Stopping Malware Spread From Untrusted Hosts

Nevis Networks
Defining Enterprise LAN Security

Table of Contents

Summary

This white paper discusses developing trends for malware threats, metrics for measuring the effectiveness of solutions, and Nevis' approaches to the problem from a technical perspective. This paper is intended for network and security architects with responsibility for operational networks, and for managers who desire a general understanding of the issues in preventing spread of malware internal to their organization.

Introduction

It's an IT professional's worst nightmare. You get a call in the middle of the night that there's a problem with the network. By the time you get into the office, most of the network is not functioning and your voice mailbox is overflowing with messages from half the planet, or so it seems, in the first trawl through the evidence.

No doubt you have read about a similar experience by staff at San Diego County last year, where an infection of the Zotob worm crashed 12,000 desktops and impacted the network for three or four days, including a full day of total outage.

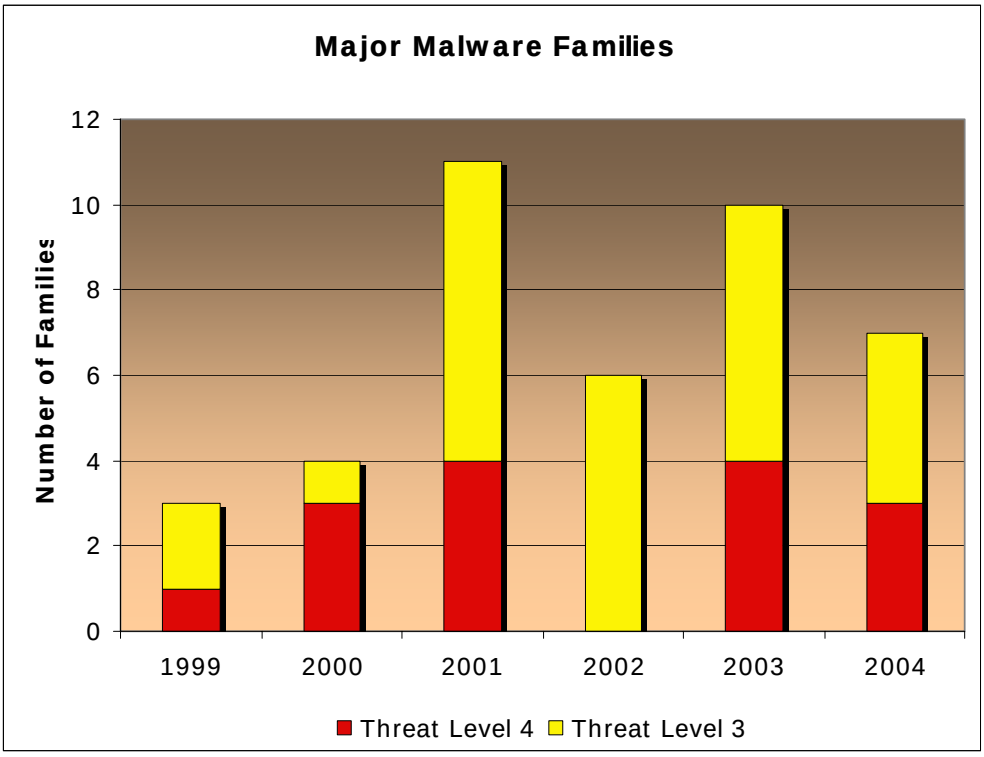

Figure 1 *- Number of major self-propagating malware families 1999-2004 as classified in seriousness (variants not counted separately). Source: Nevis Labs*

Although always serious, the threat from malware fluctuates from time to time; **Figure 1** shows the number of malware families that were Threat Level 3 or Threat Level 4 threats in the last six years.

Like several other vendors, Nevis organizes malware into five classes based on the seriousness of the incident (refer to **Table 1** below). This scale is roughly modeled on the Saffir-Simpson scale for hurricane seriousness.

Threat Level 1	Malware exists in the wild and is causing some damage.
Threat Level 2	Malware is widespread and is causing significant disruption to individual computers, but is not creating widespread organizational disruption.
Threat Level 3	Malware is creating major organizational disruption to at least some organizations.
Threat Level 4	Malware is creating major organizational disruption to many organizations.
Threat Level 5	(Not seen to date). Malware is creating a business existence threat to a number of organizations.

Table 1 *- Classification Levels for Malware Seriousness*

Although 2005 has seen a slightly lower number of Threat Level 3 and 4 threats, there have been a large number of worms and viruses this year (including Mytob, Spybot, Rontokbro, Toxbot, Erkez, Comdor, Lile, Alcra, Magflag, Suclove, Autex, Peerload, Looked, Lanieca, Ahker, Dafet, Iberio, and Pexmor) but they are using old strategies that aren't as effective at spreading broadly in the current environment. It's taking the worm-writers a little while to learn their way around the simple anti-worm measures in Service Pack 2, and the restrictions on email traffic recently imposed by most ISPs.

But worm writers will evolve solutions to these issues as the arms race unfolds, and IT staff have learned never to discount the threat completely, because when it does happen, the entire organization can come to a grinding halt until IT can remediate the problem. At least it helps your bosses to remember they need you, right?

In this white paper, we'll explain the nature of the threat, give you a framework for thinking about your organizational response to malware, and explain where Nevis' solutions can compliment your existing lines of defense, thus letting you sleep better at night.

Understanding Malware

The most severe outbreaks of the last few years have been network scanning worms and mass-emailing worms/viruses. Email malware incidents are more common (see **Figure 2**), but scanning worms spread faster and are harder to stop.

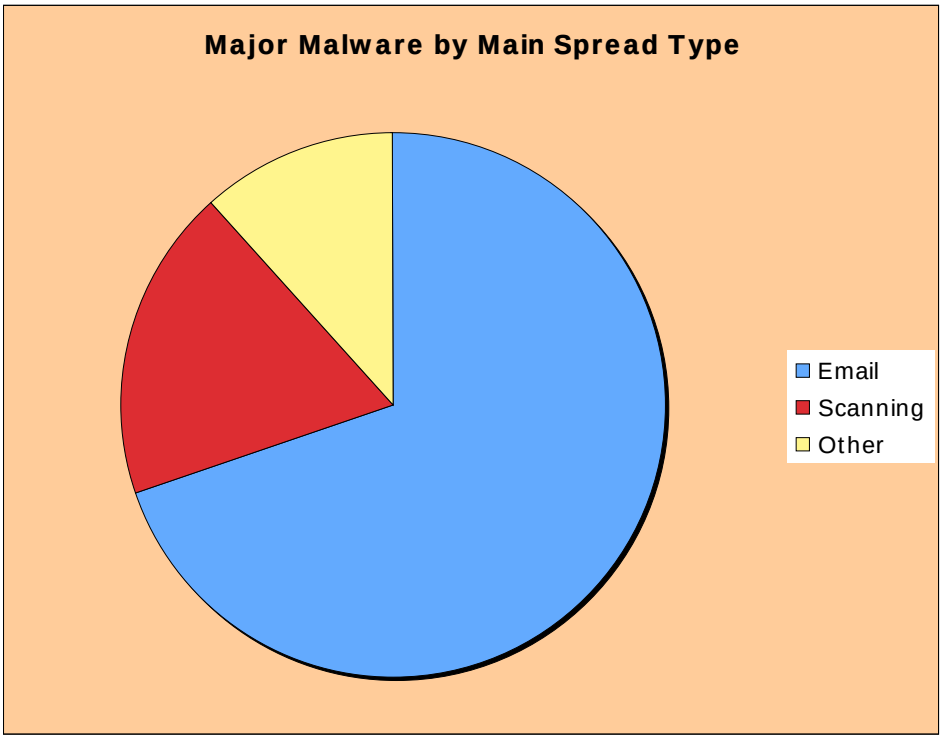

Figure 2 *- The main spread mechanism of Threat Level 3 and 4 malware (1999-2004). Source: Nevis Labs*

Network scanning worms use a network-exploitable vulnerability in computer systems (typically Windows systems, but not necessarily) and find new systems to attack by simply guessing IP addresses. More sophisticated versions will bias their guesses to be more likely to be in the same /8 or /16 range as the address of the infected computer. Some of the famous worms of the last few years of this type included Code Red in 2001 (which put scanning worms firmly on the map of IT nightmares), Slammer in 2003 (which was the fastest worm to date with a doubling time of less than 10s), and the Blaster/Welchia double hit also in 2003. In 2005, the Zotob worm family was in this class.

With the release of Service Pack 2 for Windows XP, Microsoft did place a low fence in the way of this style of worm. The most significant measure was a restriction in the kernel to only allow an application to have ten simultaneous connections open at once. TCP scanning worms had tended to use many threads to try to open many connections at once, something that Microsoft will now interdict. However, there are very simple workarounds for the worm-writers, such as using raw sockets to send crafted packets (as the nmap tool did within 24 hours of the release of SP2), or patching the kernel.

Mass emailing worms do not rely on a network vulnerability, but instead spread by tricking users into opening attachments to an email. A variety of strategies are used for generating emails – sometimes the worm mass-mails to everyone in an infected user's address book, sometimes it answers the user's email, and sometimes it watches outgoing email by the user and adds itself to that.

The chances of any given email worm attachment being opened are not very large (most users are better educated at this point), but by sending out very large numbers of copies of itself, the worm can succeed in propagating from the small percentage of users who do fall victim to it.

For email worms, a new challenge is that most DSL and Cable providers have now implemented outbound port 25 blocking except via their own SMTP servers, where they can deploy anti-virus software. This has inconvenienced users, but not as much as it has inconvenienced worm writers, who wrote a large number of email worms in 2005 using tried-and-true tactics, only to see them spread poorly. Worm writers will need to become smarter and write more believable or more polymorphic worms, write software to identify user's webmail accounts and take advantage of those, or have the worm set up its own email servers on other ports as it goes, and then use those to bounce emails off as it spreads and discovers new addresses to use. It is likely they will make those innovations and the arms-race will continue in future years.

For any piece of self-propagating malware, its critical challenge in spreading is as follows: the average instance must create more than one offspring. If it cannot do this, the infection will peter out. If it can do this, it is spreading exponentially. The more offspring it can create, and the faster it can do it, the faster it will spread and the harder it will be to stop. But creating 1.0 offspring is the critical threshold for viral success: epidemiologists call this the epidemic threshold.

The number of offspring a worm or virus succeeds in creating is equal to the number of attempts it makes before being shut down multiplied by the odds of any given attempt succeeding in actually creating an infection. Thus for email worms, we have to look at the mailing rate (the number of worms being created per unit of time), the average length of time till detection and remediation of an infected machine, and the odds of each attachment actually being opened by a naïve user on a computer where the malware code will execute successfully.

For a network scanning worm, the corresponding quantities are the scan rate (the number of addresses tried per second), the time until containment and/or remediation, and the vulnerability density (the proportion of the relevant network addresses that are actually vulnerable to this particular worm).

Thus any framework for thinking about organizational malware response needs to be in these terms: we must "think like a worm". Regardless of where they get their initial foothold, worms must face a low chance of success and a rapid containment response.

What is particularly important to understand is the importance of early containment. Just as in the case of a human epidemic, it is vitally important to contain the disease before a large number of cases develop. So in computer malware epidemics, once there are a large number of infections on the network, it becomes very difficult to contain the threat in the rest of the network as so many paths must be blocked off, and/or so many hosts identified.

This turns traditional network security prioritization on its head. Instead of worrying about identifying our most critical assets and putting protective devices in front of them, one also have to worry about identifying the least trusted, lowest security portions of our network, and make sure they cannot infect the better regulated parts of it. If we characterize these low trust risks, it implies

contractors with their own systems; business partner networks; and guests coming for short stays or to make presentations, etc. These are systems you likely have less control over than your own users, but they can still introduce immense problems.

We now turn to a more detailed consideration of threat containment metrics.

Metrics for Success

Of course, the malware metrics that are truly near-and-dear to your heart are very simple: you want the organization to control malware as quickly and cost-effectively as possible and you want to make the right choices in doing so. But we need to break the process down into a series of technical steps that we can measure, so you can make good choices and justify them to your organization.

System Response Time

It's vitally important to realize the very short timescales that can be involved in malware spread. A slow threat containment system is a system liable to fail to do its job. To give an example, Nevis has data showing instances of the Slammer worm sending out 30,700 worms per second. That's a single host producing a new copy of the worm every 33 microseconds. We calculate that a modern computer on a gigabit network infected with an efficient TCP scanning worm could issue substantially over one million TCP syns per second. That's one every single microsecond.

In peer-reviewed Nevis research, we established that an advanced worm design could infect almost all of one million target computers all over the world in 510 milliseconds.[1]

If you consider systems that gather information from around a global network, process it, and then begin laboriously querying switch MIBS and setting ACLS to respond to a worm, you'll realize such systems must take literally seconds to respond. That's an eternity in worm time. In the time a centralized system takes to think and respond, the worm can have set in motion enough infective packets to target every system on the network.

So a critically important metric for a threat containment system is its response time. It must be right there, where the heat is, in front of infected systems. And it must be responding in microseconds, not seconds or even milliseconds.

Containment Threshold

Another way to think about the response of a containment system is the containment threshold. This is the number of infection attempts that a system will allow before choking off further spread attempts. For a system with a pre-existing signature for the problem, this number should be zero. However, worms can spread much faster than signatures can be developed and propagated, so good systems will have a behavior-based component to back up any signature mechanism. These algorithms necessarily must

[1] The Top Speed of Flash Worms. S. Staniford et al. Proceedings of ACM CCS WORM, October 2004. Also available at http://www.icir.org/vern/papers/topspeed-worm04.pdf

gather a certain amount of evidence of bad behavior before cutting the system off. A few attempts make it out, but unless the odds of any given attempt succeeding are high, the epidemic will not spread.

For a system operating as the network access point for an infected host, this number can be quite small – the state of the art is in the range of 1-6 depending on the application. This mushrooms, however, as the detection and containment mechanism is moved deeper into the network. Using only a handful of "zone firewalls" has the drawback that an entire zone can get infected before the worm makes any attempt to move out of the zone. This effect is shown in **Figure 3** which plots the achievable containment threshold as a function of which tier in the network the detection/containment mechanism is implemented in. Mechanisms near the core allow many more worm instances to be delivered, endangering the network.

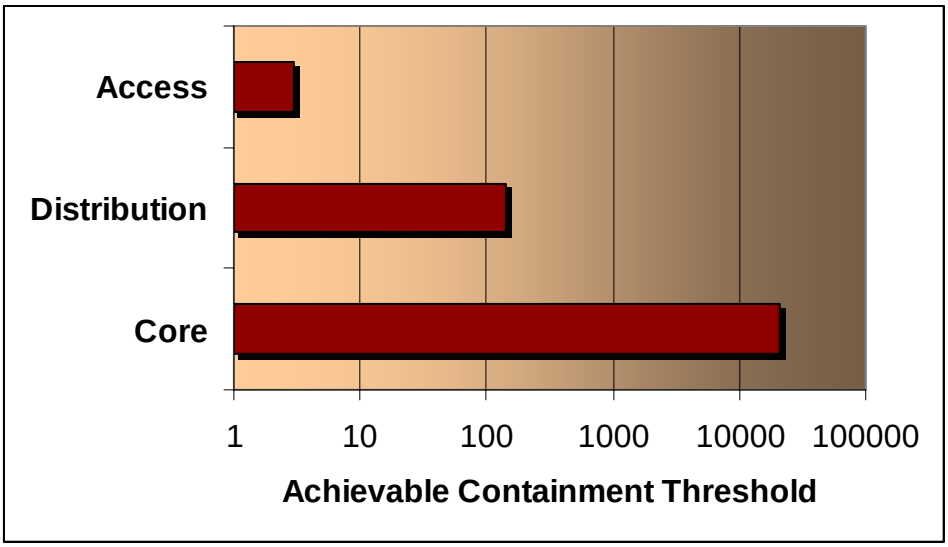

Figure 3 *- Typical achievable containment thresholds as a function of location in the network. Scale is logarithmic. Deep in the network, the containment threshold grows large because of the risk of all the hosts in a zone infecting each other. This analysis assumes 48 port access and distribution switches. Source: Nevis Labs*

Worm Throughput

Another variable of importance is the rate at which a device can process worm traffic. If you place it at choke point, you want to be sure that it can and will work hard when it's doing one of the most important jobs you bought it for: chewing up worms and spitting them out. Some devices with a slow centralized processor for the actual inspection and identification tasks will slow to a crawl when faced with a significant infestation: the network throughput of the device for legitimate traffic will drop, the latency will rise, and the device's management interface will become sluggish and unresponsive.

To avoid this problem, a device should be able to process a large number of worms per second while blocking them. Obviously, not all worms require the same amount of effort to block, but a decent recent metric is the Slammer worm, since it was capable of very high rates of packet throughput. An effective device should be able to block in excess of 100,000 Slammer packets per second. (This corresponds to three very high speed infestations downstream). It should do this while still passing normal traffic without losing any packets.

Incident Resolution Time

A very important metric is the total amount of time required before the network is fully operational. Alternatively, when a new but infected host attempts to come on the network and is blocked, how long does it take until the host can be remediated and its user can be productive again? This metric is partly a function of your organization's own skill and knowledge, of course, but the quality of your tools also matters.

Some threat containment systems use only behavior-based systems and provide reports that amount to "Application X on System Y is anomalous and we blocked it." This tells you nothing about what the problem is or what to do. The best systems provide detailed identification of problems, and documentation about what you should do in response to the problem. Thus when a guest needs to get on your network to make an important presentation, and the security system identifies a problem, you know what needs to be done; you aren't starting diagnosis with only the vaguest indication that there's some kind of problem.

Signature Update Time

Some systems rely on signatures to recognize and block malware. A system that relies solely on signatures will be blind to malware until the signatures can be updated. Given the speed of worm spread, this is a potentially crippling disability. Thus the best systems have a behavior based component capable of blocking worms prior to the availability of the signature, but also use signatures to identify malware as and when the signatures become available. So it's important to ask how quickly the signatures are available for the product in question when a new piece of malware begins spreading – what is the company's commitment to providing accurate signatures in a prompt and timely way, together with detailed documentation that makes them actionable. This is particularly important for major malware epidemics.

The Nevis Solution

In this section, we outline the technologies relevant to threat containment in the Nevis solution. Nevis LANenforcer security appliances provide no fewer than four independent lines of defense, and below is an explanation as to how they work together to contain threats.

Endpoint Integrity Agent: The First Line of Defense

Nevis supplies a demand downloadable agent which can perform an endpoint integrity assessment and verify compliance with enterprise policies before allowing normal network access. This integrity assessment includes the current state of critical security patches for a Windows computer, whether an approved and up to date anti-virus application is currently running on the host, and whether an approved and up-to-date anti-spyware application is running, among other things. When a new computer attempts to authenticate to the network using captive portal login, we can insist that the agent be downloaded and run before access is granted. Nevis enables quarantine of vulnerable or potentially infected end systems on a restricted network until such time as they are remediated and they can be reassessed as compliant. Nevis gives you the option to enforce a policy on the host to ensure that known threats are not let loose on the network due to poorly maintained hosts.

While very effective in what it is intended to do, endpoint integrity assessment alone is not the whole answer. First of all, some hosts just cannot accommodate the agent. This includes endpoints not running Windows 2000 or XP, such as a PDA, Mac, Linux, or

Unix computer. Furthermore, even if the agent can be run, things like performing a full disk anti-virus scan cannot be forced on the user, or even measured accurately.

For these, and unknown/novel threats, and for those situations where it's not practical to impose a strict update policy on a guest, three lines of defense remain.

Firewall: The Second Line of Defense

Nevis LANenforcer can be used to implement comprehensive identity based firewall access control policies. When a user authenticates (whether via 802.1x or captive portal), they get the policy you have specified for the groups they are members of. This allows entire services and large areas of the network to be screened off from any attempt at infection by malware. For example, in LANenforcer's default guest policy, outbound SMTP access is denied altogether. Thus SMTP based worms have no possibility to infect anything anywhere. Guests by default do have the ability to use Internet webmail systems such as Gmail or MSN, and you can create selective exemptions to allow users to access SMTP servers as needed.

Additional policy groups can be defined specifically for the purpose of trapping suspected malware. For example, the default policies include a "bogon" group, that drops all traffic with a source or destination address that is not a real address, according to the IETF. A group policy could be defined that just raises high priority alerts on accesses to trojan ports. These and other such policy groups can be readily combined with identity based policy groups when users login, and merged with other groups as part of the effective user policy.

Signatures: The Third Line of Defense

We recognize that to get work done, you have to allow your employees, your contractors, and even your short-term guests to access your network. And it's not always practical to create ultimately fine-grained access control policies that precisely limit each user to only the resources they really need (though we allow you to do that). Most organizations have to strike a balance of creating broad groups with access to reasonable sections of the network resources, while more precisely protecting sensitive servers.

But in those holes that you must create in your policy, we are still watching, doing deep content inspection on every single packet that crosses our device. For known threats, we will recognize and identify the threat, block it (if so configured), and report the situation to you. We maintain labs of researchers who have no other job than to identify new threats, rapidly develop signatures for them, and ship them to you. By having labs in California, India, and China, we provide twenty four hour coverage of any situations that may arise. Our signature update commitments vary according to the severity of the threat as shown in **Table 2**.

Threat Level 5	Signatures within 6 hours of first discovery of threat. All variants that become widespread will get their own unique identification and detailed documentation.
Threat Level 4	Signatures within 6 hours of first discovery of threat. All variants that become widespread will get their own unique identification and detailed documentation.
Threat Level 3	Signatures within 12 hours of first discovery of malware. All variants that become widespread will get their own unique identification and detailed documentation.
Threat Level 2	Signatures within 24 hours of first discovery of threat. Many variants of a threat may be covered by single signature with generic documentation.
Threat Level 1	May or may not be covered by signatures at Nevis option.

Table 2 *- Signature support at varying threat seriousness levels.*

Behavior Blocking: The Fourth Line of Defense

But given that you must leave some access through firewalls, and neither Nevis nor anyone else can disseminate a signature as fast as a new worm with a zero-day vulnerability might spread, Nevis devices also incorporate state-of-the-art behavior-based techniques. Worms and malware must behave in an anomalous fashion in order to spread – they do not know where they will succeed, and they do not know the normal pattern of usage on the network. Thus they typically make many failed attempts for each successful one, and they typically go to unusual destinations far more often than normal users. Our algorithms combine all this evidence and distinguish bad behavior from good. Typically, our containment threshold causes blocking on the third attempt, but for a number of unusual services we do significantly better, while for a small handful of services we do a little worse.

Nevis' behavior blocking technology is completely integrated with our other lines of defense. For example, if a worm attempting to scan, causes connections which get blocked by the firewall, that evidence will be intelligently correlated with other behaviors of that source. If a worm issues malformed scanning packets (eg with null TCP flags) that match signatures, those too will be incorporated into the analysis.

Additionally, our analysis works across applications; if a threat has several exploits for several services, we analyze all of its behavior to rapidly draw the conclusion that it is misbehaving and shut it down, rather than allowing a separate threshold for each service (which would result in a large degradation in our containment threshold).

Nevis Labs research staff are world-class innovators and they continue to push the state of the art in understanding novel classes of worms and in developing ways to distinguish their behavior from that of normal users and applications.

Conclusions

In this paper, we've surveyed the problem of self-replicating malware and developed a framework for thinking about preventing it on your network. We emphasized the importance of containing near the source of the problem, and discussed five key metrics of a complete system:

1. Very rapid (microseconds) response time
2. Low containment threshold
3. Large worm analysis bandwidth to stay robust under fire.
4. Low total time for incident resolution, with rapid malware identification and good documentation enabling low total cost of ownership.
5. Rapid signature and documentation updates as incidents progress.

We outlined the way the various aspects of the Nevis solution which combine to provide excellent defense-in-depth on the network. Downloadable host agent to verify update compliance on endpoints; user-based firewall policies; threat identification signatures with rapid updates to signatures and documentation; and finally, behavior-based approaches to novel and rapid threats.

About Nevis Networks

Nevis Networks develops and markets ASIC-based LAN security appliances designed to help corporations protect information privacy and integrity, ensure network availability, and maintain regulatory compliance. With its patent-pending LANsecure™ architecture, the Nevis LANenforcer product family combines the most comprehensive access control, deepest threat defense, and fastest threat response to create a "Personal DMZ™" around every user on the LAN. Nevis was founded in 2002 by seasoned executives with strong track records in security, semiconductor, and networking technologies, and has raised over $40 million from veteran Silicon Valley investors New Enterprise Associates, BlueRun Ventures, and New Path Ventures. The company is headquartered in Mountain View, California, with an R&D center in Pune, India.

Nevis Networks
500 N. Bernardo Avenue
Mountain View, CA 94043
T: 650-254-2500
F: 650-254-2555
www.nevisnetworks.com