security sandboxes, looks for anomalous behavior in protected area of memory or using software emulators	<ul style="list-style-type: none">• Eliminates viruses that conform to known virus characteristics	<ul style="list-style-type: none">• Slows down system• Significant false positives• Cannot detect viruses with previously unseen

			behavior
Source Filters	Eliminates traffic from known bad sources and domains	<ul style="list-style-type: none"> • Reduces computational overhead relative to heuristics and behavior based detection 	<ul style="list-style-type: none"> • Impossible to track all bad sites • Can be fooled by bots
Attachment blocking	Prevent messages with attachments from passing mail server	<ul style="list-style-type: none"> • Always prevents viruses, if all Attachments are blocked 	<ul style="list-style-type: none"> • Huge impact on productivity • Many users create secret accounts
Email quarantining	Hold messages with email attachments for a period assumed greater than the vulnerability window	<ul style="list-style-type: none"> • Effective for most viruses • Low cost • Easy to implement 	<ul style="list-style-type: none"> • Slows down worker productivity • Signature and patch updates must be kept very current or protection is lost

Organizations that have experimented with heuristics, behavior-based detection or source filters have found that the reduction in risk is not sufficient to outweigh the administration expense, particularly when false positives are taken into account. Consequently, a surprisingly large number of organizations have decided to either block attachments altogether or to quarantine them for a period calculated to exceed the window of vulnerability. Blocking attachments certainly increases security, but the cost in terms of productivity is generally substantial and is guaranteed to aggravate managers and employees alike. Quarantining messages for a week or two sounds less draconian, but is in reality almost the equivalent to blocking attachments because the information so often has a short shelf life.

In conclusion, the second generation “zero-day” virus solutions are only marginally better than the first generation signature recognition products because they still rely for protection on something that must be known in advance or guessed at in the present. The only difference is that first generation products rely on knowing virus signatures, whereas second generation products rely on understanding virus properties, behavior or sources. Either way, advance knowledge is required in order to thwart viruses and so neither category can really be considered “zero-time”.

Protection With No Requirement For Advance Knowledge

What is required, then, is a third generation of virus protection that requires no advance knowledge of virus signatures, behaviors, or sources. This essential requirement led to the development of Avinti’s iIsolation Server, a proprietary software product that runs on standard Windows platforms. In contrast with all previous types of virus solutions, iIsolation Server un.masks viral executables instead of trying to identify specific viruses or anticipate what behaviors.

The fundamental innovation behind iIsolation Server is the use of a virtual machine that simulates any PC environment. Rather than looking at how an executable is constructed and theorizing what it might do, as is the case with emulators, the iIsolation Server examines suspect code in an environment that behaves exactly like a full PC. The iIsolation Server can be rapidly configured to simulate any PC or server environment and to compress virtual time so that any hidden viruses will go active and give themselves away. Anything that behaves like a virus is deemed dangerous and blocked from subsequent distribution; anything questionable is quarantined for review by administrators.

Paradoxically, iIsolation Server's simulation of reality provides the best approach to combating real-world zero-day viruses. In contrast to other zero-day solutions, Avinti does not rely on being more clever than virus designers. No attempt is made to determine in advance how a virus may spread or infect PCs. Instead, iIsolation Server takes a much more straightforward and fundamental approach – the only way to know if code is viral is to see what it does, and this should only be done in a controlled, isolated space.

Isolation Server is designed as a network gateway product that stops viral infections before they can pass inside the perimeter to attack multiple machines. Because iIsolation Server can determine with an extraordinary degree of accuracy whether or not an attachment is viral, there are virtually no false positives or false negatives. Further, because iIsolation Server runs on standard Windows platforms, it can easily scale for large organizations. The one weakness is that iIsolation Server may not catch viruses that require user input as a trigger.

Avinti Technology Comprises a Third Generation “Time-Zero” Anti-Virus Solution

Generation of Solution	Reactive Solutions	Pro-Active Solutions
3rd Generation – Time Zero		<ul style="list-style-type: none"> • Avinti virtual machine testing • Prohibit email attachments
2nd Generation – “Day Zero” Solutions	<ul style="list-style-type: none"> • Quarantine messages with suspect attachments 	<ul style="list-style-type: none"> • Heuristics • Source Filters • Behavior based detection • Prohibit email attachments
1st Generation – Solutions require one or more days	<ul style="list-style-type: none"> • Virus signature recognition • Update patches frequently 	

Summary

Given the virulence and sophisticated propagation of viruses, any solution that requires time cannot provide effective protection. Thus the notion of a “zero-day” solution must be changed to a zero-time solution to provide an accurate perspective. Current “zero-day” solutions are not zero-time solutions because they all require advance knowledge about how viruses behave or where they might originate. In contrast, the Avinti iIsolation Server

is truly a zero-time solution because it operates effectively with no advance knowledge of the signature or behavior of attacks propagated via email attachments.

Notes

Objective: Increase sales success rate and decrease sales cycle time for iSolation server by giving sales personnel and channel reps an easy way to:

1. Provide an accurate definition of Zero-time Threats that exposes specious claims by various vendors claiming a “zero-time” solution
2. Explain that existing solutions all suffer from the same weakness – a belief in the ability to determine in advance how to detect and defend against all new types of attacks
3. Show that iSolation Server is completely distinct; instead of attempting to try to determine in advance how to detect all new threats, iSolation Server isolates them to prevent damage
