



webroot[®]
SOFTWARE, INC.

Privacy. Protection. Peace of Mind.

White Paper

Threat Chaos: Making Sense of the Online Threat Landscape

by Richard Stiennon

Webroot Software, Inc.

2560 55th Street, Boulder, CO 80301

Toll Free: 800.870.8102

Telephone: 303.442.3813

Facsimile: 303.442.3846

www.webroot.com

Privacy. Protection. Peace of Mind.

Index

Executive Summary	1
Major Online Threat Categories	1
Denial of Service Attacks and Distributed Denial of Service Attacks	3
Hacking Attacks	4
The Motivation Behind Creating Threats	4
The Threat Chaos Model	6
Into the Fourth Dimension	8
Summary	8

Executive Summary

Why is there a constant barrage of new viruses, worms, spyware, adware, hacker attacks and other potentially unwanted programs? Why is there so much confusion over naming these different forms of installation? Why don't firewalls and anti-virus products adequately protect computers from these threats? What is a threat model and how is it useful? This paper addresses these questions and proposes a threat model to clear up the confusing nomenclature.

A recent study by Gartner G2 shows that more than 20 million computer users are infected with spyware.

One third of European companies are infected with spyware.
– EIT Survey 2003

The FBI used spyware to get a computer password from jailed mob boss Nicodemo "Little Nicky" Scarfo

Major Online Threat Categories

Viruses are self-replicating code that spread between applications. Today viruses often need the user to take some action to infect a computer and spread the infection. Occasionally a new vulnerability in Microsoft Exchange or Internet Explorer allows code to execute automatically without user intervention. That spells disaster as a new virus spreads throughout the Internet with very little resistance. Most of today's viruses rely on somehow tricking the user into opening an e-mail or clicking on a hyper-link. As spammers perfect these replication techniques and as virus writers and phishers adjust their own techniques, we continue to see the spread of modern day viruses such as Netsky and Mydoom (now in their 35th, 34th variations respectively).

Worms are self replicating code that spread between services. Worm infiltration is directly connected to the history of the network security market over the past three and a half years. Prior to 2001 firewalls, anti-virus and IDS (Intrusion Detection Systems) defined network security. In the summer of 2001, a worm dubbed **Code Red** (the caffeine laced soft drink that the original discoverers were drinking at the time) exploited a well known vulnerability (patch available) in Microsoft's web server product, IIS. Code Red spread from web server to web server over port 80 undetectable by firewalls. Because there was no signature for it, IDS systems could not identify it. However, IDS could identify the huge amounts of traffic and issue alerts on it, which added to the network traffic just when networks were bogging down. Antivirus systems look for infections in file systems and need signatures, or definitions, which can take, at best, several hours to develop and deploy. This reaction time is much too slow to counter mass propagating worms. Code Red taught an important lesson: patch servers when critical vulnerabilities are announced.

To the everlasting embarrassment of the "security community", **Nimda** struck on September 18, 2001. Nimda (Admin spelled backwards) exploited the *same* vulnerability in IIS as Code Red. This worm also exploited an Internet Explorer vulnerability and opened file shares to spread to the desktop where it then replicated through e-mail like virus. Lesson learned from Nimda: *really* patch servers when critical vulnerabilities are announced. This double whammy against unpatched IIS servers led John Pescatore of Gartner to issue a now infamous recommendation that if an organization was hit by both Code Red and Nimda, they should re-evaluate their choice of web server platforms.¹

¹Pescatore, John. *Nimda worm shows you can't always patch fast enough. (2001)*. Retrieved January 7, 2005, from <http://www4.gartner.com/resources/101000/101034/101034.html>

On January 25, 2003, along comes **SQL Slammer**, a worm that targeted a known (patch available) vulnerability in Microsoft's database server product. This worm used the UDP protocol to propagate over port 1434. It spread to 80,000 machines in less than 10 minutes. SQL Slammer taught three important lessons:

1. Really, *really* patch vulnerable systems,
2. Block high level ports in routers and firewalls. (There is very little need to do SQL requests across the Internet.)
3. Sometimes services lurk in the most unlikely places.

It turns out that Microsoft development platforms, and several Microsoft applications harbored copies of SQL Server so even enterprises that had patched their servers succumbed to SQL Slammer.

Another worm deserves mention. The **Blaster** worm took advantage of a Window's vulnerability in RPC DCOM. Highly anticipated, nearly every security expert recommended that everyone immediately patch their systems and block port 445 to prevent the exploitation of the known vulnerability. Regardless, when the worm hit on August 14 2003, it caused major outages of networks and even shut down rail lines in the United States. Most enterprises reported that the worm did not penetrate properly configured firewalls, but rather entered the network through infected laptops.

Worms started a transformation of network security that is still continuing. Gateway security devices are looking deeper and deeper into the packets that pass through them. Intrusion Prevention technology has replaced Intrusion Detection systems. Gateway antivirus systems are gaining in popularity.

Trojan Horses are code that masquerade as one thing and seem benign, but are actually something else, usually malicious, but sometimes just a nuisance. For instance you can download a screen saver and unwittingly install an unwanted program through the backdoor. Trojan horses can be masters of deception. In an example of very clever opportunism, hackers created a Trojan horse that masqueraded as a controversial screen saver that Lycos was distributing. The original Lycos version launched denial of service attacks against spammers. The fake screen saver installed Perfect Keylogger, a spyware application. To avoid confusion, think of Trojan horses as disguised delivery mechanism (similar to the legendary horse used by the Greeks in the Trojan War). Do not confuse them with their payload which could be any software application. There are emerging types of programs that are similar to Trojans, but rely on user carelessness or inattention in reading end user agreements or not fully understanding Internet downloads, rather than pure deceit. Through this method, hackers can trick users into downloading a program with modest utility while its primary purpose is to deliver advertising or other unwanted applications.

Backdoors are codes that allow a hacker to have access to a computer, usually with powerful interfaces that give the hacker complete control over the machine as if they were sitting in front of it. Back Orifice, SubSeven, NetBus are some famous back doors. IRC controlled backdoors are used to kick off denial of service attacks against targets. See the following pages for more information.

"The main function of adware and spyware is to take information from a system and send it to an external source. For legitimate programs that do this, such as Web servers, standard security practice is to lock them down as much as possible and keep their patches up-to-date. But how can you secure and patch a program that you don't even know is on your system? Spyware programs are a ripe target for crackers and malicious coders looking for holes into systems."

– "Spyware Needs to Go" by Jim Rapoza
Eweek, December 1, 2003

Identity Theft on the Rise

Online consumers were defrauded of \$700M in 2002. 500,000 to 700,000 people are victims of identity theft each year.

50% of computer users polled have lost personal information to a hacker attack within the last 12 months, costing over \$285M.

– DOJ Reports

System monitors include keystroke loggers which are code that record all input from the keyboard and ship it off to the hacker. This is a useful tool for gathering usernames, passwords and other account information.

There are examples of code that will also record voice and video from a computer equipped with a microphone or camera. Results from an ongoing spy audit of enterprise systems (available on the Webroot web site) indicate that more than 14% of the tested systems have system monitors on them.²

Spyware is an elusive, ever changing threat that falls into several categories. At its most basic, spyware are programs that track your online and offline activities without your knowledge and/or consent, and share that information with a third party. Spyware can include **System Monitoring Tools** that record everything from visited web sites to chat sessions, instant messages and e-mail messages. Additionally, spyware can refer to **Keyloggers**, programs that capture every keystroke performed on a keyboard, including usernames, passwords and private data like bank account or credit card numbers.

The largest emerging supply of spyware comes from commercial sources who want to feed advertising to unsuspecting users. Spyware's more benign cousin is **Adware**. Adware is often bundled with or embedded within freeware, utilitarian programs like file sharing applications, search utilities and information providing programs (i.e. clocks, messengers, alerts, weather, etc.) and funware like screensavers, cartoon cursors, backgrounds, sounds, etc. Adware applications are usually advertising supported software that anonymously monitor your Internet surfing activities and typically displays targeted pop-up, pop-under, and other advertisements on your computer in exchange for free software. Some adware may track your Web surfing habits. Deleting adware will usually result in deletion of the bundled freeware application.

Also related to spyware and Trojans are **jokeware** or **spoofware**. These unwanted applications are usually non-malicious programs that can change settings, like cursor, sounds and background, but cause no serious damage to your computer system.

Denial of Service Attacks and Distributed Denial of Service Attacks

Opposed to the threats discussed thus far, this type of threat targets a specific victim for its attack. The target of these attacks varies, depending on the hacker's motivation. For example, sometimes the hacker determines he wants to cause damage or, possibly, extort protection money from a popular site. In some recent cases, a competitor hires a hacker to take down a web site. Various tools can spew packets at or launch specific exploits against a target.

The most powerful targeted attack is a distributed denial of service attack (DDOS). Thousands of machines, infected with IRC controlled backdoors (bots) are aimed at a specific site. These bots can generate SYN floods or just HTTP GET requests at rates that no server can handle. The controlling cyber criminals can extort payment from the owner of the targeted site. In 2004, DDOS attacks became a major threat.

²Threat library: Corporate statistics. (2005). Retrieved January 7, 2005, from http://research.spysweeper.com/threat_library/corporate_stats.php

Originally targeted at poorly defended online gaming sites, the trend now is directed towards other online services that have large numbers of transactions, such as e-commerce sites, financial services firms and foreign exchange sites.

Hacking Attacks

With the plethora of viruses, worms and spyware, it is ironic that targeted hacking attempts no longer get the attention they used to garner. A targeted attack is usually a systematic attempt by a cyber criminal to break into computers and steal data, or do other damage. As more and more services are made available over the web, these attacks will continue to target weakly designed and implemented business processes. The anatomy of an attack involves these stages:

- Foot print analysis
- Scanning
- Exploitation
- Damage

Frequently, an insider with intimate knowledge of systems and processes coupled with the motivation to either steal from their employer or cause damage is at the heart of the most destructive hack attacks.

The Motivation Behind Creating Threats

Two predominant motivations exist for creating any of the threats listed above. Since the dawn of the computer age, the most frequently cited motivation for the writers of worms and viruses was “fun.” It was viewed as adolescent males getting their kicks.

Often overlooked, however, is that these adolescent males were motivated by a desire to set the world right. They viewed faulty operating systems and applications as an affront to their geek sensibilities and their creators as evil empires taking over their domain with shoddy, insecure products. In an attempt to demonstrate to the world that these systems were flawed and the programmers were not doing their jobs, hackers would write rapidly spreading viruses and worms to infect as many machines as possible.

While this is a broad categorization of hackers, it is supported by the lack of malicious payloads in viruses and worms. Look at some of the most widely spread viruses in years past. The LoveBug virus spread through email with the subject heading “I Love You”. Its only purpose was to spread to other email addresses from the address book.

One of the most enigmatic worms ever released was the SQL Slammer worm. This was a very small piece of code that spread via UDP. Once installed on a vulnerable machine, it would send out thousands of copies of itself to random IP addresses. If any machine running the unpatched version of Microsoft SQL Server got hit by one of these packets, it too became infected. Released at midnight on Friday, January 25, 2002, the SQL Slammer worm’s only purpose was to spread itself. There was no other payload to it.

Spyware is costing the enterprise money by:

- Reducing system performance and stability
- Reducing worker productivity
- Creating software conflicts with legitimate programs
- Increasing support needs
- Stealing network resources
- Increasing network overhead
- Decreasing network performance

Slammer infected 80,000 machines in under 10 minutes. These machines generated so much traffic that the Internet almost completely melted down. Traffic volume at some major backbones doubled to near capacity limits. It was not until about noon the following day that the problem was contained. Note that if Slammer had been released at say 1 p.m. Monday on the east coast of the United States, damage could have been drastically higher as financial markets and e-commerce sites went down.

Slammer was a turning point for internet security. It had spread over high level ports that since then have been blocked by the majority of networks and even some ISP's. Blocking high level ports is a standard best practice, yet thousands of networks allowed external communication over these ports. After that fateful weekend, most firewalls and many routers were reconfigured to follow best practice. This was also a turning point for telecom carriers. For years ISP's and backbone providers had treated themselves as open conduits. Spam, denial of service attacks, viruses, and worms crossing their networks were not their problem. "The packets must get through" was their mantra. Slammer *was* their problem and they had to block it to gain back control of their networks. Now carriers tout their "secure networks" and "clean pipes" as they do more to block worms and denial of service attacks.

So not only did Slammer demonstrate some sort of ethical behavior on the part of the hacker that created and released it, but the final outcome improved the overall health of the Internet. It is not possible to condone the behavior of anyone who creates worms and malware, but SQL Slammer sheds some light on their motivation.

Now turn to 2004, the year of the virus wars. Competing factions spewed dozens of versions of their viruses in an attempt to:

1. Evade AV engines
2. Infect machines controlled by the opposing group.

The Netsky and MyDoom variants even had messages embedded in them, written in broken English, declaiming the actions of their competitors.

Each variant's purpose, however, was part of the latest trend in dangerous software, a trend with frightening implications. These viruses and, indeed most of the malware introduced in 2004, had criminal intent. Netsky and MyDoom made it possible for hackers to install software on infected machines that gave the hackers control of those machines through command channels such as IRC (Internet Relay Chat) effectively turning millions of machines into zombies (bots) that could be commanded to launch network attacks against any target leading to the extortion attempts discussed above.

2004 shaped up to be the year that the primary motivation for writing and distributing software shifted from the old motivations of fun, bragging rights, or fighting the evil empire to financial gain, sometimes legitimate, sometimes using deception, nonsense, extortion and outright theft.

Spyware can open new hacker exploits into end user computers.

Example:

An end user has installed Gator on their system. Gator has an open port that allows it to communicate to it's master server. A hacker could easily use a buffer overflow attack in order to gain access to the computer through Gator.

The Threat Chaos Model

The model proposed here attempts to add structure to understanding the threat scope, thus eliminating the chaotic interpretation of threats.

There are three axes to the threat model. The first is **vulnerabilities**.

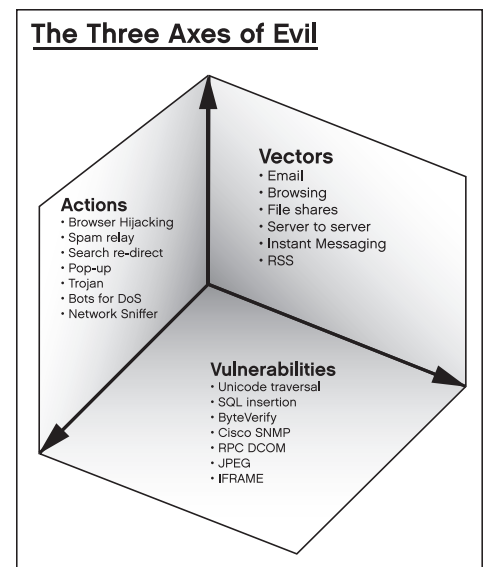
Although thousands of vulnerabilities are reported in recent years, in hundreds of applications and all of the major operating systems, only vulnerabilities in widely distributed systems tend to have major, global impact. Hackers are much more likely to exploit vulnerabilities in Windows, Internet Explorer, Exchange, Outlook, or Cisco routing platforms for commercial gain. All of these critical vulnerabilities would be arrayed along the first axis.

Of course, the criticality associated with every vulnerability decays soon after a patch is available. That is to say, the universe of vulnerable machines starts to shrink as the patch is deployed. It never goes to zero though. Witness the number of SQL Slammer packets still bouncing around the Internet, or Nimda or Code Red, for that matter. Examples of these vulnerabilities are: the Unicode traversal flaw in Microsoft IIS that led to Code Red and Nimda. The byte verify flaw in Microsoft Internet Explorer used by many spyware programs. The vulnerability in Microsoft SQL Server that was exploited by SQL Slammer. The Cisco IOS TCP and SNMP vulnerabilities that could lead to denial of service attacks against routing infrastructure.

The second axis is **vectors**, which are all the modes of communication that malware can utilize to infect a machine. These include floppy disks, flash memory sticks, e-mail, web browsing, ftp, peer-2-peer file sharing, netbui, UDP, Instant Messaging, Wi-Fi, and one that has not been exploited yet: RSS real time news feeds.

And the third dimension is the axis of **nefarious action**. While the other two axes are fairly easy to populate with complete lists of known vulnerabilities and known communication channels, this axis requires extrapolation and use of a crystal ball to envision all of the possible actions cyber criminals could develop. But a starting list of nefarious actions would include:

- Propagation
- Data theft
- Identity theft
- Browser hijacking
- Search hijacking
- HOSTS file overwriting
- System crash (ping of death)
- File destruction
- Hardware destruction (only theoretical to-date)
- Allow remote control of PC
- Steal CPU cycles (for cracking encryption keys for instance)
- Pop-up ads
- Phishing
- Network outage
- Network re-directs



Spyware in the Enterprise

Keyloggers were planted on a number of machines at Valve Software creating a security hole used to gather information and steal a good portion of the source code for Half Life 2. Ultimately this intrusion delayed the release of one of the most highly-anticipated games in history by at least six months. Not only has this delay negatively affected revenue for the creator, distributors, other parties licensing the engine, retailers, etc., but it has also generated unauthorized copies around the world.

JuJu Jioang was able to install key loggers on computers in at least 15 Kinko's Stores in New York, which he used to capture user information such as user names and passwords. Ultimately, he used this material to gain access to existing bank accounts and open new lines of credit affecting more than 450 victims.

This threat model assumes that every threat, both historical and emerging, is a combination of these three aspects: exploitable vulnerability, vector and action. A few examples:

MyDoomXX:

Vulnerability: Files execute automatically in Windows Outlook
Vector: E-mail
Intended Action: Harvest bots for eventual use in denial of service attacks

Nimda:

Vulnerabilities: Unicode traversal in MS IIS
File execution in Outlook
Backdoor left by Code Red
Vectors: Network over port 80
E-mail
Browsing
Open file shares
Intended Action: Propagation

FunnerA.32:

Vulnerability: User privilege
Vector: IM
Intended Action: Populate hosts file with Chinese domains

Note that Nimda was a worm that took advantage of three different vulnerabilities and four different vectors to spread, making it one of the most difficult worms to contain ever.

A few observations are possible with this three dimensional threat model. First, malware is eventually observed for every combination of vulnerability, vector and action. There must be enough malware writers all seeking new, undefended avenues of attack to fully explore this space. The malware pioneers today are writing spyware and phishing attacks. Their rewards are immediate, paying off in click-throughs to affiliate sites or in harvesting of account names, passwords and numbers. The Webroot Threat Research Center discovers and publishes more than 20 new pieces of spyware a week and 80 variants of existing spyware. This discovery rate is approaching the rate at which virus variants are being discovered. One additional observation, however, is that those viruses have, for the most part, shifted in intent from just propagation to harvesting bots or spreading spyware payloads.

Into the Fourth Dimension

Note that because of its purpose, spyware and related programs introduce a fourth dimension to a complete threat model: the extent to which a spy prevents detection and removal. Spyware, adware and promoters of other potentially unwanted programs profit only as long as a program is resident on a PC. If it is readily caught and removed, it is less effective and less profitable than if it is stealthy and resistant to eradication. Thus, spyware and other potentially unwanted programs can install with randomized file names and registry keys, randomized code (to hide from “fingerprinting”), and in several, even dozens, of places. Special listening code can be installed to detect removal from start up directories and re-install the program with yet more randomization. The latest spyware programs change security settings in Windows and replace key Microsoft .dll’s making it almost impossible to remove.

If included in a threat model, this fourth dimension would serve to identify the most pernicious software and help to target defenses. Changes to the operating system and standard settings could be devised to reduce the effectiveness and, therefore, the cost of the most resilient programs.

Summary

Confusion abounds over the various names of threats. A model that categorizes them by action, vector and vulnerability will allow future efforts to counter these threats by focusing on the most dangerous ones. With careful consideration about new intents, it should be possible to predict new threats based on combining those nefarious intents with existing vulnerabilities and vectors. The Worm-SDBot that carried a network sniffer as payload could have been predicted, for instance.

As the overall malware intent shifts from pure propagation to propagation and earnings potential, the number of threats will grow at multiples of rates seen before. Each cyber criminal (or gang) needs to have their own tools generating these earnings, after all.

About Webroot Software, Inc.

Webroot Software, a privately held company based in Boulder, Colorado, creates innovative privacy, protection and performance products and services for millions of users around the world, ranging from enterprises, Internet service providers, government agencies and higher education institutions, to small businesses and individuals. The company provides a suite of high-quality, easy-to-use software that guides and empowers users as they surf the Web, protecting personal information and returning control over computing environments. Webroot’s software consistently receives top ratings and recommendations by respected third-party media and product reviewers.

For more information about Webroot Software or Spy Sweeper Enterprise, please visit www.webroot.com or call 800-870-8102.

Privacy. Protection. Peace of Mind.

2560 55th Street, Boulder, CO 80301

Toll Free: 800.870.8102

Telephone: 303.442.3813

Facsimile: 303.442.3846

www.webroot.com