Battling Botnets for Control of Computers

Microsoft | Security Intelligence Report

Volume 9

January through June 2010

Microsoft®

Authors

David Anselmi
Digital Crimes Unit

Richard Boscovich
Digital Crimes Unit

T.J. Campana
Digital Crimes Unit

Neil Carpenter
CSS Security

Greg Cottingham
CSS Security

Joe Faulhaber
Microsoft Malware Protection Center

Vinny Gullotto
Microsoft Malware Protection Center

Paul Henry
Wadeware LLC

Jeannette Jarvis
CSS Security

Jeff Jones
Microsoft Trustworthy Computing

Jimmy Kuo
Microsoft Malware Protection Center

Scott Molenkamp
Microsoft Malware Protection Center

Michelle Meyer
Microsoft Trustworthy Computing

Bala Neerumalla
Microsoft Secure SQL Initiative Team

Daryl Pecelj
Microsoft IT Information Security and Risk Management

Anthony Penta
Microsoft Windows Safety Platform

Paul Pottorff
Windows Consumer Product Management

Tim Rains
Microsoft Trustworthy Computing

Javier Salido
Microsoft Trustworthy Computing

Navaneethan Santhanam
Bing

Christian Seifert
Bing

Frank Simorjay
Microsoft Trustworthy Computing

Holly Stewart
Microsoft Malware Protection Center

Adrian Stone
Microsoft Security Response Center

Matt Thomlinson
Microsoft Security Response Center

Jossie Tirado Arroyo
Microsoft IT Information Security and Risk Management

Scott Wu
Microsoft Malware Protection Center

Terry Zink
Microsoft Forefront Online Protection for Exchange

Contributors

Ian Brackenbury
Microsoft Trustworthy Computing

Doug Cavit
Microsoft Trustworthy Computing

Eva Chow
Microsoft IT Information Security and Risk Management

Greg Cottingham
CSS Security

Dave Dittrich
University of Washington

Enrique Gonzalez
Microsoft Malware Protection Center

Cristin Goodwin
Microsoft Legal and Corporate Affairs

Satomi Hayakawa
CSS Japan Security Response Team

Robert Hensing
Microsoft Security Response CenterCon-sulting Services

Yuhui Huang
Microsoft Malware Protection Center

CSS Japan Security Response Team
Microsoft Japan

Joe Johnson
Microsoft Malware Protection Center

John Lambert
Microsoft Security Engineering Center

Laura Lemire
Microsoft Legal and Corporate Affairs

Nishanth Lingamneni
Microsoft Security Essentials

Ken Malcolmson
Microsoft Trustworthy Computing

Russ McRee
Global Foundation Services

Charles McColgan
Microsoft ISD

Mark Miller
Microsoft Trustworthy Computing

Price Oden
Microsoft IT Information Security and Risk Management

Kathy Phillips
Microsoft Legal and Corporate Affairs

Anthony Potestivo
Microsoft IT Information Security and Risk Management

Hilda LarinaIna Ragragio
Microsoft Malware Protection Center

Tareq Saade
Microsoft Malware Protection Center

Andrei Florin Sayigo
Microsoft Malware Protection Center

Jireh Sanico
Microsoft Malware Protection Center

Richard Saunders (EMEA)
Microsoft Trustworthy Computing

Marc Seinfeld
Microsoft Malware Protection Center

Jasmine Sesso
Microsoft Malware Protection Center

Norie Tamura (GOMI)
CSS Japan Security Response Team

Gilou Tenebro
Microsoft Malware Protection Center

Patrik Vicol
Microsoft Malware Protection Center

Steve Wacker
Wadeware LLC

Jeff Williams
Microsoft Malware Protection Center

Dan Wolff
Microsoft Malware Protection Center

Table of Contents

Authors & Contributors . . . 3

About This Report . . . 5
Scope . . . 5
Reporting Period . . . 5
Conventions . . . 5

What Is a Botnet? . . . 6
History . . . 7
Botnets Today . . . 8
How Botnets Are Used . . . 12
How Botnets Work . . . 16
Botnet Commerce . . . 21

The Scope of the Problem: Botnet Data and Metrics . . . 23
Most Active Botnet Families in 2Q10 . . . 23
Where's Conficker? . . . 31
Operating System Statistics . . . 32
Geographic Statistics. . . . 33
Spam from Botnets . . . 35

Fighting Back Against Botnets . . . 37
Detecting Botnets . . . 37

Win32/Waledac and the Law: Fighting Botnets in Court . . . 40
Microsoft Digital Crimes Unit (DCU) . . . 40
A New Approach . . . 40
Why Waledac, Why Now? . . . 40
Technical Action Plan . . . 41
The Legal Action Plan . . . 46

Works Cited . . . 51

Worldwide Bot Infection Rates . . . 52

Appendix A: Threat Naming Conventions . . . 55

Appendix B: Data Sources . . . 57
Microsoft Products and Services . . . 57

Glossary . . . 59
Threat Families Referenced in This Report . . . 65

About This Report

Scope

The *Microsoft® Security Intelligence Report (SIR)* focuses on malware data, software vulnerability disclosure data, vulnerability exploit data, and related trends, as observed through data provided by a number of different Microsoft security-related products, services, and technologies. The telemetry used to produce this report was generated by millions of computers in more than 200 countries and regions on every inhabited continent. The ninth volume of the *Security Intelligence Report* also includes a special section, "Battling Botnets for Control of Computers," that provides an in-depth look at the botnet phenomenon, the threat it poses to people and computers worldwide, and steps that IT administrators and computer users can take to fight back. Past reports and related resources are available for download at www.microsoft.com/sir. We hope that readers find the data, insights, and guidance provided in this report useful in helping them protect their networks and users.

Reporting Period

In the current volume of the *Security Intelligence Report*, statistics about malware families and infections are reported on a quarterly basis, while other statistics continue to be reported on a half-yearly basis. In future volumes, Microsoft expects to report all statistics on a quarterly basis.

Throughout the report, half-yearly and quarterly time periods are referenced using the *n*H*yy* or *n*Q*yy* formats, respectively, where *yy* indicates the calendar year and n indicates the half or quarter. For example, 1H10 represents the first half of 2010 (January 1 through June 30), whereas 2Q10 represents the second quarter of 2010 (April 1 through June 30). To avoid confusion, always pay attention to the reporting period or periods being referenced when considering the statistics in this report.

Conventions

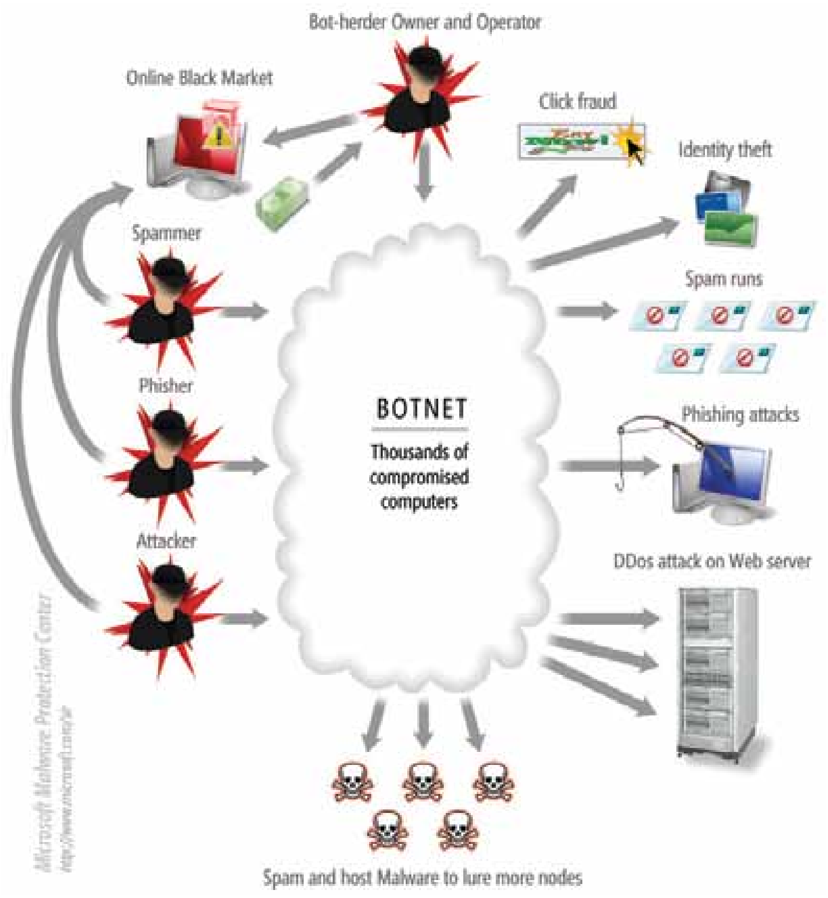
This report uses the Microsoft Malware Protection Center (MMPC) naming standard for families and variants of malware and potentially unwanted software. For information about this standard, see "Microsoft Malware Protection Center Naming Standard" on the MMPC website.

What Is a Botnet?

Since the early days of the public Internet, the word *bot* (from *robot*) has referred to automated software programs that perform tasks on a network with some degree of autonomy. Bots can perform many beneficial and even vital functions. For example, the Web crawling software programs used by popular search engines to index Web pages are a class of bots, and participants in the well-known SETI@HOME program (http://setiathome.berkeley.edu) voluntarily install bots on their computers that analyze radio telescope data for evidence of intelligent extraterrestrial life. Unfortunately, bots can also be developed for malicious purposes, such as assembling networks of compromised computers—*botnets*—that are controlled remotely and surreptitiously by one or more individuals, called *bot-herders*.

Computers in a botnet, called *nodes* or *zombies*, are often ordinary computers sitting on desktops in homes and offices around the world. Typically, computers become nodes in a botnet when attackers illicitly install malware that secretly connects the computers to the botnet and they perform tasks such as sending spam, hosting or distributing malware or other illegal files, or attacking other computers. Attackers usually install bots by exploiting vulnerabilities in software or by using social engineering tactics to trick users into installing the malware. Users are often unaware that their computers are being used for malicious purposes.

FIGURE 1. Example of a typical botnet in action


Bot-herder Owner and Operator
Online Black Market
Click fraud
Identity theft
Spammer
Spam runs
Phisher
BOTNET
Thousands of compromised computers
Phishing attacks
Attacker
DDos attack on Web server
Spam and host Malware to lure more nodes
Microsoft Malware Protection Center
http://www.microsoft.com/av

In many ways, a botnet is the perfect base of operations for computer criminals. Bots are designed to operate in the background, often without any visible evidence of their existence. Victims who detect suspicious activity on their computers are likely to take steps to find and fix the problem, perhaps by running an on-demand malware scan or by updating the signature files for their existing real-time malware protection. Depending on the nature of the bot, the attacker may have almost as much control over the victim's computer as the victim has, or perhaps more.

By keeping a low profile, bots are sometimes able to remain active and operational for years. The growth of always-on Internet services such as residential broadband has aided bot-herders by ensuring that a large percentage of the computers in the botnet are accessible at any given time. Botnets are also attractive to criminals because they provide an effective mechanism for covering the tracks of the botnet herder—tracing the origin of an attack leads back to the hijacked computer of an innocent user, which makes it difficult for investigators to proceed further.

In practice, many threats include limited command and control capabilities that are tailored to specific tasks, like downloading files, but do not provide the attacker with the kind of full-featured control that bots typically do. Malware authors also often add command and control capabilities to existing families as they develop them, so it is possible for malware families to evolve into botnets over time as new variants are released. For the purposes of this analysis, the *Security Intelligence Report* defines botnet as a network of computers that can be illicitly and secretly controlled at will by an attacker and commanded to take a variety of actions. Under this definition, a trojan downloader that is only designed to download arbitrary files and cannot otherwise be controlled by the attacker would not be considered a bot.

History

Many prevalent botnet families today have their roots in innocuous and beneficial utilities that were developed to manage Internet Relay Chat (IRC) networks. IRC is a real-time Internet chat protocol, designed for group (many-to-many) communication. IRC was designed to provide Internet users around the world with a casual way of communicating with each other in text–based discussion forums called *channels*.[1] A typical IRC network is comprised of a number of servers in various locations that connect users so that they can chat. Channels are administered by channel operators, who can take actions such as muting or ejecting unruly users. To extend the functionality of IRC, some channel operators used automated scripts—the original IRC bots—to perform functions such as logging channel statistics, running games, and coordinating file transfers.

As the popularity of IRC communities grew and the number of servers increased, so did the number of conflicts between users, which led to battles over the control of popular channels. IRC is structured so that when all of the designated channel operators disconnect from a channel, another member of the channel is automatically assigned as the new operator. In an effort to gain control of a channel, some malicious users created scripts

1 (Oikarnen n.d.)

that could perform denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks against IRC servers. By targeting the server used by a specific channel operator, these scripts could force the operator offline so that the attacker or someone else could gain operator status. Eventually, these same bots were used to target individual users.[2]

One of the earliest malware families to make use of IRC as a means of *command and control* (C&C) was a mass-mailing worm called Win32/PrettyPark, which appeared in 1999. After the PrettyPark bot infects a victim's computer, it secretly connects to a remote IRC server using its built-in client program and waits for instructions. An attacker serving as channel operator can command the bots to collect basic information about victims' computers, such as the operating system version and computer name, and user information such as sign-in names, email addresses, nicknames, and dial-up usernames and passwords. This technology was enhanced over the next few years, and a number of more sophisticated bots, such as Win32/AgoBot and Win32/Sdbot, emerged, which extended the basic functionality of the technology and added other attack methods. Much of the IRC–based C&C functionality of the PrettyPark worm is still seen in the current generation of IRC-based bots operating today.

Botnets Today

The botnet world is divided between bot families that are closely controlled by individual groups of attackers and bot families that are produced by *malware kits*. These kits are collections of tools, sold and shared within the malware underground, that enable aspiring bot-herders to assemble their own botnet by creating and spreading customized malware variants. Several malware kits are freely available for downloading and sharing; some have been published as open source code, which enables malware developers to create modified versions of the kits.[3] Other kits are developed by individual groups and sold like legitimate commercial software products, sometimes even including support agreements. For example, variants in the Win32/Zbot family are built from a commercial malware kit called Zeus; Win32/Pushbot bots are built from a kit called Reptile. The existence of botnet malware kits is one of the reasons why it is difficult for security researchers to estimate the number and size of botnets currently in operation. Detections of malware samples from a family like Zbot, for example, do not necessarily represent a single large botnet controlled by one individual or group, but instead may indicate an unknown number of unrelated botnets controlled by different people, some of which might encompass just a handful of controlled computers.[4]

Bot operators use several tactics to attack organizations, companies, and individuals in an effort to achieve their goals. Botnets typically exhibit a variety of behaviors based on the purpose of the attacks and the tools used to establish them. Being aware of and understanding the different attacking mechanisms can help IT and security professionals gain a deeper understanding of the nature of the botnet, the purpose behind it, and sometimes even the origin of the attack.

[2] (Canavan 2005, 5–6)
[3] (Bächer, et al. 2008)
[4] (Ollmann 2009, 3)

Bots, like other kinds of malware, can be spread in a number of different ways. Three common ways that computers are successfully compromised involve the following tactics:

- Exploiting weak or non-existent security policies.
- Exploiting security vulnerabilities.
- Using social engineering tactics to manipulate people into installing malware.

Some bots are designed to spread using these techniques directly, as worms; security researchers analyze the behavior of these self-replicating bots to learn more about how they spread. Other bots don't spread themselves directly, and are delivered by other malware families as payloads.

Many attackers and types of malware attempt to exploit weak or non-existent security policies. The most common examples of such exploits are attackers taking advantage of weak passwords and/or unprotected file shares. A threat that gains control of a user's account credentials could perform all of the actions the user is allowed to perform, which could include accessing or modifying resources as a local or domain administrator.

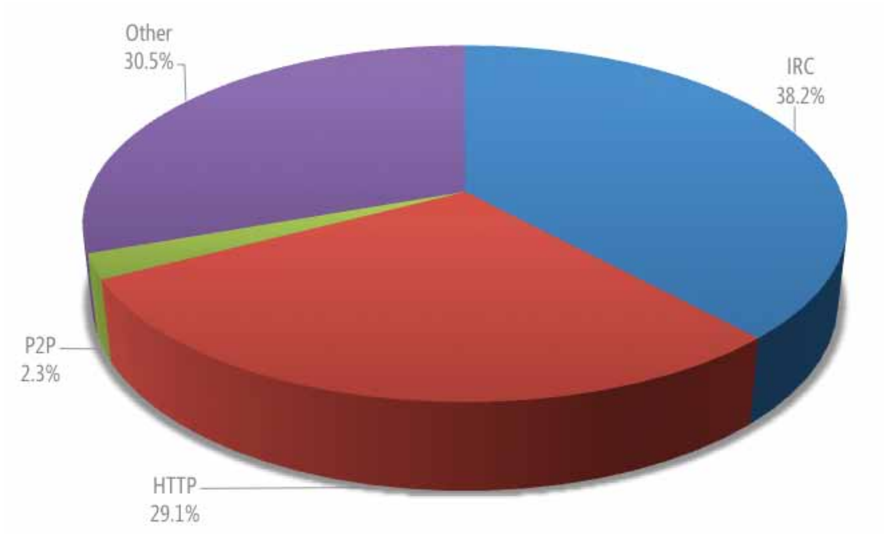
Other types of malware attempt to exploit security vulnerabilities to gain unauthorized access to computer systems. This type of attack is more successful on older operating systems than on newer systems that are designed with security as a core requirement. An analysis of infections reported by the Microsoft® Malicious Software Removal Tool (MSRT) during the second quarter of 2010 (2Q10) reveals that infection rates for computers around the world are significantly lower on newer versions of the Windows® operating system than on older versions. (See Figure 13 on page 32 for more information.)

Although these kinds of attacks remain a significant part of the threat landscape, improvements in software development practices and the increased availability and awareness of automatic software update mechanisms have greatly limited the kinds of technical exploit opportunities that are available to attackers. Instead, most attackers today rely heavily on *social engineering* techniques to mislead victims into unwittingly or even knowingly giving them information and access that would be much harder to take by force. Although media attention on social engineering attacks, such as phishing, have raised public awareness of this type of threat in recent years, attackers continue to find success with a variety of techniques for manipulating people. (See "Social Engineering as a Weapon" on page 15 of *Microsoft Security Intelligence Report, Volume 6 (July through December 2008)* for more information.)

IRC Botnets

The IRC protocol is used by many applications to support simple text–based chatting environments. Because the earliest bots were derived from benign IRC bots (and probably also because IRC has many legitimate uses), this protocol is still the most common C&C mechanism used by bots. As shown in the following figure, IRC–based families account for the largest share of the botnet–infected computers cleaned by Microsoft desktop anti-malware products in 2Q10, 38.2 percent.

FIGURE 2. C&C mechanisms used by botnet families in 2Q10, by number of unique computers reporting detections



Upon infection, the IRC clients built into the bot connect to a specified IRC server and channel like a typical chat client, and wait for instructions from the operator in the form of specially formatted text messages. Some of the more sophisticated bot operations have also encoded or encrypted bot commands in the channel topic, which is displayed to each client as it enters the channel. These commands can be complex enough to partition large botnets and give each subset its own task, which can be done based on country, network location, bot uptime, available bandwidth, and other variables. (See "How Botnets Work", beginning on page 16, for more information and examples.)

HTTP Botnets

The use of HTTP as a botnet C&C mechanism has increased in recent years as malware authors have moved beyond the first generation of malicious bots, although HTTP bots are still responsible for fewer infections than IRC bots. HTTP has the advantage of being the primary protocol for web browsing, which means that botnet traffic may be more difficult to detect and block. HTTP may be used to facilitate control either by having the bot sign in to a site that the bot controller operates, or by having the bot connect to a website on which the bot controller has placed information that the bot knows how to interpret as commands. This latter technique has an advantage in that the controller doesn't need to have an affiliation with the website. Some botnets even use blogs or social networking accounts for C&C, such as Win32/Svelta, a family discovered in 2009 that receives instructions from specially coded entries the attacker posts on the Twitter social networking service.

The HTTP protocol is also commonly used by bots to download updates and other malware, regardless of which C&C mechanism the bots use. Many bots include their own HTTP servers for hosting phishing Web sites or illegal content such as child pornography, or to provide an HTTP proxy that enables bot-herders to hide the location of their main (and usually illegal) websites.

Other Protocols

Modern botnets sometimes use other protocols and ports for command and control. These less common mechanisms may be implemented as a way to avoid detection by intrusion detection systems (IDS) and intrusion prevention systems (IPS).[5] In many scenarios these "custom protocols" simply involve transmitting binary or text data over a TCP or UDP port of the attacker's choosing, which is sometimes a port assigned to a different protocol or service.

P2P Command and Control

Historically, most large botnets have relied on a centralized control mechanism. In such scenarios, bots contact a preconfigured IRC channel or URL to receive commands directly from the bot-herder. Recently, a number of prevalent botnet families have adopted *peer-to-peer* (P2P) control mechanisms in an effort to evade scrutiny and resist shutdown. Although bot creators attempted to use P2P networking as a C&C mechanism for bots as far back as 2003, it has become more popular in recent years. Some botnet families use mechanisms derived from open source P2P implementations such as Kademlia. Others, including Win32/Waledac, use their own custom P2P implementations. (See "Win32/Waledac and the Law: Fighting Botnets in Court" on page 40 for more information about actions Microsoft has taken to bring the perpetrators of the Waledac botnet to justice.)

In a conventional botnet, the C&C server acts as a single point of failure; if it is taken offline or communications are disrupted, the entire botnet becomes inoperative. Some botnet families include redundancy features such as backup C&C servers, use of dynamic DNS services, and the ability for the bot-herder to redirect bots to a different server on demand, but the centralized control mechanism still introduces a measure of fragility to the system. Passing commands through intermediate peers in a P2P network makes it more difficult for analysts to identify bot controllers and determine the size of a network. This technique also makes the botnet more robust, in that any number of bots might be lost (in theory, at least) without fatally disrupting the botnet.

[5] (Nazario and Linden, Botnet Tracking: Techniques and Tools 2006)

How Botnets Are Used

Getting a botnet up and running is only the first step. A botnet can be used as a platform for a variety of criminal activities, depending on how the bot-herder chooses to configure the individual nodes. In addition to identity theft, botnets have many uses, some of which are described in the following subsections.

Spamming

Most of the spam that is sent today originates from botnets, which use several different techniques to get their unwanted messages past recipients' mail filters. In addition to renting out their botnets to spammers, bot-herders also use the botnets' spamming functionality themselves, sending out disguised copies of the bot malware (or hyperlinks to hosted copies of it) in an effort to increase the size of the network.

To understand how bots have come to play a central role for spam and phishing schemes, consider the typical life cycle of a spam or phishing attack. Attackers must first find a list of email addresses to target, and then they must craft their messages in a way that is likely to bypass email spam filters. They usually also host a landing page on which the product or service that is advertised in the message can be purchased. Bots can assist the attackers in all of these phases.

Attackers have traditionally found new potential victims by crawling the web or buying lists from other spammers. However, the email addresses obtained in this way are often of low quality—they might be old and no longer used, or the lists they buy might include trap accounts that notify the administrators of the receiving email system upon receiving a spam message so that the originating IP address can be quickly blocked.

Bots can be used to harvest high quality email addresses. For example, the HTTP botnet family Win32/Waledac searches through many different kinds of files on fixed and remote drives on compromised computers looking for addresses. In addition to sending spam to the harvested addresses directly, Waledac also transmits the addresses to a list of remote websites, presumably for the attacker to retrieve. The market for lists of email addresses is well-established, and bot owners can easily turn the lists they harvest this way into profit by selling them to spam and phishing attackers. Addresses that include demographic information, such as name and address, or targeting information, such as the name of the bank the person uses, command a premium. Bots can steal very specific information from computers, which makes them especially useful for *spear phishing*, a type of phishing attack that targets the employees or customers of a particular institution.

Spambots, as bots that send spam are sometimes called, also give attackers access to tens of thousands of computers or more that can be used to originate spam. Typically, a prospective spammer contacts a bot-herder to rent the services of a botnet. Several modern botnet families are designed to be partitioned into segments that can be controlled separately. Partitioning allows the bot owner to control how much capacity to rent at a time, prevents valuable parts of the botnet from being used without their permission, and segments the market by collecting higher-quality bots (newly compromised, high-bandwidth, rarely rebooted) together and renting them for a premium.

After the spammer and bot controller negotiate the product and price, the bot controller instructs each of the selected bots to start a proxy server, and typically provides the spammer with access to a webpage that lists the IP addresses of the selected bots and the ports on which they are running their proxy servers. These bots go online and offline as the compromised computers' real owners reboot them; IP addresses might change, and the ports on which the proxy servers run might change over time. Every time there is a change, the bot notifies its C&C server, which automatically updates the list of IP addresses the spammer can use.

Phishing

Phishing is a method of credential theft that tricks Internet users into revealing personal or financial information online. Attackers send messages purporting to be from a trusted institution, such as a bank, auction site, online game, or other popular website. These phishing messages—which are often generated and sent by bots, like spam—direct victims to webpages run by the attackers, where they are instructed to submit private information such as login credentials or credit card details. Some malware families, such as Win32/Pushbot, have also been observed to redirect URLs for banking websites in the victim's browser directly.

Stealing Confidential Data

Many bots can be commanded to search a computer's hard disks for personal information, including computer authentication credentials, bank account numbers and passwords, product keys for popular computer games and other software products, and other sensitive data. Some of the earliest malicious bots included the ability to transmit product keys back to the attacker, and such features are common in kit–based bots such as Win32/Rbot and Win32/Zbot.

Perpetrating Distributed Denial-of-Service (DDoS) Attacks

One of the oldest attacker uses of botnets is as a mechanism for launching distributed denial-of-service (DDoS) attacks, as discussed earlier. A denial-of-service (DoS) attack is an attempt to make a computer resource, such as network connectivity and services, unavailable to its intended users. Typically, such attacks flood the resource with network traffic, which saturates its bandwidth and renders it unavailable to perform legitimate services. A DDoS attack involves multiple computers—such as those in a botnet—attacking a target at the same time, making it harder to defend against than an attack from a single computer. Several common bots, including Win32/Hamweq, Win32/Pushbot, and Win32/Waledac, have been observed participating in DDoS attacks.

Bot-herders usually target high-profile websites or companies, but any service available on the Internet can be a victim, including corporate, government, education, and private computer systems. However, as with most malware in the modern era, the motive is usually financial—attackers use the threat of a DDoS attack to extort money from the target. There have been numerous reports of DDoS attacks against public commercial websites and even Internet infrastructure components such as DNS root servers. The potential impact of DDoS can be very significant. As botnets grow larger, their ability to launch DDoS attacks grows more powerful, with one recent attack involving sustained traffic in excess of 7.3 Gbps over a 10Gbps link.

Installing Malware and Potentially Unwanted Software

Bot-herders often use their botnets to download additional malware to victims' computers to reap additional profits. Early botnets often focused on installing adware, spyware, and other potentially unwanted software in an effort to earn quick profits. In a typical incident in 2005, a bot-herder in California used the bot family Win32/Rbot to install adware on more than 20,000 computers as part of a pay-per-click advertising scheme that brought in more than U.S. $50,000, according to the U.S. Department of Justice.[6]

Malware installed by botnets often works silently to avoid tipping off the victim that the computer is infected, but not always. Some botnets, including Win32/Waledac, have been observed to download *rogue security software*—programs that masquerade as legitimate anti-malware products, displaying false alerts about nonexistent infections on the victim's computer and offering to remove them if the victim pays for the "full version." Botnets have also been observed downloading packet sniffers and additional downloaders. Some botnets are even instructed to download and install other bots. Variants of Win32/Hamweq have been observed to download Win32/Rimecud, a botnet family with more sophisticated backdoor features.

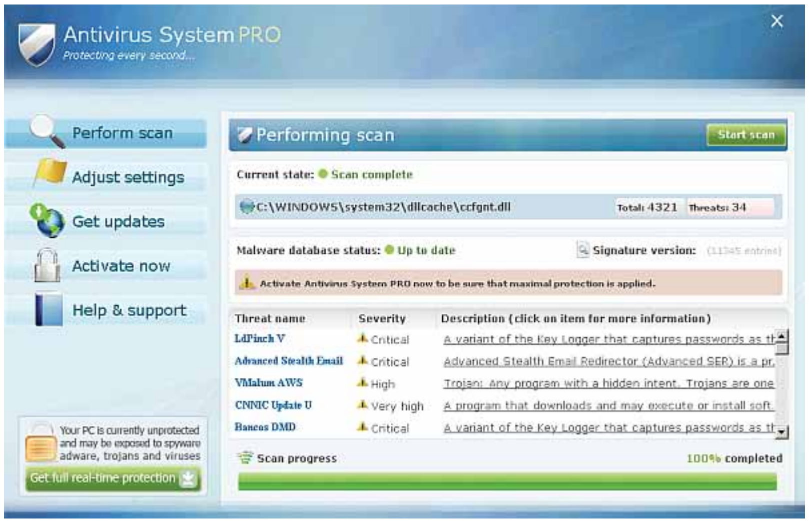


FIGURE 3. Win32/FakeSpypro, a rogue security software family downloaded by Win32/Waledac

Current state: Scan complete

C:\WINDOWS\system32\dllcache\ccfgnt.dll Total: 4321 Threats: 34

Malware database status: Up to date Signature version: (11345 entries)

Activate Antivirus System PRO now to be sure that maximal protection is applied.

Threat name	Severity	Description (click on item for more information)
LdPinch V	Critical	A variant of the Key Logger that captures passwords as th
Advanced Stealth Email	Critical	Advanced Stealth Email Redirector (Advanced SER) is a pr.
VMalum AWS	High	Trojan: Any program with a hidden intent. Trojans are one
CNNIC Update U	Very high	A program that downloads and may execute or install soft.
Bancos DMD	Critical	A variant of the Key Logger that captures passwords as th

Your PC is currently unprotected and may be exposed to spyware adware, trojans and viruses

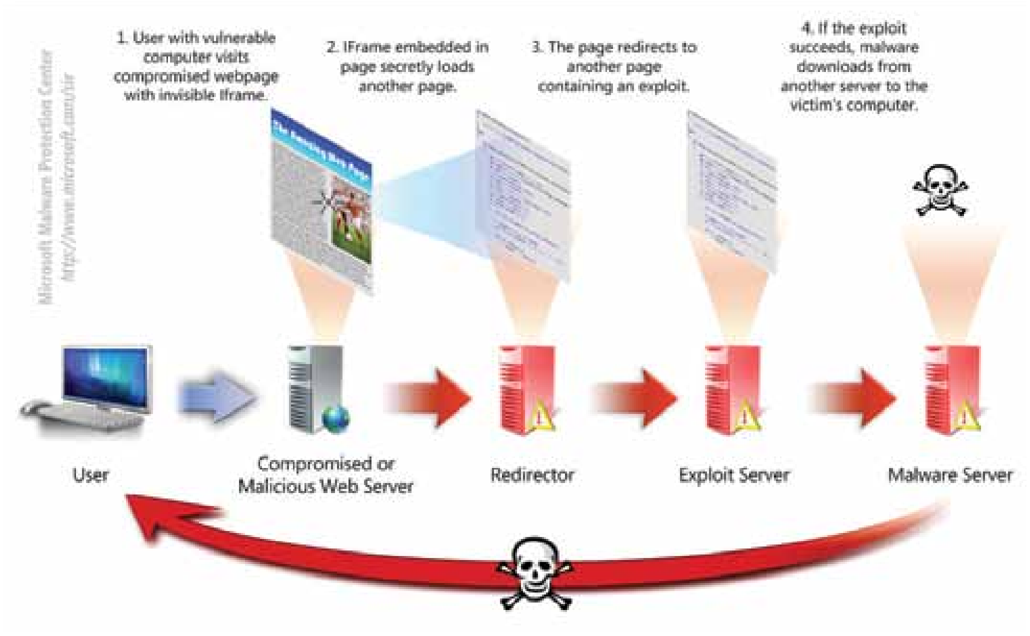
Scan progress 100% completed

[6] (United States of America v. Jeanson James Ancheta 2005)

Distributing Malware

Botnets often play important roles in malware distribution schemes. In a typical scenario, an attacker uses bots to send spam messages that contain links to malware, which itself is often hosted by the botnet. The messages use social engineering techniques to convince recipients to click the links, such as disguising the message as a news digest with provocative-sounding fake headlines, or as a message from a friend purporting to offer a link to an embarrassing photo of the recipient. The malware is either offered for download directly, as a disguised executable file, or is hosted on a webpage that includes exploits that are designed to use specific browser vulnerabilities to secretly install malware on visitors' computers (a tactic sometimes called *drive-by downloading*). (See "Analysis of Drive-By Download Pages") in the Reference Guide section of the *Security Intelligence Report* website for more information.

FIGURE 4. One example of a drive-by download attack


1. User with vulnerable computer visits compromised webpage with invisible Iframe.
2. IFrame embedded in page secretly loads another page.
3. The page redirects to another page containing an exploit.
4. If the exploit succeeds, malware downloads from another server to the victim's computer.
Microsoft Malware Protection Center
http://www.microsoft.com/sir
User
Compromised or Malicious Web Server
Redirector
Exploit Server
Malware Server

Occasionally an attacker sends malware directly to recipients as a file attachment, although most popular email programs and services block users from downloading actual or suspected malicious files. (See "Email Threats" in the Reference Guide section of the *Security Intelligence Report* website for more information.)

Click Fraud

Much advertising on the Internet operates on a "pay-per-click" model, in which the operator of a website that contains advertisements receives a fee from the advertiser every time a visitor clicks on the ad. Criminals sometimes use botnets to generate fraudulent "clicks" on pay-per-click advertisements. For example, a bot-herder may set up a website with advertisements and negotiate an arrangement with a web host that pays the bot herder for clicks on their ads. The bot-herder then writes bot code that automates the click-throughs so that the bots in the botnet, which can number in the thousands, instantly click on the ads on that website. The pay-per-click marketer, honoring the advertisement agreement, pays for these illegitimate clicks without gaining the sales or leads that were hoped for.[7] This process can be further enhanced if the bot hijacks the start page of a compromised computer so that the click-throughs are executed each time the owner of the compromised computer opens their browser, victimizing both the owner of the compromised system and the web host.[8]

How Botnets Work

Most botnets are similar in the details of their operation, whether they are created specifically for the benefit of one criminal organization or built from a kit by a single attacker or small group. The details given in this section apply most directly to "traditional" IRC–controlled botnets built from kits, but much of this information is also relevant to other kinds of botnets.

Botnet Creation

The first step for the prospective botnet operator is acquiring the necessary software. Professional criminal organizations with the resources to build the largest botnets may create or commission their own proprietary software, but most prospective bot-herders are likely to seek out software and services from an online black market forum (for more information, see "Botnet Commerce" on page 21). This approach doesn't require a great deal of technical sophistication, and in fact some researchers have observed the programming skills of low-level bot-herders to be quite limited.[9] With the tools and assistance available on the black market, however, prospective bot-herders have access to all the information they need to create, maintain, and profit from a botnet.

Unless the botnet software uses a P2P control mechanism, which is still relatively rare, bot-herders must also find a home for the C&C server. A bot-herder who has access to a compromised computer in advance may install the IRC server software on it for C&C use. Alternatively, the bot-herder might open an account with a "bulletproof" provider that is resistant to efforts to disconnect lawbreakers. Other choices include establishing secret channels on public IRC servers (inexpensive, but risky) or setting up servers on their own computers (which gives them the most control, but can be expensive, and also risks termination by their upstream providers).

[7] (United States of America v. Jeanson James Ancheta 2005)
[8] (Bächer, et al. 2008)
[9] (Bächer, et al. 2008)

Most bot-herders choose to register domain names for hosting their C&C servers. Although it is technically possible to control a botnet without a domain name by configuring its bots to connect directly to the IP address of the server, this approach has significant disadvantages. If bot-herders find it necessary to quickly move a server to a different provider to avoid detection or in response to a termination of services, they might not have time to reconfigure the bots to connect to a new IP address, and will lose control of them entirely. Bot-herders might choose to register domain names directly with one of the many registrars around the world, or open accounts with a *dynamic DNS* service, which provides stable hostnames for resources that change IP addresses frequently. Using a domain name incurs a risk that the domain's DNS provider might terminate service to the server, which is one reason P2P mechanisms have become more popular recently.

The server software for some bots can be quite complex, offering functionality such as geographic segmentation and task maintenance, although most kit–based bot servers are relatively simple and can be controlled through an ordinary IRC client or web browser. Bot-herders often use the same IRC server packages as legitimate IRC operators, with open source programs like UnrealIRCd being among the most popular. IRC server software is often minimized and modified by the botnet owner or the kit developer. Common modifications include removing JOIN, PART, and QUIT messages on channels to avoid unnecessary traffic. In addition, the functionality provided by the WHOIS (information about specific users), WHO (host details about specific users), LUSERS (information about number of connected clients), and RPL_ISUPPORT (information about the features the server supports) commands is removed to hide the identity of the bots that join the channel and to conceal the size of the botnet from unauthorized people who connect to the IRC server.

In an effort to block unauthorized people such as security researchers and rival bot-herders from entering the channel and seizing control of the bots, botnet owners typically secure the channel using standard IRC commands such as the following:

- `/mode #channel +k [password]`. This command password-protects the channel. The bot client must be configured to supply the correct password when attempting to access the channel. (Some herders choose to password-protect the whole server as well.)
- `/mode #channel +q`. This command marks the channel as quiet. System messages such as JOINs, PARTs, and nickname ("nick") changes are not broadcast, which makes it appear to each client as if that client is the only one on the channel.
- `/mode #channel +s`. This command marks the channel as secret, so it will not appear in channel listings. Users who are outside the channel will not be able to discover the names of the channel participants.
- `/mode #channel +t`. This command locks the channel topic so only channel operators can change it. Channel topics are often used to send commands to bots as they enter the channel.

- `/mode #channel +u`. This command puts the channel into auditorium mode, in which channel operators are the only participants who can see the names of all of the clients connected to the channel. Along with `+q`, this mode makes it difficult for investigators and others to measure the size of the botnet.

After the server and channel are set up, bot-herders can build and distribute the bots. Technically sophisticated bot-herders might choose to code their bots themselves; others build bots using malware creation kits, or simply hire someone to do it for them on the black market. For IRC botnets, bot-herders configure the bots with the name or IP address of the server to connect to, the channel name, and any passwords they will need to connect.

Controlling the Botnet

After a bot infects a computer, it attempts to contact its C&C server for instructions. A typical communication that can be observed after a successful infection might look like the following excerpt:

```
<- :irc.XXX.XXX NOTICE AUTH :*** Looking up your hostname...

<- :irc.XXX.XXX NOTICE AUTH :*** Found your hostname

-> PASS s3rv3rp455

-> NICK [d1f]-511202

-> USER hlxahl 0 0 :hlxahl

<- :irc.XXX.XXX NOTICE [d1f]-511202 :*** If you are having prob-
lems connecting due to ping timeouts, please type /quote pong
SF125722 or /raw pong SF125722 now.

<- PING :SF125722

-> PONG :SF125722

<- :irc.XXX.XXX 001 [d1f]-511202 :Welcome to the irc.XXX.XXX IRC
Network [d1f]-511202!hlxahl@heh

<- :irc.XXX.XXX 002 [d1f]-511202 :Your host is irc.XXX.XXX, run-
ning version Unreal3.2.7

<- :irc.XXX.XXX 003 [d1f]-511202 :This server was created Mon Sep
10 2007 at 20:30:33 PDT

<- :irc.XXX.XXX 004 [d1f]-511202 irc.XXX.XXX Unreal3.2.7 iowghraA-
sORTVSxNCWqBzvdHtGp lvhopsmntikrRcaqOALQbSeIKVfMCuzNTGj
```

After connecting, the bot tries to join the operator's channel as configured:

```
-> JOIN #[d1f] channelpassword

-> MODE [d1f]-511202 +iwx
```

If the topic does not contain any instructions for the bot it remains idle in the channel, awaiting commands.

To control the bots, bot-herders enter the channel like ordinary IRC users and issue specially formatted commands. With some commands, such as commands to collect and report information about the victim's computer, the bots report their results as chat messages within the IRC channel, or save them locally as files that the herder can retrieve later. Depending on the capabilities of the bot malware, bot-herders can execute a wide range of actions, as described in "How Botnets Are Used" on page 12. A brief selection of typical botnet commands, in this case from the Win32/Rbot family, provides an idea of the kinds of operations a herder can execute:

- **.capture**. Generates and saves an image or video file. Depending on the parameters used, this file could be a screenshot of the victim's desktop or a still image or video from the victim's webcam. The operator can recover the saved picture using the **.get** command.
- **.ddos.syn, .ddos.ack, .ddos.random**. Launches a DDoS attack on a specified IP address for a specified length of time.
- **.download**. Downloads a file from a specified URL to the victim's computer and optionally executes it.
- **.findfile**. Searches for files on the victim's computer by name and returns the paths of any files found.
- **.getcdkeys**. Returns product keys for software installed on the victim's computer.
- **.keylog**. Logs the victim's keystrokes and saves them to a file.
- **.login, .logout**. Authenticates the bot-herder with the bots. Before issuing commands to any bots in the channel, the bot-herder must use the **.login** command with a password that is specified in the bots' configuration data so the bots will recognize the bot-herder as an authorized controller.
- **.open**. Opens a program, an image, or a URL in a web browser.
- **.procs**. Lists the processes running on the victim's computer. Other commands can then be used to kill processes by name or ID.

Spreading Bots

Some bots include worm functionality for spreading themselves through exploits, a mechanism that early malicious botnets used widely. Today, however, vulnerabilities that are conducive to worm activities are rare and herders rely heavily on social engineering to distribute malware to victims. One leading method involves distributing infected files on P2P networks, purportedly as pirated copies of software or films. Another way is through drive-by downloads in which the attacker hosts a webpage (or compromises a legitimate one) with malicious code that downloads the bot malware to vulnerable computers that visit.

After the botnet is up and running, it can be used to attack and infect additional computers. Bot-herders can designate a few nodes as malware servers, using various techniques to disguise their locations and to provide protection in case one or more of them are discovered and shut down. Other nodes can be used to send spam with links to exploit-laden pages on the malware servers, using various forms of social engineering to lure recipients to click the link in the message. Committed bot-herders can use these techniques to build networks of thousands of compromised computers over time.

Defending the Botnet

Bot creators use many techniques to prevent bots from being detected or removed. Many bots contain rootkit components, which are designed to be hard to remove. Bots that enter a computer by exploiting a vulnerability also sometimes "fix" the vulnerability after infection to prevent other malware from exploiting the same vulnerability and interfering with the bot.

Many bot creators package their software using *packers*, software utilities that compress and obfuscate binary code. Packers are designed to make software more difficult to reverse-engineer so that malicious programmers, or *crackers*, cannot make unauthorized modifications to it. Malware authors often use packers both to protect their own code and to evade detection by anti-malware software. Bots are often packaged using many of the same packers used by legitimate software publishers, such as the open source program UPX and commercial utilities such as Silicon Realms' Armadillo and Oreans Technologies' Themida.

Scripting languages used to author many common bots are modular enough to support various encoding and obfuscation techniques to hide the original source from other malware authors and the anti-malware industry. There are many documented techniques, ranging from simple string manipulation functions to uuencode or MIME encoding techniques.

Bot creators sometimes use *polymorphism* in an effort to make it more difficult for antivirus software to identify and remove bots. Polymorphism results in malware files that are functionally identical but differ from one another in file size, content, or other respects. There are two general types of polymorphism that malware creators use:

- **Server-side polymorphism**, in which a server is configured to serve a slightly different version of a file every time it is accessed, possibly by changing the file name of a component to a new random value, or by encrypting or compressing it in a slightly different way.
- **Malware polymorphism**, in which the malware itself is designed to change slightly every time it replicates.

Before distributing a new bot or variant, a bot-herder may have it scanned by popular anti-malware products and services to see if any of them successfully detect it as malware. A number of free online services allow users to upload files to be scanned by multiple anti-malware engines. Legitimate services of this nature share the malware samples they receive with security software vendors to help them improve their detection signatures, so a malware author who uses one of these services to test a new family or variant is likely to guarantee a limited active lifespan for it, even if it initially goes undetected. Malware authors can also use private online testing services that do not share samples with vendors, which can prolong the amount of time a variant remains active before being detected by major anti-malware products.

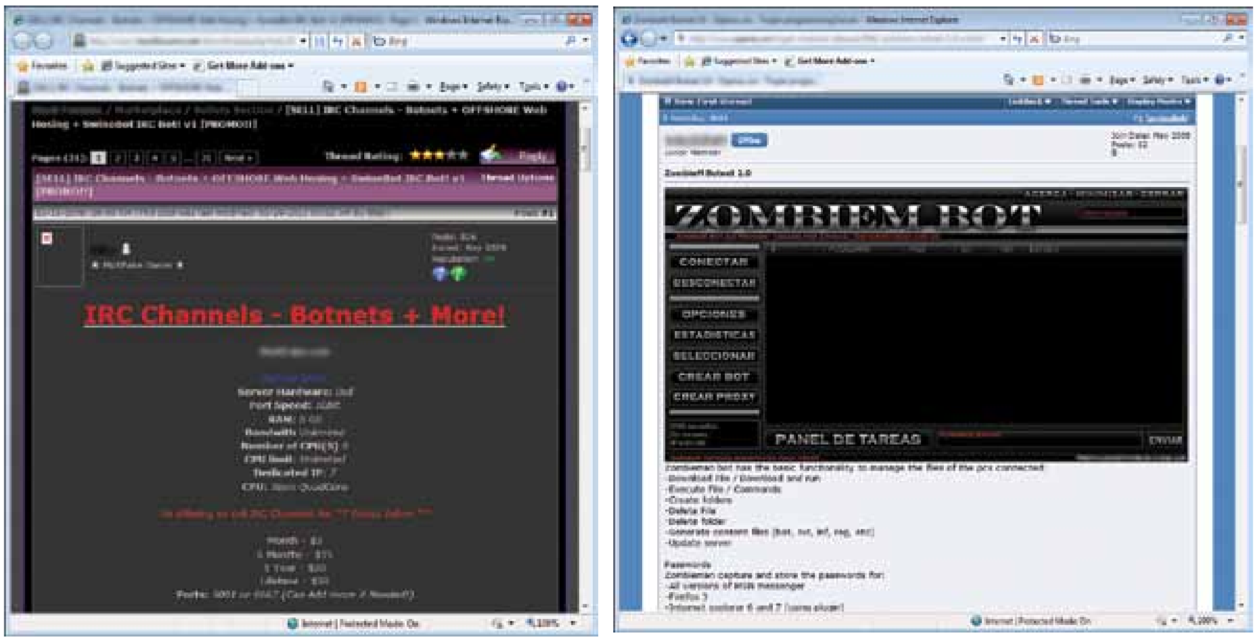
Botnet Commerce

Attackers have developed a number of different ways to make money with botnets, as described in "How Botnets Are Used" beginning on page 12. Bot-herders might choose to attempt these activities individually, or they might return to a black market forum and advertise the services of the new network. These black market forums are online communities that bring sellers of malware and services together with interested buyers. These communities have their roots in the virus exchange (VX) forums that operated in the late 1990s and early 2000s, when the malware scene was dominated by amateur hobbyists who created and distributed malware as a means of raising their status in such communities. The hobbyists were eventually displaced by professional criminals, and today malware on the Internet is almost entirely a profit-oriented enterprise.[10]

Black market sites often offer a comprehensive selection of products and services related to botnets and other malware. Sellers may offer malware kits that enable prospective bot-herder to build their own botnets. Existing botnet owners may rent their networks to spammers and attackers, or sell collections of bots to new owners. Other sellers offer lists of vulnerable IP addresses, "bulletproof" hosting for C&C servers that are supposedly less likely to be taken down by law enforcement and upstream providers, utilities such as packers and encryptors that they claim render malware undetectable by current antivirus signatures, and more. Building a botnet typically involves several different components—the bot itself, the server component, other malware such as downloaders and rootkits to use as payloads, and so on—so black market sites greatly simplify the process of assembling the necessary tools. New sellers with no established reputation might offer potential buyers a free trial in the form of access to the botnet for a very short time, typically less than an hour. As with many mainstream businesses, a few successful transactions give a seller a good reputation in the community, which in turn leads more buyers to bot-herders' virtual doors.

[10] (Fogie 2006)

FIGURE 5. Criminals sell botnet software and services in online black markets



[SELL] IRC Channels - Botnets + OFFSHORE Web Hosting + Swissbot IRC Bot! v1 [PROMO!!]

IRC Channels - Botnets + More!

Server Hardware: [illegible]
Port Speed: [illegible]
RAM: [illegible]
Bandwidth: Unlimited
Number of CPU(S): [illegible]
CPU Instl: Unlimited
Dedicated IP: [illegible]
CPU: Xeon QuadCore

Month - [illegible]
Ports: [illegible]

ZombieM Botnet 1.0

ZOMBIEM BOT

Zombiem bot has the basic functionality to manage the files of the pcs connected
-Download File / Download and run
-Execute File / Commands
-Create folders
-Delete File
-Delete folder
-Generate content files (bat, txt, inf, reg, etc)
-Update server

Passwords
Zombiem capture and store the passwords for:
-All versions of MSN messenger
-Firefox 3
-Internet explorer 6 and 7 (uses plugin)

Black markets often feature many of the trappings of legitimate e-commerce, such as "verified sellers" whom the site operators assert are trustworthy, and even round-the-clock technical support for purchasers. Perhaps owing to the nature of the commerce involved, such marketplaces tend to be rife with swindlers, and buyers frequently complain about "rippers" who took their money but didn't deliver the products or services ordered.

Some of the goods on sale in these markets, especially up-to-date malware creation kits, can be expensive. However, the rates bot-herders and malware authors can charge for access to their wares are usually quite low, at least for renters in wealthy, developed nations. Adam Sweaney, a U.S.-based bot-herder who pled guilty in September 2007 in U.S. District Court to a one-count felony violation for conspiracy, fraud, and related activity in connection with computers, was caught after he offered an undercover federal investigator access to more than 6,000 compromised computers for just U.S.$200 per week.[11]

Black markets are frequently associated with online communities of bot-herders and malware authors, who come together to share techniques and stories, like many groups of people with shared interests. Such forums are often fairly friendly to novices, with experienced malware authors writing tutorials and answering questions posed by newcomers. Such support networks make it relatively easy and convenient for a novice bot controller to get started, although gaining real competence probably still takes a while.[12] This might change if vulnerable computers become more of a scarce resource, or if the marginal reward for a new bot increases significantly for some reason, thus increasing the competition for resources.

[11] (United States of America v. Adam Sweaney 2007)
[12] (Nazario and Linden, Botnet Tracking: Techniques and Tools 2006)

The Scope of the Problem: Botnet Data and Metrics

It is difficult to measure with any certainty the numbers of bots and botnets in existence, and estimates from botnet researchers can vary by an order of magnitude or more. Counting the number of bot-infected computers found and cleaned by antivirus software can sometimes yield figures that are very different from estimates produced by researchers who concentrate on the effects of botnets, such as the amounts and origins of spam and the number of known active C&C servers. There is no widespread agreement about which methods are best for estimating the size of botnets. The information presented in this section is intended as a straightforward presentation of telemetry data produced by Microsoft tools and services, and should not be taken as making or supporting any particular estimates of botnet size and scope.

For information and guidance about defending computers and networks against botnet infection, see the Managing Risk section of the *Security Intelligence Report* website.

Most Active Botnet Families in 2Q10

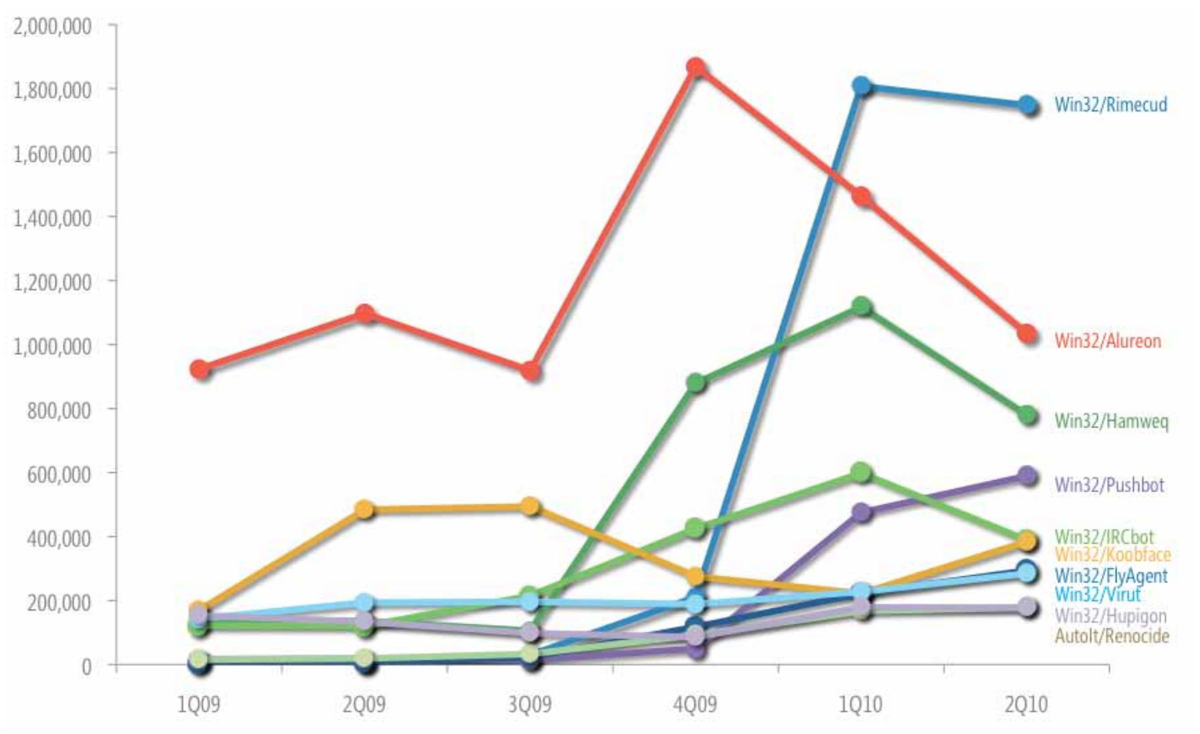
Microsoft anti-malware products and utilities include detection signatures for many individual bot families, and the number continues to grow as malware creators pursue efforts to evade detection and create more effective botnets.

Microsoft desktop anti-malware products removed bots from 6.5 million computers around the world in 2Q10. Figure 6 shows the top 25 bot families.

FIGURE 6. Top 25 bot families detected by Microsoft desktop anti-malware products worldwide, 1Q10-2Q10, by number of computers cleaned

Rank	Family	Primary Control Mechanism	Computers Cleaned (1Q10)	Computers Cleaned (2Q10)	Change
1	Win32/Rimecud	Other	1,807,773	1,748,260	-3.3% ▼
2	Win32/Alureon	HTTP	1,463,885	1,035,079	-29.3% ▼
3	Win32/Hamweq	IRC	1,117,380	779,731	-30.2% ▼
4	Win32/Pushbot	IRC	474,761	589,248	24.1% ▲
5	Win32/IRCbot	IRC	597,654	388,749	-35.0% ▼
6	Win32/Koobface	HTTP	222,041	383,633	72.8% ▲
7	Win32/FlyAgent	HTTP	221,613	293,432	32.4% ▲
8	Win32/Virut	IRC	227,272	284,519	25.2% ▲
9	AutoIt/Renocide	IRC	167,041	178,816	7.0% ▲
10	Win32/Hupigon	Other	178,706	177,280	-0.8% ▼
11	Win32/Sdbot	IRC	125,466	146,922	17.1% ▲
12	Win32/Nuwar	P2P	8,098	133,951	1554.1% ▲
13	Win32/Bubnix	HTTP	91,144	132,771	45.7% ▲
14	Win32/Zbot	HTTP	107,363	131,078	22.1% ▲
15	Win32/Ursap	IRC	121,239	121,302	0.1% ▲
16	Win32/Rbot	IRC	145,107	110,316	-24.0% ▼
17	Win32/Pasur	Other	95,040	91,612	-3.6% ▼
18	Win32/Rustock	HTTP	82,712	52,312	-36.8% ▼
19	Win32/Slenfbot	IRC	56,898	51,228	-10.0% ▼
20	Win32/Bagle	Other	48,326	34,240	-29.1% ▼
21	Win32/Tofsee	HTTP	29,367	32,031	9.1% ▲
22	Win32/Bifrose	Other	28,966	30,466	5.2% ▲
23	Win32/Waledac	P2P	83,580	29,816	-64.3% ▼
24	Win32/Prorat	Other	26,913	25,726	-4.4% ▼
25	Win32/Trenk	Other	24,093	21,749	-9.7% ▼

FIGURE 7. Top 10 bot families detected by Microsoft desktop anti-malware products worldwide in 2Q10, by quarter since 1Q09



Win32/Rimecud

Win32/Rimecud was the most commonly detected bot family in 2Q10, with 68.9 percent more detections than the next most common family. Detections for Rimecud were added to the MSRT in January 2010.

Rimecud is a "kit" family: different people working independently use a malware creation kit to create their own Rimecud botnets. (See "Botnets Today" on page 8 for more information about malware creation kits.) Rimecud is the primary malware family behind the so-called Mariposa botnet, which infected millions of computers around the world in 2009 and 2010. In July of 2010, the Slovenian Criminal Police arrested a 23-year-old Slovenian citizen suspected of writing the malware code, following the February 2010 arrests of three suspected Mariposa botnet operators by the Spanish Guardia Civil.[13]

Rimecud is a backdoor worm that spreads via fixed and removable drives, and by sending malicious hyperlinks to a victim's contacts via several popular instant messaging programs. Rimecud can be commanded to take a number of typical botnet actions, including spreading itself via removable drives, downloading and executing additional malware, and stealing passwords.

[13] (Federal Bureau of Investigation 2010)

Rimecud is somewhat unusual in that it uses its own UDP-based protocol and dynamically configurable port for command and control functions, rather than a standard IRC- or HTTP-based mechanism like most bots.

Win32/Rimecud uses a variety of obfuscators to hinder detection. These obfuscators typically use virtual environment detection and anti-emulation tricks to make the malware harder to detect.

As bots often install other malware as payloads, or are installed by other malware as payloads themselves, it is not uncommon for anti-malware utilities to detect multiple malware families on a single infected computer. Figure 8 lists the other threats that were most often detected on computers infected with Win32/Rimecud in 2Q10. For example, 36.9 percent of the computers that were infected with Rimecud were also infected with Win32/Autorun, a significant correlation. Like Rimecud itself, four out of the five families on the list are worms, which illustrates how vulnerabilities and unsafe practices can put a computer at risk of infection by multiple families that use similar infection vectors.

FIGURE 8. Other threats found on computers infected with Win32/Rimecud in 2Q10

Other Family	Most Significant Category	Percentage of Win32/Rimecud-Infected Computers
Win32/Autorun	Worms	36.9%
Win32/Conficker	Worms	23.2%
Win32/Taterf	Worms	15.6%
Win32/Sality	Worms	15.3%
Win32/VBInject	Miscellaneous Potentially Unwanted Software	10.1%

For more information about Win32/Rimecud, see the following posts at the MMPC blog (http://blogs.technet.com/mmpc):

- Rimecud and Hamweq – birds of a feather (January 12, 2010)
- Win32/Rimecud: MSRT's success story in January 2010 (January 19, 2010)
- In focus: Mariposa botnet (March 4, 2010)

Win32/Alureon

Win32/Alureon was the second most common bot family detected by Microsoft anti-malware products in 2Q10. Detections for Alureon were first added to the MSRT in March 2007.

Alureon is a large family of data-stealing trojans, some variants of which include bot components that use HTTP for command and control. Alureon variants typically perform such actions as stealing confidential information, downloading and executing arbitrary files, and redirecting URLs for popular search engines.

Some Alureon variants include a rootkit component, which has caused problems for some infected users when they apply security updates. In February 2010, a number of infected computers experienced repeated “blue screen” stop errors caused by the Alureon rootkit after Microsoft Security Bulletin MS10-015 was installed. The Microsoft Security Response Center (MSRC) subsequently addressed the issue with a revised version of the update that notifies Alureon-infected users of the problem and helps them resolve it.

For more information, see the following posts at the MMPC blog (http://blogs.technet.com/mmpc):

- MSRT April Threat Reports & Alureon (April 30, 2010)
- MSRT May Threat Reports and Alureon (May 21, 2010)

For more information about the MS10-015 issue, see the following post at the MSRC blog (http://blogs.technet.com/msrc):

- Update - Restart Issues After Installing MS10-015 and the Alureon Rootkit (February 17, 2010)

Win32/Hamweq

Win32/Hamweq was the third most common bot family detected by Microsoft anti-malware products in 2Q10. Detections for Hamweq were added to the MSRT in December 2009.

Hamweq shares some similarities with Win32/Rimecud, the most prevalent bot family in 2Q10, although Rimecud supports a more diverse and sophisticated set of commands than Hamweq. Both include IRC–controlled backdoors and spread via removable volumes, instant messaging, and P2P networks. Both families also inject code into the explorer.exe process and use the Recycle Bin as the target drop folder for copies of themselves.

Worms that spread via removable drives often spread with relative ease in enterprise environments because of the widespread use of such devices, and Hamweq is no exception. Variants of Hamweq have been observed to download other malware to infected computers, including Rimecud. Some Hamweq variants are also used to perform DDoS attacks.

FIGURE 9. How the Win32/Hamweq bot family works

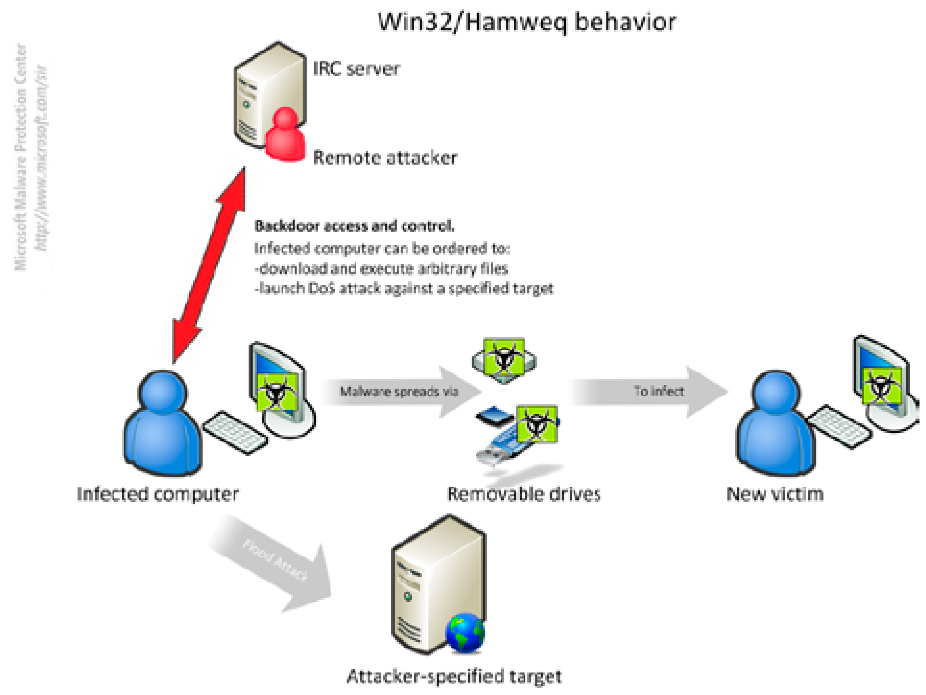

Win32/Hamweq behavior
Microsoft Malware Protection Center
http://www.microsoft.com/sir
IRC server
Remote attacker
Backdoor access and control.
Infected computer can be ordered to:
-download and execute arbitrary files
-launch DoS attack against a specified target
Malware spreads via
To infect
Infected computer
Removable drives
New victim
Flood Attack
Attacker-specified target

Figure 10 lists the other threats that were most often detected on computers infected with Win32/Hamweq in 2Q10.

FIGURE 10. Other threats found on computers infected with Win32/Hamweq in 2Q10

Other Family	Most Significant Category	Percentage of Win32/Hamweq-Infected Computers
Win32/Autorun	Worms	44.3%
Win32/Rimecud	Worms	34.1%
Win32/Conficker	Worms	28.2%
Win32/Taterf	Worms	23.0%
Win32/Sality	Worms	15.6%

For more information about Win32/Hamweq, see the following posts at the MMPC blog (http://blogs.technet.com/mmpc):

- MSRT slices the Hamweq for Christmas (December 8, 2009)
- Rimecud and Hamweq – birds of a feather (January 12, 2010)

Win32/Pushbot

Win32/Pushbot, the fourth most commonly detected bot family in 2Q10, is an IRC–controlled bot that spreads via removable volumes and popular instant messaging programs, as Win32/Rimecud and Win32/Hamweq do. Detections for Pushbot were added to the MSRT in February 2010.

Pushbot is a kit family, so Pushbot detections represent numerous small networks controlled by independent operators, rather than a single monolithic botnet. Pushbot variants are based on a kit called Reptile, which dates to 2005 and is itself based on the even older Win32/Sdbot family.

Because Pushbot bots are created and released by different people, the functionality can vary from one bot to the next, although they all share a number of basic features. Fundamentally they are all IRC bots, although each may be controlled through a different IRC server. Like Rimecud and Hamweq, some variants spread by copying themselves to the file sharing directories of popular P2P programs, or use the Recycle Bin to spread to removable volumes. Some recent variants have the instant messaging feature disabled.

The core bot command set is fairly typical. A remote attacker can order a bot to spread via instant messaging, stop spreading, update itself, remove itself, and download and execute arbitrary files, among other possibilities. Some variants may also be able to perform one or more of the following additional activities:

- Spread via removable drives.
- Spread via P2P networking.
- Attempt to terminate other backdoors running on the system by searching the memory of other running processes for particular strings.
- Participate in DDoS attacks.
- Add extra instant messaging contacts.
- Send other messages to the user's contacts.
- Modify the Hosts file to redirect banking sites to a specified location.
- Retrieve data from Windows Protected Storage, which might include auto-complete data and stored passwords from Microsoft Internet Explorer®, Microsoft Outlook®, and Windows Live® Messenger.
- Connect to websites without downloading files.
- Return various spreading and uptime statistics.

Figure 11 lists the other threats most often detected on computers that are infected with Win32/Pushbot.

FIGURE 11. Other threats found on computers infected with Win32/Pushbot in 2Q10

Other Family	Most Significant Category	Percentage of Win32/Pushbot-Infected Computers
Win32/Autorun	Worms	49.7%
Win32/Rimecud	Worms	36.5%
Win32/Conficker	Worms	31.9%
Win32/Taterf	Worms	24.2%
Win32/Renos	Trojan Downloaders & Droppers	22.4%

For more information, see the following post at the MMPC blog (http://blogs.technet.com/mmpc):

- MSRT February – When Push Comes to Shove (February 9, 2010)

Win32/IRCbot

Win32/IRCbot, the fifth most commonly detected bot family in 2Q10, is a large family of IRC–controlled backdoor trojans. Variants of this family, which has been detected for many years, exhibit a wide range of behaviors and use several different mechanisms to spread, but they all share the same basic IRC–based C&C design. Some IRCbot variants download and run other malicious software, including other bot families. Some variants are also designed to report system information from the victim's computer to the attacker. Detections for IRCbot were first added to the MSRT in December 2005.

Figure 12 lists the other threats most often detected on computers infected with Win32/IRCbot.

FIGURE 12. Other threats found on computers infected with Win32/IRCbot in 2Q10

Other Family	Most Significant Category	Percentage of Win32/IRCBot-Infected Computers
Win32/Oficla	Trojan Downloaders & Droppers	61.0%
Win32/Autorun	Worms	51.1%
Win32/Rimecud	Worms	39.7%
Win32/VBInject	Miscellaneous Potentially Unwanted Software	33.9%
Win32/Conficker	Worms	33.9%

Other Notable Families

Win32/Virut, the eighth most commonly detected bot family in 2Q10, is a family of file-infecting viruses that target and infect .exe, .scr, and (in some recent variants) .html files accessed on infected computers. Win32/Virut also opens a backdoor by connecting to an IRC server, allowing a remote attacker to download and run files on the infected computer. Most variants attempt to connect to IRC servers located at proxima.ircgalaxy.pl and ircd.zief.pl, which suggests that Virut is closely controlled by a single individual or group. Detections for Virut were first added to the MSRT in August 2007.

Win32/Rbot, the 16th most commonly detected bot family in 2Q10, is another large IRC family with variants that exhibit a wide range of behaviors. Like Win32/Pushbot, Win32/Rbot is a kit family—variants are typically built from an open source botnet creation kit called RxBot, which in turn is based on the older SDBot family. The RxBot kit is widely available among malware operators, and many different groups have produced their own versions with different functionality. Detections for Rbot were first added to the MSRT in April 2005.

Win32/Rustock, the 19th most commonly detected bot family in 2Q10, is a closely controlled family of rootkit–enabled backdoor trojans that were originally developed to help distribute spam. (See "Spam from Botnets" on page 35 for more information about spam sent by Rustock and other botnets.) First discovered sometime in early 2006, Rustock has evolved to become a prevalent and pervasive threat. Recent variants appear to be associated with the incidence of rogue security programs. Detections for Rustock were added to the MSRT in October 2008.

Win32/Waledac, the 24th most commonly detected bot family in 2Q09 is a closely controlled bot family that is best known for sending spam, but has also been observed to download other malware families, including Win32/FakeSpypro and Win32/Rugzip. It uses its own P2P network to distribute commands between infected computers. In February 2010, a U.S. federal court judge granted a Microsoft request[14] to cut off 276 Internet domains controlled by the Waledac attackers, thereby preventing them from issuing commands to the botnet. (See "Win32/Waledac and the Law: Fighting Botnets in Court" beginning on page 40 for more information.) Detections for Waledac were added to the MSRT in April 2009.

Where's Conficker?

Win32/Conficker is a worm that infects computers across a network by spreading via removable hard drives, exploiting weak passwords on file shares, or exploiting a vulnerability in the Server service that was addressed by Microsoft Security Bulletin MS08-067. Infection can result in remote code execution when file sharing is enabled. The worm also disables important system services and some security products and may download arbitrary files. Conficker was the fifth most prevalent malware family in the first half of 2010. For domain-joined computers, Conficker has held the top spot for the past year.

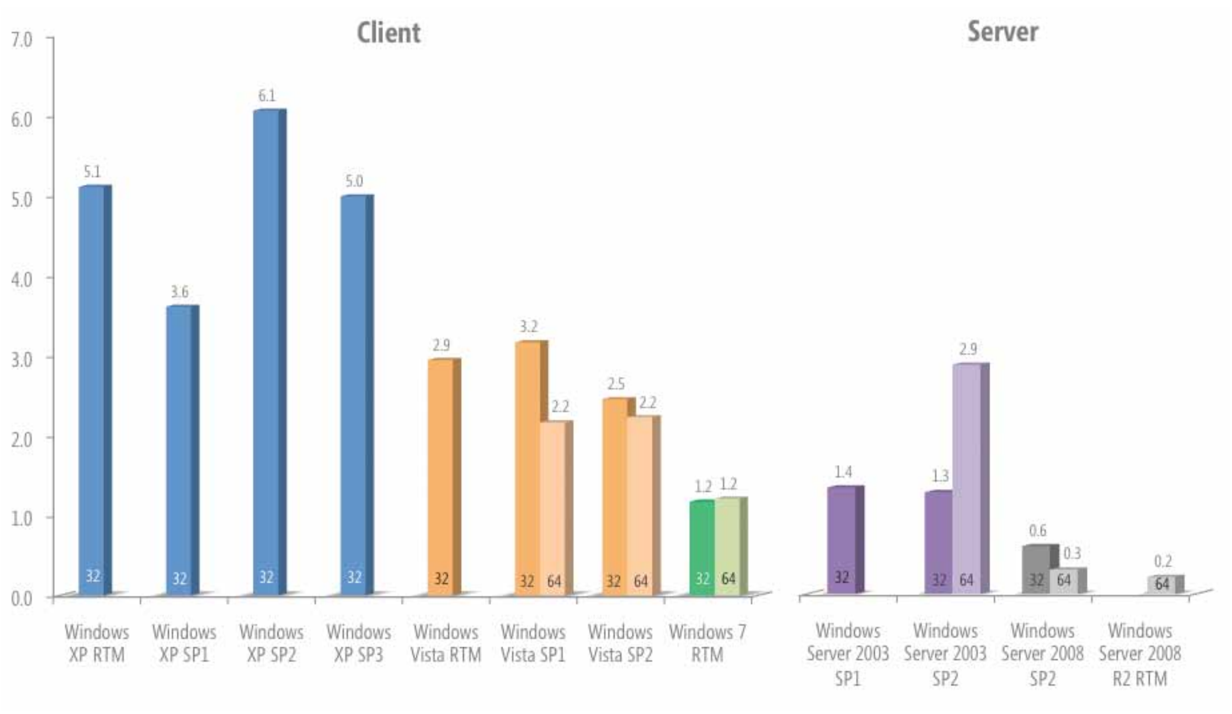
If placed among bot families, Conficker would rank second to Win32/Rimecud. However, the coordinated efforts of the Conficker Working Group effectively limited or eliminated the Conficker bot-herders' ability to issue instructions to infected computers. (See "Case Study: Conficker Working Group" on page 29 of *Microsoft Security Intelligence Report, Volume 7 (January through June 2009)* for more information.) In this report, Win32/Conficker is not tabulated with the other bots and botnet malware so as to more accurately reflect botnets under active control by attackers.

[14] (Microsoft Corporation v. John Does 1-27, et. al 2010)

Operating System Statistics

The features and updates available with different versions of the Windows operating system, along with the differences in the way people and organizations use each version, affect the infection rates seen with different versions and service packs. The following figure shows the average monthly botnet infection rate for each Windows operating system/service pack (SP) combination that accounted for at least 0.05 percent of total MSRT executions in 2Q10, expressed in *bot CCM*, or the number of computers from which bot-related malware was removed by the MSRT for every 1,000 MSRT executions. (See "Infection Rates" in the Reference Guide section of the *Security Intelligence Report* website for more information about the CCM metric and how bot CCM differs from the CCM figures used elsewhere in this report.)

FIGURE 13. Number of computers cleaned of bot-related malware for every 1,000 executions of the MSRT, 2Q10



("32" = 32-bit; "64" = 64-bit. Systems with at least 0.05 percent of total executions shown.)

The botnet infection rate for Windows 7 and Windows Vista® is significantly lower than that of their desktop predecessor Windows XP with any service pack installed, which reflects the security improvements that have been made to the more recent versions of Windows. Considering only computers that have had the most recent service pack for their operating systems installed, the infection rate for Windows XP SP3 is twice as high as that of Windows Vista SP2 and more than four times as high as that of the release-to-manufacturing (RTM) version of Windows 7.

Infection rates for Windows XP RTM and SP1 are lower than those of more recent versions of Windows XP. MSRT installations on the older versions, which are no longer supported by Microsoft, have decreased significantly over the past several quarters as computers have been decommissioned or upgraded. As IT departments and computer users move to more recent service packs or Windows versions, computers running older operating system versions are often relegated to non-production roles or other specialized environments, which may explain the lower infection rates.

Infection rates for Windows Server® are generally lower than those of the client versions of Windows. Servers tend to have a smaller attack surface than computers that run client operating systems because they are more likely to be used under controlled conditions by trained administrators and to be protected by one or more layers of security. Server versions are hardened against attack in a number of ways, which reflects this difference in usage. For example, Internet Explorer Enhanced Security Configuration is enabled by default, and the Roles Wizard automatically disables features that are not needed for the configured server role.

Geographic Statistics

The telemetry data generated by Microsoft security products includes information about the location of the computer, as determined by the setting of the **Location** tab or menu in **Regional and Language Options** in the Control Panel. (See "Geographic Statistics" in the Reference Guide section of the *Security Intelligence Report* for more information.) This data makes it possible to compare infection rates, patterns, and trends in different locations around the world.

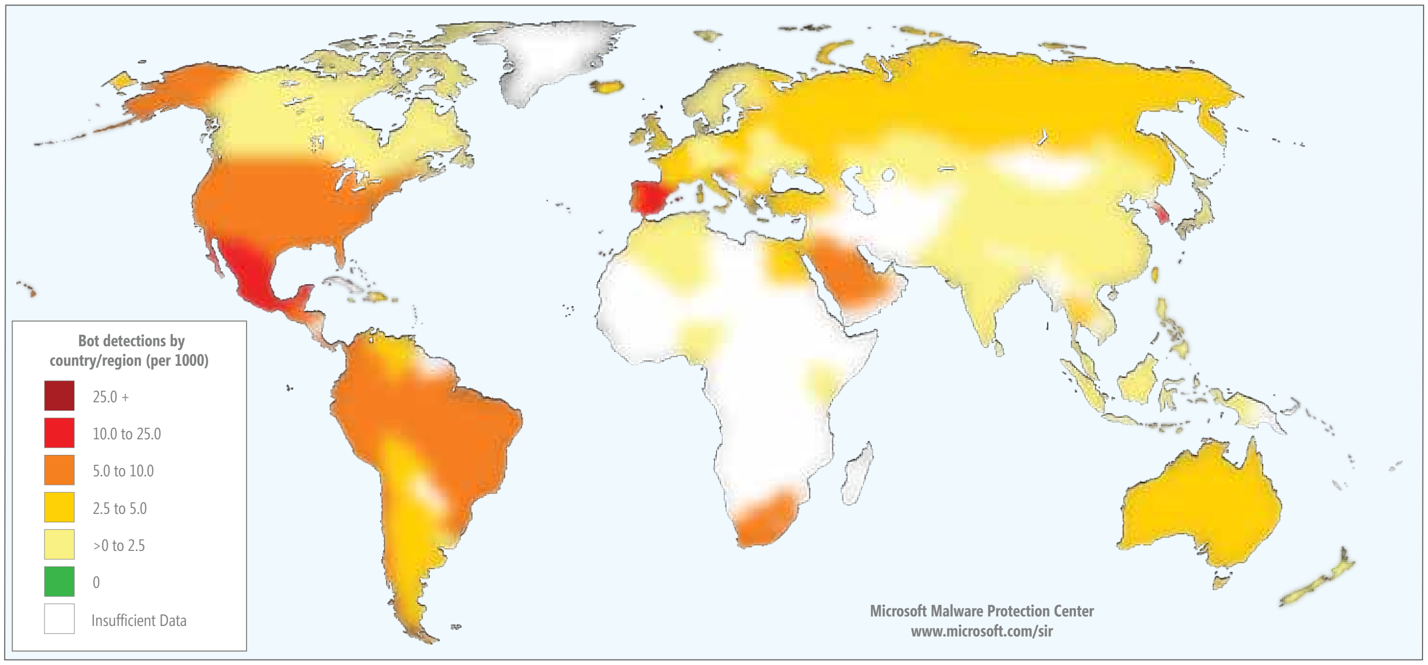
The following table shows the top 25 countries and regions around the world with the most bot infections detected and removed in 2Q10.

FIGURE 14. The 25 locations with the most bot cleanings in 2Q10

Rank	Country/Region	Computers with Bot Cleanings (1Q10)	Computers with Bot Cleanings (2Q10)	Bot Cleanings Per 1000 MSRT Executions (Bot CCM)
1	United States	2,163,216	2,148,169	5.2
2	Brazil	511,002	550,426	5.2
3	Spain	485,603	381,948	12.4
4	Korea	422,663	354,906	14.6
5	Mexico	364,554	331,434	11.4
6	France	344,743	271,478	4.0
7	United Kingdom	251,406	243,817	2.7
8	China	227,470	230,037	1.0
9	Russia	181,341	199,229	4.3
10	Germany	200,016	156,975	1.4
11	Italy	191,588	130,888	2.6
12	Turkey	91,262	98,411	4.7
13	Canada	96,834	87,379	1.4
14	Netherlands	115,349	77,466	2.5
15	Colombia	76,610	71,493	5.8
16	Portugal	83,379	68,903	5.7
17	Australia	72,903	66,576	2.8
18	Poland	87,926	62,704	3.9
19	Taiwan	52,915	54,347	3.4
20	Japan	63,202	52,827	0.6
21	Argentina	38,229	43,162	3.8
22	Saudi Arabia	33,283	40,793	5.5
23	Belgium	51,689	39,508	3.4
24	Chile	37,705	39,245	5.1
25	India	37,895	38,954	1.0

Unsurprisingly, the list is dominated by populous locations with large numbers of computer users, led by the United States and Brazil. When these differences are accounted for by calculating the number of infections found per 1,000 instances of MSRT being run, Korea, Spain, and Mexico have the highest infection rates among the locations on this list, as shown in Figure 15.

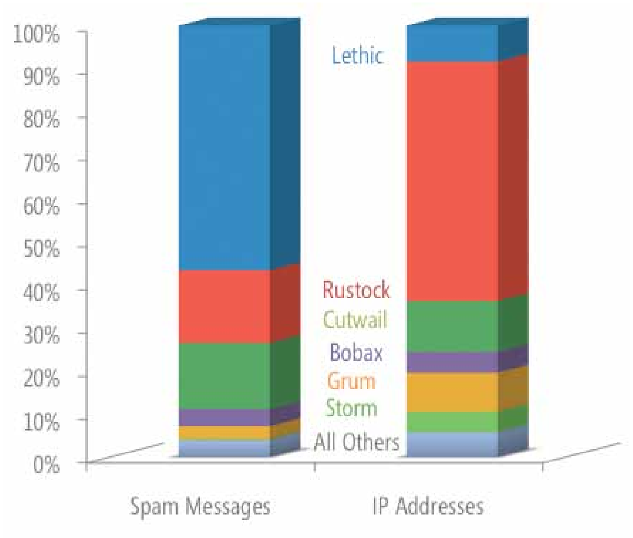
FIGURE 15. Bot infection rates by country/region in 2Q10



Spam from Botnets

Microsoft Forefront® Online Protection for Exchange (FOPE) provides spam, phishing, and malware filtering services for thousands of enterprise customers. To measure the impact that botnets have on the spam landscape, FOPE monitors spam messages sent from IP addresses that have been reported to be associated with known botnets.

FIGURE 16. Botnets sending the most spam from March to June 2010, by percentage of all botnet spam messages and percentage of all botnet IP addresses sending spam



The names used to track botnets usually coincide with the names assigned to the dominant malware families associated with each one, but not always. Some botnets are associated with more than one threat family because of development activity on the part of malware creators and to the existence of generic detections. For example, some Win32/Lethic variants are detected as VirTool:Win32/CeeInject.gen!AS.

The Lethic botnet sent 56.7 percent of the botnet spam received between March and June of 2010 from just 8.3 percent of known botnet IP addresses. Lethic is a closely controlled botnet that uses a custom binary protocol for C&C. A takedown of the Lethic C&C servers in January 2010 disrupted the attackers' ability to send spam,[15] although they subsequently regained control of the botnet, as Figure 16 illustrates.

The Rustock botnet controlled 55.6 percent of known botnet IP addresses, but sent only 16.9 percent of the spam. See "Other Notable Families" on page 31 for more information about Win32/Rustock.

The Cutwail botnet was responsible for 15.4 percent of spam sent from 11.8 percent of known botnet IP addresses. Win32/Cutwail is a multipurpose threat family that employs a rootkit and other defensive techniques to avoid detection and removal.

[15] (M86 Security Labs 2010)

Fighting Back Against Botnets

Detecting Botnets

Methods for detecting bots can generally be divided into two categories— those that involve *static analysis*, or checking computers' characteristics against a list of known threats, and those that involve *behavioral analysis*, or monitoring communications in a network for behaviors that are known to be exhibited by botnets. Static analysis results in more reliable judgments, but requires threat signatures that are current and available. Behavioral analysis potentially allows for much broader detection methods (especially by aggregating information from multiple sources), but is more likely to result in false positives. Effective botnet detection strategies generally involve aspects of both static analysis and behavioral analysis.

Static Analysis

Static analysis methods involve checking items against a known list of malicious or dangerous items, such as executable binaries, URLs, and IP addresses. If the list is accurate and up-to-date, this process can be a very fast and relatively risk-free way to identify bad items. In practice, however, static analysis alone is not an effective way to keep a network free of botnets, because of the continuing efforts of malware authors to generate fully undetected threats. Malware authors use a variety of techniques to avoid detection by antivirus tools and security researchers. These techniques include the following:

- Polymorphism, which involves the creation of multiple unique but functionally identical malware files (See "Defending the Botnet" on page 20 for more information.)
- URL obfuscation methods, such as using escape sequences and converting an IP address to its decimal representation.
- Changing IP addresses rapidly, and using large numbers of alternate URLs that connect to the same resource (or copies of the same resource).
- Serving different downloads or web pages depending on factors like the time of day or the origin of the request (for example, serving clean web pages to requests coming from security software vendors).

Behavioral Analysis

Behavioral analysis can be a powerful tool for identifying botnets, but processing time, the need for an appropriate environment in which to observe the computer's behavior, and the danger of false positives can make diagnosis difficult. The process is further complicated by the tendency of some malware to refuse to run if it detects that it is being executed in a virtual or isolated environment, or a debugger.

It was once common to see bots that would attempt connections to each port on a target computer in sequence (a port scan). This technique allowed the target to recognize an attacker quite easily. Now it appears that most bots use targeted attacks in their efforts to spread. They examine only a small number of ports, which are generally those that are in use by some other service and therefore open to connections.

Researchers from Microsoft and elsewhere have observed that the external behavior of bots tends to have a number of distinctive characteristics:

- Bot activities are often, although not always, closely coordinated with DDoS attacks and time-sensitive spam and phishing attacks, as evidenced by a sharp correlation in the timing of their network activities. For example, the controller instructs all bots to start sending their pump-and-dump spam payload at the same time. For individual bots, network activity tends to be almost silent for much of the time, and then have a very high number of connections in a short period of time.
- The intervals between a bot's acquisition of new targets (distinct destination IP addresses) are generally much shorter than the intervals between an uninfected computer's communication with other distinct IP addresses. In other words, bots talk to more distinct IP addresses in a shorter period of time than uninfected computers do.[16] In addition, bots tend to evenly distribute their attentions among their targets—they make about the same number of connections to each of a large number of destination IP addresses.
- Bots often have a higher number of failed connections than uninfected computers do.[17]
- Bots that are controlled via IRC often exhibit a significant amount of IRC traffic. IRC is a well-known protocol that is used legitimately in many contexts, including games and technical support web applications, but it is still relatively rare and many computers and even whole networks have no legitimate reason to use it at all.
- Bots are much more likely than uninfected computers to send large volumes of email from locally installed Simple Mail Transfer Protocol (SMTP) servers. Most home and enterprise users connect to email servers that are operated by their ISPs or IT departments, and have no need to install SMTP servers on their desktop or laptop computers.
- Some bots use the User Datagram Protocol (UDP) exclusively, which is somewhat unusual for Internet communication.
- HTTP bots often communicate using the IP addresses of web servers rather than server names, which is less common for legitimate traffic. HTTP traffic between bots and C&C servers can include suspicious URI strings or nonstandard HTTP headers (such as the "Entity-Info:" and "Magic-Number:" headers used by Win32/Bredolab C&C servers to transmit data).

However, it is certainly possible for legitimate computer users and programs to exhibit many of the behaviors listed here. In particular, online games often use IRC and UDP extensively, and have built-in SMTP servers .[18]

[16] (Jung 2006)
[17] (Jung 2006)
[18] (Karasaridis, Rexroad and Hoeflin 2007)

Honeypots

A *honeypot* is a computer that is configured by security analysts to act as a deliberate target for malware infection. The intent is to collect malware infections so that their behavior can be analyzed in detail, and in some cases to collect logs of the bots' activities. The Honeynet Project (http://honeynet.org), one of the best known sources of honeypot data, provides information, tools, and techniques for setting up honeypots and analyzing the data they provide.

Darknets

A *darknet* is a subnet of unused IP addresses that are monitored for incoming traffic. The intent is to detect malware as it scans a subnet while crossing over the darknet. Data can also be used to identify network configuration issues. Darknets can be used to host flow collectors, DDoS backscatter detectors, and intrusion detection systems as well as to redirect traffic to honeypots.

Win32/Waledac and the Law: Fighting Botnets in Court

Microsoft Digital Crimes Unit (DCU)

Digital criminals use a variety of sophisticated tools and techniques to attack unsuspecting computer users. Often these attacks leverage social engineering techniques that trick users into installing botnet malware that is designed to conscript computers into a botnet army, such as the Win32/Waledac family of malware.

Botnet armies consist of thousands, if not millions, of computers that are called upon to execute a wide range of illegal activities on behalf of those who control them. Such large scale botnets have begun to threaten the very fabric of the Internet. Given this threat, for the past two years the Microsoft Digital Crimes Unit (DCU) has been examining how botnets leverage the Internet's Domain Name System (DNS) infrastructure and how they have evolved into criminal infrastructures used to abuse all that use the Internet.

A New Approach

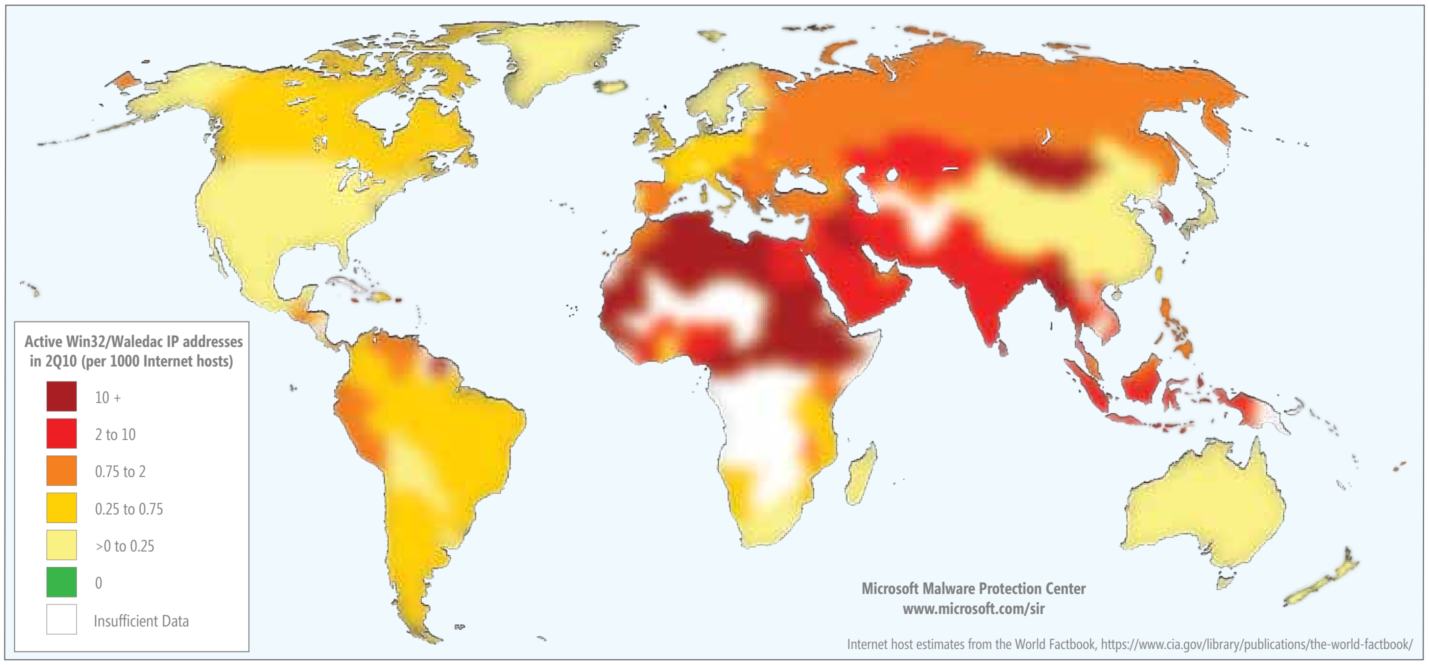
At the Digital Crimes Consortium (DCC) meeting held in Redmond, Washington in October of 2009, the DCU called for industry, law enforcement agencies, government entities, and academics to become more proactive in protecting customers and citizens. Within Microsoft, DCU partnered with Trustworthy Computing (TWC) and sought to develop a new program that would begin to reshape the way Microsoft approached large scale systemic threats that affect users on the Internet. The Microsoft Active Response Strategy (MARS) was created to examine new ways of tackling such threats.

MARS builds on the "Protect Your PC" messaging while adding proactive real world disruptive action and incorporating new ways of working with ISPs, governments, law enforcement, CERTS and academics to help customers and citizens ensure their computers are clean and secure.

Why Waledac, Why Now?

Waledac was selected because of its technical complexity. The malware utilized a robust communication mechanism, with both a peer-to-peer (P2P) and hypertext transport protocol (HTTP) command and control infrastructure that is able to control thousands of infected computers around the world. A unique feature of this communication mechanism was the way its structure was divided into tiers, with each tier serving to abstract it from all the others. The complexity of Waledac's communication mechanism made it a favorite of numerous researchers within the security community, many of whom had been examining Waledac for a number of years and publishing their findings in various technical and academic journals. Consequently, Microsoft began its analysis by first conducting a detailed examination of the research already conducted by researchers at TrendMicro, iDEFENSE, the Microsoft Malware Protection Center (MMPC), and a host of other organizations.

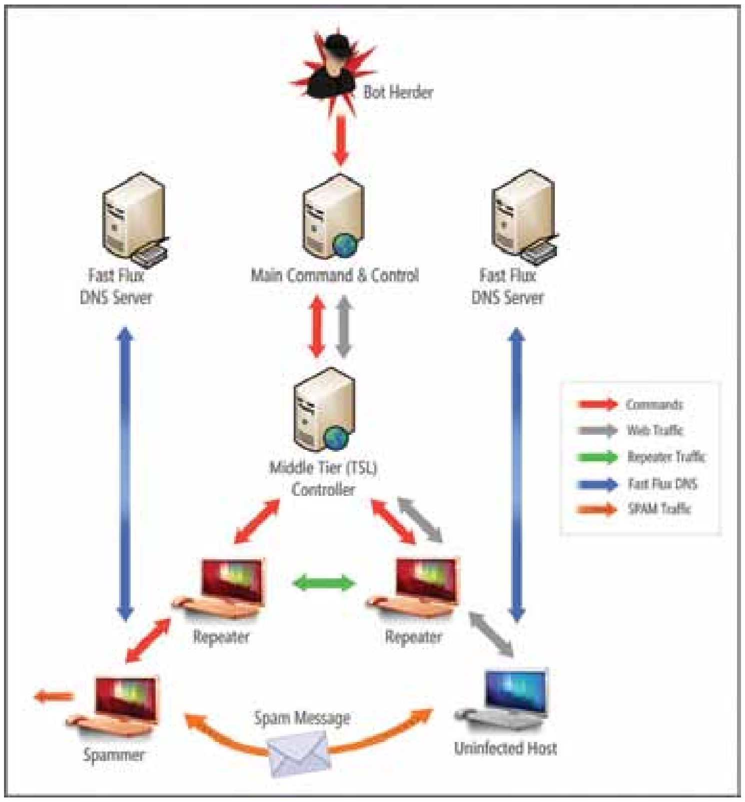
FIGURE 17. Active IP addresses in the Win32/Waledac botnet in 2Q10, per 1000 Internet hosts


Active Win32/Waledac IP addresses in 2Q10 (per 1000 Internet hosts)
10 +
2 to 10
0.75 to 2
0.25 to 0.75
>0 to 0.25
0
Insufficient Data
Microsoft Malware Protection Center
www.microsoft.com/sir
Internet host estimates from the World Factbook, https://www.cia.gov/library/publications/the-world-factbook/

Technical Action Plan

Working with resources from the University of Washington, the Shadowserver Foundation, the Technical Universities of Vienna and Bonn, and the MMPC, Microsoft began to outline a plan of action. Ultimately a plan was developed by Microsoft, the Technical University of Vienna, and iDEFENSE for the technical disruption of the Waledac botnet. The plan called for a three-pronged approach: 1) disruption of the peer-to-peer command and control mechanism; 2) disruption of the DNS/HTTP command and control mechanism; and 3) disruption of the top two tiers of command and control. The timing of the operations had to be coordinated to avoid detection and so that the team could maximize the effect of the countermeasures by leveraging the element of surprise.

FIGURE 18. The Win32/Waledac tier infrastructure



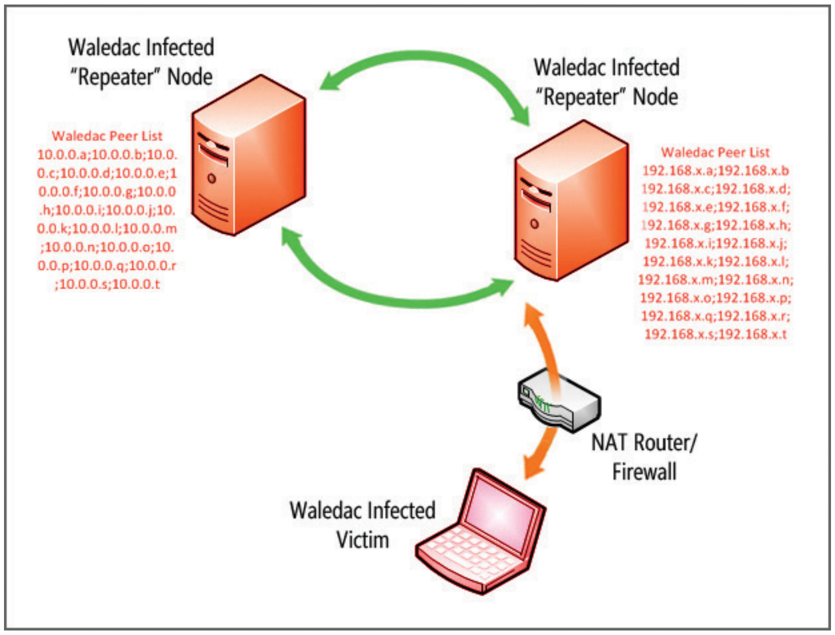
Waledac was a relatively robust and decentralized botnet, which made it difficult to attack. The preceding figure illustrates the overall design of the Waledac botnet infrastructure. To take down such a threat it was important to understand its operation.

A central aspect of the innovative command and control mechanism is the bot classification routine during infection. When a computer becomes infected with the Waledac malware it is classified as either a spammer node or repeater node. If the infected computer is using an RFC 1918 address, it is assumed to be behind a network address translation (NAT) device and is automatically relegated to a spammer node status. The assumption is that the newly infected computer will not be accessible via TCP port 80, the protocol and port that Waledac uses for command and control. If the infected computer is configured with a publicly routable IP address, it is then tested to see if it is accessible via TCP port 80.I If the computer passes both tests, it becomes part of the repeater node tier and is added to the fast flux DNS infrastructure.

Repeater nodes make up the bulk of the computers in the peer-to-peer infrastructure. The job of the repeater nodes is twofold:

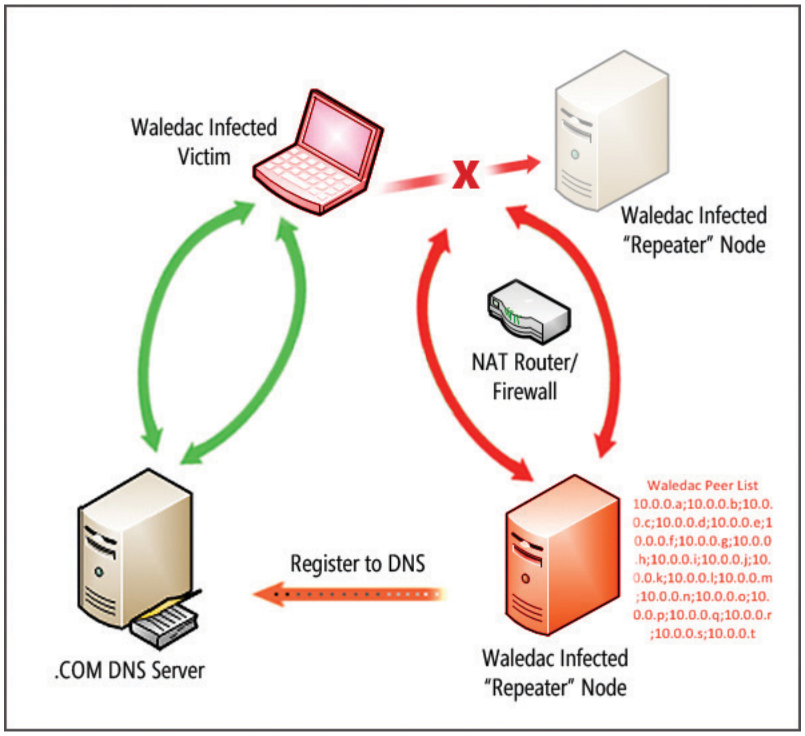
1. Repeater nodes are responsible for maintaining valid peering lists. Because the repeater level of the botnet was large (several thousand at any given time), it was not practical for each repeater to maintain a list of all repeater nodes. Therefore, each repeater node would only maintain a list of approximately 100 other repeater nodes known to it at any given time. This list would be traded or exchanged with other repeater nodes, as well as with spammer nodes, that make contact with the repeater.

FIGURE 19. Win32/Waledac peering list exchange between infected spammer node and repeater node


Waledac Infected "Repeater" Node
Waledac Peer List
10.0.0.a;10.0.0.b;10.0.0.c;10.0.0.d;10.0.0.e;10.0.0.f;10.0.0.g;10.0.0.h;10.0.0.i;10.0.0.j;10.0.0.k;10.0.0.l;10.0.0.m;10.0.0.n;10.0.0.o;10.0.0.p;10.0.0.q;10.0.0.r;10.0.0.s;10.0.0.t
Waledac Infected "Repeater" Node
Waledac Peer List
192.168.x.a;192.168.x.b 192.168.x.c;192.168.x.d;192.168.x.e;192.168.x.f;192.168.x.g;192.168.x.h;192.168.x.i;192.168.x.j;192.168.x.k;192.168.x.l;192.168.x.m;192.168.x.n;192.168.x.o;192.168.x.p;192.168.x.q;192.168.x.r;192.168.x.s;192.168.x.t
NAT Router/ Firewall
Waledac Infected Victim

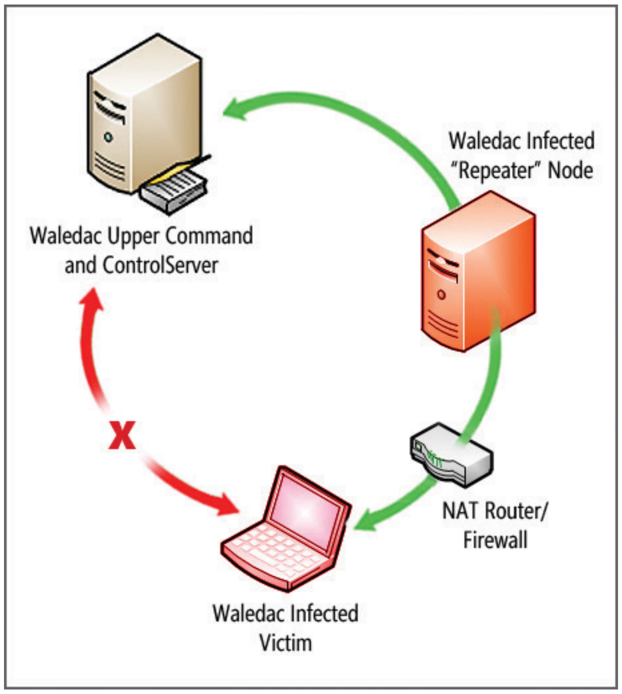
Each repeater node plays a role in the fast flux DNS infrastructure. When a repeater node has been online for a certain amount of time, it is registered in DNS as an authoritative name server for the registered domains and is used for command and control of the Waledac botnet. If a Waledac-infected computer is not able to connect to a repeater node through its seeded repeater list, it will call out to any number of domain names (usually less than 10) as a backup mechanism for bootstrapping into the botnet. These domain names are constantly being updated by the botnet with newly added repeater node computers. This update process serves as a failover mechanism if the seed list becomes stale.

FIGURE 20. Win32/Waledac peering list exchange using fast flux DNS


Waledac Infected Victim
X
Waledac Infected "Repeater" Node
NAT Router/ Firewall
Waledac Peer List
10.0.0.a;10.0.0.b;10.0.0.c;10.0.0.d;10.0.0.e;10.0.0.f;10.0.0.g;10.0.0.h;10.0.0.i;10.0.0.j;10.0.0.k;10.0.0.l;10.0.0.m;10.0.0.n;10.0.0.o;10.0.0.p;10.0.0.q;10.0.0.r;10.0.0.s;10.0.0.t
Register to DNS
.COM DNS Server
Waledac Infected "Repeater" Node

2. Each repeater node serves as a proxy to the upper tier command and control servers as a way of obfuscating these upper, more traditional command and control mechanisms. The upper tiers of the Waledac botnet are comprised of servers configured by the bot herder and are not Waledac-infected computers. Because these computers are significantly more static, the repeater nodes offer a mechanism to prevent direct access to these upper tiers. In addition, the upper tiers are only accessible through the repeater node proxy functionality, which provides a kind of firewall effect.

FIGURE 21. Win32/Waledac proxy function through the repeater tier

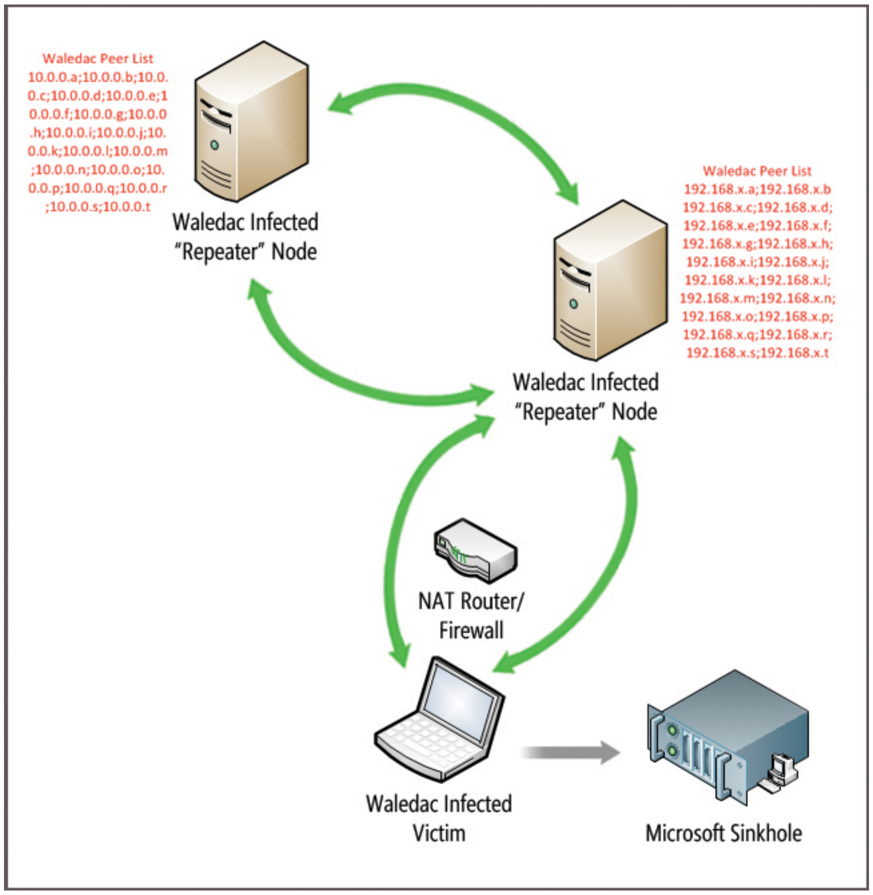

Waledac Upper Command and ControlServer
Waledac Infected "Repeater" Node
X
NAT Router/ Firewall
Waledac Infected Victim

Tackling P2P

Disrupting the P2P network used by Waledac for command and control was the first component in the technical plan. The "repeater" tier of the Waledac botnet was comprised of infected computers that have public IP addresses and were reachable on TCP port 80. When a computer became infected with the Waledac botnet malware it would reach out to any number of preconfigured IP addresses compiled in the malicious binary. After a connection was established to the repeater node, the infected victim would download an updated list of active repeater nodes.

In later versions of the Waledac malware, the P2P bootstrap mechanism was less than reliable; it was not uncommon for an infected honeypot to work through 20 or so IP addresses before reaching a node that was an active repeater, often using the DNS fallback mechanism. Security researcher partners were able to develop a custom repeater that would provide the ability to "poison" the peering tables of all computers that connected to the custom repeater. The thought was not to get a list of available Waledac repeaters, but for the security team to poison the network with its own repeater nodes as well as the sinkhole devices they deployed. This part of the operation needed to happen first so that the security team could infiltrate the P2P network and thereby limit the ability of the bad actor(s) from being able to inject themselves back into the command and control infrastructure. However, this approach alone wasn't sufficient. Because of Waledac's heavy reliance on DNS as a backup mechanism for bootstrapping infected computers into the botnet, the domain names had to be dealt with in close coordination with the P2P countermeasures.

FIGURE 22. Poisoning the Win32/Waledac botnet with Microsoft sinkhole IP addresses



After the P2P poisoning was underway, the DNS issue had to be addressed. Although the DNS mechanism that was used to control the Waledac botnet was a very complex mechanism, attorneys at the Microsoft DCU developed a new approach to deal with the command and control mechanism.

The Legal Action Plan

Informal efforts to take down the domains that supported the Waledac botnet posed serious limitations. Cease and desist letters would not force immediate action. Similarly, domain takedown is inexact and somewhat limited under typical ICANN procedures. For example, ICANN's Uniform Domain-Name Dispute-Resolution Policy (UDRP) provides a relatively long window in which bad actors would be able to register new domains, update the botnet code, or take other evasive actions to move the botnet while the ICANN process unfolded.

One effective alternative to such informal efforts was to apply in federal court for a temporary restraining order (TRO) to shut down the malicious domains at the registry level. An *ex parte* TRO is an extraordinary remedy that allows a party to temporarily restrain (between 14 to 28 days) a bad actor's harmful conduct ***without notice*** to the bad actor and ***without giving them an opportunity to be heard***. In particular, federal courts may temporarily restrain a bad actor's conduct without notice in situations where notice would give the bad actor an opportunity to destroy evidence, relocate the instrumentalities of the harmful conduct, or avoid prosecution. A court can issue an *ex parte* TRO to shut down malicious domains with limited delay (within 48 hours of the application for relief). And because bad actors will not receive notice, they are deprived an opportunity to register new domains or take other actions to redirect the botnet.

Federal courts, however, are reluctant to issue *ex parte* TROs because of concerns about their inherent inconsistency with constitutional due process. Microsoft was able to twice obtain *ex parte* relief by providing the court with strong evidence of the merits at the outset of the case and a definitive strategy for effecting legal notice to the bad actors immediately after the domains had been shut down, thereby preserving their due process rights.

Legal Basis for *Ex Parte* Relief: Under Rule 65 of the Federal Rules of Civil Procedure, a party can apply for a TRO and preliminary injunction to temporarily stop the harm caused by a bad actor's wrongful conduct. Rule 65 allows a party to proceed without notice if it can show (1) that it will suffer immediate and irreparable harm if the relief is not granted, and (2) the party attempts to provide the other side with notice. Rule 65 typically requires that the moving party attempt to provide the adverse party with notice and, if unsuccessful, explain why it was unable to provide notice. Therefore, Rule 65 is by itself insufficient to overcome the obstacles posed by informal efforts to shut down malicious domains.

However, courts have recognized in some circumstances that providing notice to a bad actor would give them an opportunity to avoid prosecution and continue their wrongful conduct. In those circumstances, courts have issued *ex parte* TROs ***– even where a moving party has not attempted notice –*** where notice to the bad actor in those circumstances would render fruitless further prosecution of the action. This principle was applied more than 30 years ago in the context of clothing counterfeiters. A federal appellate court concluded that an *ex parte* TRO to seize counterfeit contraband was necessary, because it was apparent from prior experience that notifying one member of the counterfeiting enterprise would allow the bad actors an opportunity to transfer the contraband to other unknown counterfeiters and thereby avoid prosecution. *See In re Louis Vuitton Et Fils S.A.,* 606 F.2d 1 (2d Cir. 1979). Courts have since applied this reasoning to issue *ex parte* TROs for computer–related activity. In 2009, a federal court in California issued an *ex parte* TRO to suspend Internet connectivity of a company that enabled botnet activity on the basis that "Defendant is likely to relocate the harmful and malicious code it hosts and/or warn its criminal clientele of this action if informed of the [plaintiff's] action." *See FTC v. Pricewert LLC et al.,* Case No. 09-2407 (N.D. Cal., Whyte J., June 2, 2009). As such, an order can provide a viable alternative to informal domain takedown processes.

With substantial factual support from its investigation of the Waledac botnet, Microsoft provided a compelling argument that the botnet's fundamental characteristic – namely, the propensity of the botnet controllers to move and conceal the botnet and the ease by which they could do so – warranted the issuance of an *ex parte* TRO.

Obtaining *Ex Parte Relief*: Even if the facts warrant *ex parte* relief, a court may still refuse to grant such an order if it is not convinced that the bad actor's due process rights will be preserved. Microsoft obtained *ex parte* relief in this case by, among other things, offering the court a definitive and robust strategy for serving the domain registrants with all papers in the case. Microsoft outlined the steps it would undertake to serve the bad actors under the Federal Rules of Civil Procedure as well as proposed alternative methods for service. In addition, Microsoft explained how its proposed methods of service were reasonably calculated to provide the domain registrants with notice and satisfied due process.

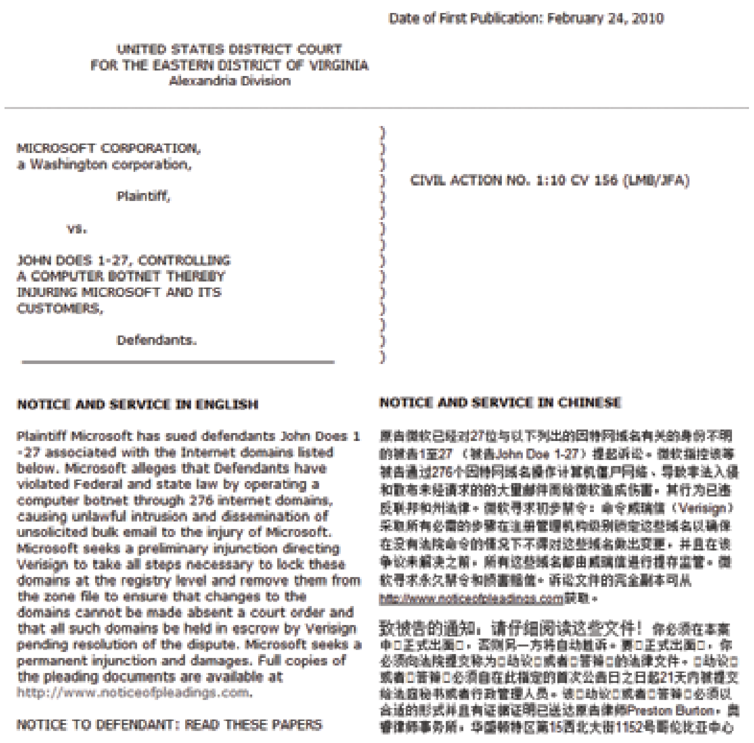
The Microsoft approach in this matter consisted of the following steps:

1. **Filing of the John Doe Lawsuit to Protect Innocent Domain Registrants:** To obtain an *ex parte* TRO, a preliminary injunction, or any other relief, a party must first initiate litigation against the bad actors. The most immediately obvious candidates to be named as defendants were the registrants of the malicious Waledac domains. However, this information raised an obstacle in that there was a possibility that the domains themselves had been hijacked. Therefore, to protect potentially innocent domain registrants, Microsoft filed a John Doe lawsuit that did not name the domain registrants as defendants, but rather named 27 John Does, with each "doing business as" a particular domain registrant. In this way, Microsoft provided the court with an identifiable target for legal service and notice while protecting the registrants' due process rights.
2. **Articulating a Definitive Strategy to Effect Legal Service:** Notwithstanding the concern about innocent domain registrants, a review of the domain registrant information revealed a likelihood that much of the information was false. Most of the domains were registered through registrars in China. Microsoft worked with its attorneys in China to research and confirm the validity of this information, to the extent possible.

The falsified registrant information posed a fundamental question regarding how to quickly and effectively identify, locate, and notify the botnet operators of the lawsuit immediately after the domain takedown was ordered by the Court. Anticipating this issue, in its application for an *ex parte* TRO Microsoft articulated a definitive strategy to effect service on the domain registrants that satisfied the Federal Rules of Civil Procedure, the U.S. Constitution, and was consistent with the laws of China. Under these rules, a plaintiff can serve a non-U.S. defendant pursuant to delivery through official means set forth in various treaties and also by certain alternative means calculated to give notice, as long as not prohibited by treaty.

As part of its strategy, Microsoft proposed serving the international domain registrants through (1) the Hague Convention on Service Abroad by sending the Complaint, Summons and all other documents to the Chinese Ministry of Justice; (2) alternative methods, including service and notice by email, facsimile, and by mail; and (3) publication of all relevant pleadings on a website Microsoft set up solely to provide the domain registrants with notice (www.noticeofpleadings.com). When the court ultimately issued its *ex parte* TRO, it concluded that these methods satisfied due process requirements. In part, this conclusion was made because the registrants had agreed in their registrar-registrant agreements to receive notices through their contact information and because publication is a well-established alternative means of providing notice.

FIGURE 23. Notice posted at www.noticeofpleadings.com



Date of First Publication: February 24, 2010

UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF VIRGINIA
Alexandria Division

MICROSOFT CORPORATION,
a Washington corporation,

Plaintiff,

vs.

JOHN DOES 1-27, CONTROLLING A COMPUTER BOTNET THEREBY INJURING MICROSOFT AND ITS CUSTOMERS,

Defendants.

CIVIL ACTION NO. 1:10 CV 156 (LMB/JFA)

NOTICE AND SERVICE IN ENGLISH

Plaintiff Microsoft has sued defendants John Does 1-27 associated with the Internet domains listed below. Microsoft alleges that Defendants have violated Federal and state law by operating a computer botnet through 276 internet domains, causing unlawful intrusion and dissemination of unsolicited bulk email to the injury of Microsoft. Microsoft seeks a preliminary injunction directing Verisign to take all steps necessary to lock these domains at the registry level and remove them from the zone file to ensure that changes to the domains cannot be made absent a court order and that all such domains be held in escrow by Verisign pending resolution of the dispute. Microsoft seeks a permanent injunction and damages. Full copies of the pleading documents are available at http://www.noticeofpleadings.com.

NOTICE TO DEFENDANT: READ THESE PAPERS

NOTICE AND SERVICE IN CHINESE

原告微软已经对27位与以下列出的因特网域名有关的身份不明的被告1至27（被告John Doe 1-27）提起诉讼。微软指控该等被告通过276个因特网域名操作计算机僵尸网络，导致非法入侵和散布未经请求的的大量邮件而给微软造成伤害，其行为已违反联邦和州法律。微软寻求初步禁令：命令威瑞信（Verisign）采取所有必需的步骤在注册管理机构级别锁定这些域名以确保在没有法院命令的情况下不得对这些域名做出变更，并且在该争议未解决之前，所有这些域名都由威瑞信进行提存监管。微软寻求永久禁令和损害赔偿。诉讼文件的完全副本可从http://www.noticeofpleadings.com获取。

致被告的通知，请仔细阅读这些文件！你必须在本案中□正式出面□，否则另一方将自动胜诉。要□正式出面□，你必须向法院提交称为□动议□或者□答辩□的法律文件。□动议□或者□答辩□必须自在此指定的首次公告日之日起21天内被提交给法庭秘书或者行政管理人员。该□动议□或者□答辩□必须以合适的形式并且有证据证明已送达原告律师Preston Burton，奥睿律师事务所，华盛顿特区第15西北大街1152号哥伦比亚中心

1. **Understanding the Venue:** Microsoft knew it would file in the United States District Court for the Eastern District of Virginia. Therefore, it researched the closest possible scenarios under that Court's prior decisions and paid careful attention to the concerns expressed in the Court's decisions. Understanding the venue allowed Microsoft to craft its arguments in a manner that anticipated questions and obstacles that the Court might raise. This approach was especially important, given the absence of the Defendants to raise such issues.
2. **Developing Credibility with the Court:** Microsoft worked very hard to develop and maintain credibility with the Court by ensuring that its arguments were supported by substantial evidence and law, and also by offering timely submissions, avoiding undue delay, and ensuring that it worked with counsel who was familiar with the Court's practices.

Taking Down the Botnet, Effecting Legal Service, and Preparing for the Preliminary Injunction Hearing: A TRO usually expires 14 days after it is issued. In the Waledac takedown, the court issued the *ex parte* TRO and an Order to Show Cause why it should not issue a preliminary injunction. A preliminary injunction enjoins the wrongful conduct until the case has been concluded. After the court granted the *ex parte* TRO, Microsoft had 14 days to (1) shut off the infected domains, (2) serve the domain registrants, and (3) prepare for the preliminary injunction hearing.

The Microsoft planning and coordination efforts facilitated an immediate shutdown of the infected domains within 48 hours of receiving the Court's temporary order. By having a definitive strategy in place to effect legal service abroad, Microsoft was able to begin serving the domain registrants also within 24 hours of shutting down the infected domains. Microsoft worked with its attorneys in China to coordinate attempted service through China's Ministry of Justice and to continually work with the China domain registrars who could assist in attempting to notify the registrants of the action. Microsoft went to great lengths to provide notice of the action through all of the email addresses of the registrants and by widely publicizing the action and the papers filed in the action on www.noticeofpleadings.com and in the press.

Entry of Default and Entry of Default Judgment: As part of its overall strategy, Microsoft wanted to obtain long-term control over the infected botnet domains and to enjoin the defendants from any ongoing operation of the Waledac botnet. In July 2010, Microsoft filed a motion requesting the Court to enter default against the Doe defendants and a motion requesting that the court and transfer the infected domains to Microsoft. Under Rule 55 of the Federal Rules of Civil Procedure, a Court can enter default when a party, after being served, has failed to defend the action. After the Court enters default, it may enter default judgment against the defendants and award the plaintiff its requested relief.

Despite the extraordinary efforts of Microsoft to provide notice to defendants over a period of many months, defendants have not responded in any manner whatsoever and have not objected to any of the Court's actions. On this basis, in August 2010, the Court ordered the clerk to enter default as to each of the defendants. In early September 2010, the Court held a hearing on entry of default judgment against the defendants and transferring Microsoft ownership of the domains; a permanent injunction is pending.

Why This Framework Was Successful: Aside from having a compelling investigation and evidence as well as strong legal arguments in its favor, this framework proved successful because Microsoft carefully coordinated the legal relief with sophisticated technical action, anticipated all of the arguments the absent defendants might raise, strove to answer those arguments in their absence, took steps to understand the venue it was operating in, and developed credibility with the Court.

Works Cited

Bächer, Paul, Thorsten Holz, Markus Kötter, and Georg Wicherski. "Know your Enemy: Tracking Botnets." *The Honeynet Project.* August 10, 2008.

Canavan, John. "The Evolution of Malicious IRC Bots." *VB2005 Conference.* Dublin, 2005.

Danchev, Dancho. "The Neosploit cybercrime group abandons its web malware exploitation kit." *Zero Day blog.* July 29, 2008.

Federal Bureau of Investigation. "FBI, Slovenian and Spanish Police Arrest Mariposa Botnet Creator, Operators." July 28, 2010.

Fogie, Seth. "Trojan Malware: From Non-profit to Commercial Criminals (A Brief History)." *informIT.* October 27, 2006.

Jung, Jaeyeong. "Real-time detection of malicious network activity using stochastic models." Cambridge, Mass.: Massachusetts Institute of Technology, 2006.

Karasaridis, Anestis, Brian Rexroad, and David Hoeflin. "Wide-Scale Botnet Detection and Characterization." *HotBots 2007.* 2007.

M86 Security Labs. Lethic botnet - The Takedown. January 10, 2010. (accessed September 8, 2010).

Microsoft Corporation v. John Does 1-27, et. al. 1:10CV156 (U.S. District Court for the Eastern District of Virginia, February 22, 2010).

Naraine, Ryan. "Money Bots: Hackers Cash In on Hijacked PCs." *eWeek.* September 9, 2006.

Nazario, Jose. Twitter-based Botnet Command Channel. August 13, 2009. (accessed September 2, 2010).

Oikarnen, Jarkko. "IRC History." *IRC.org.* n.d. (accessed January 21, 2010).

Ollmann, Gunter. "The Botnet vs. Malware Relationship: The one-to-many botnet myth." *Damballa.com.* May 21, 2009.

United States of America v. Adam Sweaney. Magistrate No. 07–00243M (United States District Court for the District of Columbia, June 11, 2007).

United States of America v. Jeanson James Ancheta. CR05-1060, Indictment (U.S. District Court for the Central District of California, February 2005).

Worldwide Bot Infection Rates

Figure 15 on page 37 displays the bot infection rates for locations worldwide as expressed in *bot CCM*, or the number of computers from which bot-related malware was removed by the MSRT for every 1,000 MSRT executions. For a more in-depth look at how infection rates in different places have changed over the past year, Figure 24 lists the bot CCM figures for 88 locations around the world for each quarter between 3Q09 and 2Q10. (See "Infection Rates" in the Reference Guide section of the *Security Intelligence Report* website for more information about the CCM metric and how bot CCM differs from malware CCM.)

FIGURE 24. Bot infection rates for locations around the world from 3Q09 to 2Q10, by bot CCM

Country/Region	3Q09	4Q09	1Q10	2Q10
Algeria	0.2	0.2	0.3	0.3
Argentina	2.8	2.7	4.4	3.8
Australia	2.4	2.4	3.4	2.8
Austria	1.1	1.3	1.4	1.1
Belarus	0.1	0.1	0.2	0.2
Belgium	2.9	1.8	5.3	3.4
Bolivia	1.4	1.6	3.5	3.1
Brazil	2.9	3.8	7.0	5.2
Bulgaria	2.5	4.0	4.4	3.8
Canada	1.7	1.3	1.6	1.4
Chile	2.3	4.5	7.2	5.1
China	1.4	1.0	1.3	1.0
Colombia	3.1	5.0	9.2	5.8
Costa Rica	2.7	5.0	11.2	7.2
Croatia	4.5	6.0	11.0	8.6
Czech Republic	2.5	1.7	1.8	1.4
Denmark	2.6	1.7	3.0	1.7
Dominican Republic	1.8	2.5	5.1	3.8
Ecuador	1.6	3.8	11.1	5.9
Egypt	1.6	2.8	3.6	2.7
El Salvador	2.3	3.8	11.9	10.9
Estonia	2.6	2.6	6.6	3.0
Finland	1.3	1.2	1.9	0.9
France	3.0	2.7	6.7	4.0
Germany	1.9	1.3	2.0	1.4
Greece	3.6	3.8	6.0	4.3
Guatemala	4.3	3.9	9.8	6.1
Honduras	3.0	5.2	7.9	6.5
Hong Kong S.A.R.	1.3	1.2	1.6	1.1

Country/Region	3Q09	4Q09	1Q10	2Q10
Hungary	5.7	3.9	7.7	4.8
Iceland	3.4	3.6	9.0	4.6
India	0.3	0.5	1.5	1.0
Indonesia	0.7	0.9	0.9	0.8
Ireland	2.0	2.7	3.1	2.6
Israel	4.7	5.3	6.8	4.3
Italy	3.2	2.5	4.3	2.6
Jamaica	0.5	1.3	2.9	1.8
Japan	0.6	0.5	0.8	0.6
Jordan	1.2	2.7	2.9	2.3
Kazakhstan	0.1	0.2	0.4	0.4
Kenya	0.1	0.2	0.2	0.3
Korea	6.3	6.1	17.4	14.6
Kuwait	1.1	3.6	5.2	4.5
Latvia	4.4	3.9	4.5	3.4
Lebanon	0.4	1.3	1.8	1.5
Lithuania	3.8	4.6	7.2	4.6
Luxembourg	1.4	1.4	2.1	1.8
Malaysia	0.5	0.8	1.5	1.2
Mexico	3.5	6.7	14.8	11.4
Morocco	0.1	0.1	0.3	0.2
Netherlands	2.8	2.0	4.6	2.5
New Zealand	1.3	1.4	1.7	1.3
Nigeria	0.0	0.1	0.2	0.2
Norway	2.0	2.0	3.0	2.2
Pakistan	0.0	0.1	0.1	0.1
Panama	2.2	4.1	8.6	5.8
Peru	1.9	3.4	7.8	8.3
Philippines	0.3	0.6	1.1	1.4
Poland	2.9	4.2	6.1	3.9
Portugal	4.1	8.3	9.3	5.7
Puerto Rico	0.7	2.8	2.5	1.8
Qatar	1.0	1.2	2.2	1.8
Romania	1.5	1.6	2.5	2.0
Russia	4.2	3.9	5.2	4.3
Saudi Arabia	2.4	3.6	5.4	5.5

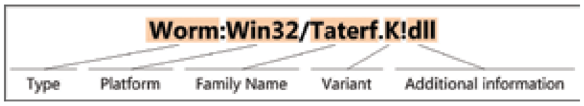
Country/Region	3Q09	4Q09	1Q10	2Q10
Serbia and Montenegro	1.9	2.3	4.2	2.6
Singapore	0.3	1.2	1.5	1.4
Slovakia	2.4	2.3	2.7	2.1
Slovenia	3.6	3.1	9.7	5.9
South Africa	2.6	5.8	9.3	8.4
Spain	6.9	11.0	17.3	12.4
Sri Lanka	0.0	0.0	0.1	0.1
Sweden	2.5	1.8	4.4	2.4
Switzerland	1.4	1.5	1.9	1.4
Taiwan	1.6	1.3	4.5	3.4
Thailand	2.6	1.9	3.0	2.8
Trinidad and Tobago	1.0	2.5	2.5	2.3
Tunisia	0.2	0.2	0.4	0.2
Turkey	4.1	2.8	5.8	4.7
Ukraine	0.7	0.6	1.3	1.1
United Arab Emirates	0.5	1.4	1.8	1.6
United Kingdom	3.2	2.5	3.0	2.7
United States	5.9	4.5	5.6	5.2
Uruguay	1.0	1.0	1.4	1.3
Venezuela	1.3	2.3	5.3	3.9
Vietnam	0.3	0.4	0.4	0.4
Worldwide	2.5	2.5	4.0	3.2

Appendix A: Threat Naming Conventions

The MMPC malware naming standard is derived from the Computer Antivirus Research Organization (CARO) Malware Naming Scheme, originally published in 1991 and revised in 2002. Most security vendors use naming conventions based on the CARO scheme, with minor variations, although family and variant names for the same threat can differ between vendors.

A threat name can contain some or all of the components seen in Figure 25.

FIGURE 25. The Microsoft malware naming convention



The *type* indicates the primary function or intent of the threat. The MMPC assigns each individual threat to one of a few dozen different types based on a number of factors, including how the threat spreads and what it is designed to do. To simplify the presentation of this information and make it easier to understand, the Security Intelligence Report groups these types into 10 categories. For example, the TrojanDownloader and TrojanDropper types are combined into a single category, called Trojan Downloaders & Droppers.

The *platform* indicates the operating environment in which the threat is designed to run and spread. For most of the threats described in this report, the platform is listed as "Win32," for the Win32 API used by 32-bit and 64-bit versions of Windows desktop and server operating systems. (Not all Win32 threats can run on every version of Windows, however.) Platforms can include programming languages and file formats, in addition to operating systems. For example, threats in the ASX/Wimad family are designed for programs that parse the Advanced Stream Redirector (ASX) file format, regardless of operating system.

Groups of closely related threats are organized into *families*, which are given unique names to distinguish them from others. The family name is usually not related to anything the malware author has chosen to call the threat; researchers use a variety of techniques to name new families, such as excerpting and modifying strings of alphabetic characters found in the malware file. Security vendors usually try to adopt the name used by the first vendor to positively identify a new family, although sometimes different vendors use completely different names for the same threat, which can happen when two or more vendors discover a new family independently. The MMPC Encyclopedia (http://www.microsoft.com/security/portal) lists the names used by other major security vendors to identify each threat, when known.

Some malware families include multiple components that perform different tasks and are assigned different types. For example, the Win32/Frethog family includes variants designated PWS:Win32/Frethog.C and TrojanDownloader:Win32/Frethog.C, among others. In the Security Intelligence Report, the category listed for a particular family is the one that

Microsoft security analysts have determined to be the most significant category for the family (which in the case of Frethog is Password Stealers & Monitoring Tools).

Malware creators often release multiple variants for a family, typically in an effort to avoid being detected by security software. Variants are designated by letters, which are assigned in order of discovery—A through Z, then AA through AZ, then BA through BZ, and so on. A variant designation of "gen" indicates that the threat was detected by a generic signature for the family rather than as a specific variant. Any additional characters that appear after the variant provide comments or additional information.

In the *Security Intelligence Report,* a threat name consisting of a platform and family name (like "Win32/Taterf") is a reference to a family. When a longer threat name is given (like "Worm:Win32/Taterf.K!dll"), it is a reference to a more specific signature or to an individual variant. To make the report easier to read, family and variant names have occasionally been abbreviated in contexts where confusion is unlikely. Thus, Win32/Taterf is referred to simply as "Taterf" on subsequent mention in some places, and Worm:Win32/Taterf.K simply as "Taterf.K".

Appendix B: Data Sources

Microsoft Products and Services

Data included in the Microsoft Security Intelligence Report is gathered from a wide range of Microsoft products and services. The scale and scope of this telemetry allows the Security Intelligence Report to deliver the most comprehensive and detailed perspective on the threat landscape available in the software industry:

- Bing, the search and decision engine from Microsoft, contains technology that performs billions of Web-page scans per year to seek out malicious content. Once detected, Bing displays warnings to users about the malicious content to help prevent infection.
- Windows Live Hotmail has hundreds of millions of active e-mail users in more than 30 countries/regions around the world.
- Forefront Online Protection for Exchange protects the networks of thousands of enterprise customers worldwide by helping to prevent malware from spreading through e-mail. FOPE scans billions of e-mail messages every year to identify and block spam and malware.
- Windows Defender is a program, available at no cost to licensed users of Windows, that provides real-time protection against pop-ups, slow performance, and security threats caused by spyware and other potentially unwanted software. Windows Defender runs on more than 100 million computers worldwide.
- The Malicious Software Removal Tool (MSRT) is a free tool designed to help identify and remove prevalent malware families from customer computers. The MSRT is primarily released as an important update through Windows Update, Microsoft Update, and Automatic Updates. A version of the tool is also available from the Microsoft Download Center. The MSRT was downloaded and executed 3.2 billion times in 1H10, or nearly 550 million times each month on average. The MSRT is not a replacement for an up-to-date antivirus solution because of its lack of real-time protection and because it uses only the portion of the Microsoft antivirus signature database that enables it to target specifically selected, prevalent malicious software.
- Microsoft Forefront Client Security is a unified product that provides malware and potentially unwanted software protection for enterprise desktops, laptops, and server operating systems. Like Windows Live OneCare, it uses the Microsoft Malware Protection Engine and the Microsoft antivirus signature database to provide real-time, scheduled, and on-demand protection.
- Windows Live OneCare is a real-time protection product that combines an antivirus and antispyware scanner with phishing and firewall protection. Microsoft has discontinued retail sales of Windows Live OneCare, but continues to make definition updates available to registered users.

- The Windows Live OneCare safety scanner (http://safety.live.com) is a free, online tool that detects and removes malware and potentially unwanted software using the same definition database as the Microsoft desktop anti-malware products. The Windows Live OneCare safety scanner is not a replacement for an up-to-date antivirus solution, because it does not offer real-time protection and cannot prevent a userís computer from becoming infected.
- Microsoft Security Essentials is a basic, consumer-oriented anti-malware product, offered at no charge to licensed users of Windows, which provides real-time protection against viruses, spyware, and other harmful software.
- The Phishing Filter (in Internet Explorer 7) and the SmartScreen Filter (in Internet Explorer 8) offer Internet Explorer users protection against phishing sites and sites that host malware. Microsoft maintains a database of phishing and malware sites reported by users of Internet Explorer and other Microsoft products and services. When a user attempts to visit a site in the database with the filter enabled, Internet Explorer displays a warning and blocks navigation to the page.

Product Name	Main Customer Segment		Malicious Software		Spyware and Potentially Unwanted Software		Available at No Additional Charge	Main Distribution Methods
	Consumer	Business	Scan and Remove	Real-Time Protection	Scan and Remove	Real-Time Protection		
Microsoft Forefront Server Security		•	•	•	•	•		Volume Licensing
Microsoft Forefront Client Security		•	•	•	•	•		Volume Licensing
Windows Live OneCare Safety Scanner	•		•		•		•	Web
Microsoft Security Essentials	•		•		•	•	•	Download Center
Windows Malicious Software Removal Tool	•		Prevalent Malware Families				•	Windows Update/ Automatic Updates Download Center
Windows Defender	•				•	•	•	Download Center Windows Vista Windows 7
Microsoft Forefront Online Protection for Exchange		•	•	•				Volume Licensing

Glossary

adware
A program that displays advertisements. Although some adware can be beneficial by subsidizing a program or service, other adware programs may display advertisements without adequate consent.

backdoor trojan
A type of trojan that provides attackers with remote access to infected computers. Bots are a sub- category of backdoor trojans. Also see *botnet*.

bot
A malware program that joins an infected computer to a *botnet*.

bot-herder
An operator of a *botnet*.

botnet
A set of computers controlled by a “command-and-control” (C&C) computer to execute commands as directed. The C&C computer can issue commands directly (often through Internet Relay Chat [IRC]) or by using a decentralized mechanism, like peer-to-peer (P2P) networking. Computers in the botnet are often called nodes or zombies.

C&C
See *botnet*.

CCM
Short for *computers cleaned per mille* (thousand). The number of computers cleaned for every 1,000 executions of the MSRT. For example, if the MSRT has 50,000 executions in a particular location in the first quarter of the year and removes infections from 200 computers, the CCM for that location in the first quarter of the year is 4.0 (200 ÷ 50,000 × 1,000).

clean
To remove malware or potentially unwanted software from an infected computer. A single cleaning can involve multiple disinfections.

command and control
See *botnet*.

coordinated disclosure
The practice of disclosing vulnerabilities privately to an affected vendor so it can develop a comprehensive security update to address the vulnerability before it becomes public knowledge.

Definition
A set of *signatures* that can be used to identify malware using antivirus or antispyware products. Other vendors may refer to definitions as DAT files, pattern files, identity files, or antivirus databases.

denial of service (DoS)
A condition that occurs when the resources of a target computer are deliberately exhausted, effectively overwhelming the computer and causing it to fail to respond or function for its intended users. There are a number of different types of attack that may be used to result in a denial of service condition, utilizing different types of flood, or malformed network traffic. Also see *distributed denial of service (DDoS)*.

disclosure
Revelation of the existence of a vulnerability to a third party. Also see *responsible disclosure*.

disinfect
To remove a malware or potentially unwanted software component from a computer or to restore functionality to an infected program. Compare *clean*.

distributed denial of service (DDoS)
A form of denial of service (DoS) that uses multiple computers to attack the target. Considerable resources may be required to exhaust a target computer and cause it to fail to respond. Often multiple computers are used to perform these types of malicious attack and increase the attack's chances of success. This can occur, for example, when a number of compromised computers, such as those that comprise a botnet, are commandeered and ordered to access a target network or server over and over again within a small period of time.

downloader/dropper
See *trojan downloader/dropper*.

exploit
Malicious code that takes advantage of software vulnerabilities to infect a computer.

firewall
A program or device that monitors and regulates traffic between two points, such as a single computer and the network server, or one server to another.

generic
A type of signature capable of detecting a large variety of malware samples from a specific family, or of a specific type.

IFrame
Short for *inline frame*. An IFrame is an HTML document that is embedded in another HTML document. Because the IFrame loads another Web page, it can be used by criminals to place malicious HTML content, such as a script that downloads and installs spyware, into non-malicious HTML pages hosted by trusted Web sites.

in the wild
Said of malware that is currently detected in active computers connected to the Internet, as compared to those confined to internal test networks, malware research laboratories, or malware sample lists.

Internet Relay Chat (IRC)
A distributed real-time Internet chat protocol designed for group communication. Many botnets use the IRC protocol for C&C.

keylogger
See *password stealer (PWS)*.

Malicious Software Removal Tool
The Windows Malicious Software Removal Tool (MSRT) is designed to helpidentify and remove specifically targeted, prevalent malware from customer computers and is available at no charge to licensed Windows users. The main release mechanism of the MSRT is through Windows Update (WU), Microsoft Update (MU), or Automatic Updates (AU). A version of the tool is also available for download from the Microsoft Download Center. The MSRT is not a replacement for an up-to-date antivirus solution because the MSRT specifically targets only a small subset of malware families that are determined to be particularly prevalent. Further, the MSRT includes no real-time protection and cannot be used for the prevention of malware. More details about the MSRT are available at http://www.microsoft.com/security/malwareremove/default.mspx.

malware
Malicious software or potentially unwanted software installed without adequate user consent.

malware impression
A single instance of a user attempting to visit a page known to host malware and being blocked by the SmartScreen Filter in Internet Explorer 8. Also see *phishing impression*.

monitoring tool
Software that monitors activity, usually by capturing keystrokes or screen images. It may also include network sniffing software. Also see *password stealer (PWS)*.

P2P
See *peer-to-peer (P2P)*.

packet sniffer
A tool that logs traffic transmitted over a network. Packet sniffers are used by network administrators to maintain networks and diagnose problems, and by attackers to eavesdrop on sensitive or private information.

packer
A program that allows a user to package or bundle a file. This may be used by malware authors to obfuscate the structure of a malware file and thus avoid detection, as packing a single file using different packers results in different packages.

password stealer (PWS)
Malware that is specifically used to transmit personal information, such as user names and passwords. A PWS often works in conjunction with a *keylogger*, which sends keystrokes or screen shots to an attacker. Also see *monitoring tool*.

payload
The actions conducted by a piece of malware for which it was created. This can include, but is not limited to, downloading files, changing system settings, displaying messages, and logging keystrokes.

peer-to-peer (P2P)
A system of network communication in which individual nodes are able to communicate with each other without the use of a central server.

phishing
A method of credential theft that tricks Internet users into revealing personal or financial information online. Phishers use phony Web sites or deceptive e-mail messages that mimic trusted businesses and brands to steal personally identifiable information (PII), such as user names, passwords, credit card numbers, and identification numbers.

phishing impression
A single instance of a user attempting to visit a known phishing page, with Internet Explorer 7 or Internet Explorer 8, and being blocked by the Phishing Filter or SmartScreen Filter. Also see *malware impression*.

polymorphic
Said of malware that can mutate its structure to avoid detection by antivirus programs, without changing its overall algorithm or function.

potentially unwanted software
A program with potentially unwanted behavior that is brought to the user's attention for review. This behavior may impact the user's privacy, security, or computing experience.

remote control software
A program that provides access to a computer from a remote location. These programs are often installed by the computer owner or administrator and are only a risk if unexpected.

rogue security software
Software that appears to be beneficial from a security perspective, but provides limited or no security capabilities, generates a significant number of erroneous or misleading alerts, or attempts to socially engineer the user into participating in a fraudulent transaction.

rootkit
A program whose main purpose is to perform certain functions that cannot be easily detected or undone by a system administrator, such as hide itself or other malware.

signature
A set of characteristics that can identify a malware family or variant. Signatures are used by antivirus and antispyware products to determine if a file is malicious or not. Also see *definition*.

social engineering
A technique that defeats security precautions in place by exploiting human vulnerabilities. Social engineering scams can be both online (such as receiving e-mails that ask you to click the attachment, which is actually malware) and offline (such as receiving a phone call from someone posing as a representative from your credit card company). Regardless of the method selected, the purpose of a social engineering attack remains the same—to get the targeted user to perform an action of the attacker's choice.

spam
Bulk unsolicited e-mail. Malware authors may use spam to distribute malware, either by attaching the malware to the message or by sending a message containing a link to the malware. Malware may also harvest e-mail addresses for spamming from compromised machines or may use compromised machines to send spam.

spyware
A program that collects information, such as the Web sites a user visits, without adequate consent. Installation may be without prominent notice or without the user's knowledge.

SQL injection
A technique in which an attacker enters a specially crafted Structured Query Language (SQL) statement into an ordinary Web form. If form input is not filtered and validated before being submitted to a database, the malicious SQL statement may be executed, which could cause significant damage or data loss.

tool
Software that may have legitimate purposes but may also be used by malware authors or attackers.

trojan
A generally self-contained program that does not self-replicate but takes malicious action on the computer.

trojan downloader/dropper
A form of trojan that installs other malicious files to the infected system either by downloading them from a remote computer or by dropping them directly from a copy contained in its own code.

virus
Malware that replicates, commonly by infecting other files in the system, thus allowing the execution of the malware code and its propagation when those files are activated.

vulnerability
A weakness, error, or poor coding technique in a program that may allow an attacker to exploit it for a malicious purpose.

vulnerability broker
A company or other entity that provides software vendors with vulnerability information provided to it by external security researchers. In exchange for such compensation as the broker may provide, the security researchers agree not to disclose any information about the vulnerability to anyone other than the broker and the affected vendor.

wild
See *in the wild*.

worm
Malware that spreads by spontaneously sending copies of itself through e-mail or by using other communication mechanisms, such as instant messaging (IM) or peer-to-peer (P2P) applications.

zombie
See *botnet*.

Threat Families Referenced in This Report

The definitions for the threat families referenced in this report are adapted from the Microsoft Malware Protection Center encyclopedia (http://www.microsoft.com/security/portal), which contains detailed information about a large number of malware and potentially unwanted software families. See the encyclopedia for more in-depth information and guidance for the families listed here and throughout the report.

Win32/AgoBot: A backdoor that communicates with a central server using IRC.

Win32/Alureon: A data-stealing trojan that gathers confidential information such as user names, passwords, and credit card data from incoming and outgoing Internet traffic. It may also download malicious data and modify DNS settings.

Win32/Autorun: A worm that attempts to spread by being copied into all removable drives.

Win32/Bagle: A worm that spreads by e-mailing itself to addresses found on an infected computer. Some variants also spread through P2P networks. Bagle acts as a backdoor trojan and can be used to distribute other malicious software.

Win32/BaiduSobar: A Chinese-language Web browser toolbar that delivers pop-up and contextual advertisements, blocks certain other advertisements, and changes the Internet Explorer search page.

Win32/Bancos: A data-stealing trojan that captures online banking credentials and relays them to the attacker. Most variants target customers of Brazilian banks.

Win32/Banker: A family of data-stealing Trojans that captures banking credentials such as account numbers and passwords from computer users and relays them to the attacker. Most variants target customers of Brazilian banks; some variants target customers of other banks.

Win32/Banload: A family of trojans that download other malware. Banload usually downloads Win32/Banker, which steals banking credentials and other sensitive data and sends it back to a remote attacker.

Win32/Bifrose: A backdoor trojan that allows a remote attacker to access the compromised computer, and injects its processes into the Windows shell and Internet Explorer.

Win32/Bredolab: A downloader that is able to download and execute arbitrary files from a remote host.

Win32/Bubnix: A generic detection for a kernel-mode driver installed by other malware that hides its presence on an affected computer by blocking registry and file access to itself. The trojan may report its installation to a remote server and download and distribute spam email messages, and could download and execute arbitrary files.

Win32/CeeInject: A generic detection for malicious files that are obfuscated using particular techniques to protect them from detection or analysis.

Win32/Chadem: A trojan that steals password details from an infected computer by monitoring network traffic associated with FTP connections.

WinNT/Citeary: A kernel mode driver installed by Win32/Citeary, a worm that spreads to all available drives including the local drive, installs device drivers and attempts to download other malware from a predefined website.

Win32/Conficker: A worm that spreads by exploiting a vulnerability addressed by Security Bulletin MS08-067. Some variants also spread via removable drives and by exploiting weak passwords. It disables several important system services and security products, and downloads arbitrary files.

Win32/Cutwail: A trojan that downloads and executes arbitrary files, usually to send spam. Win32/Cutwail has also been observed to download the attacker tool Win32/Newacc.

Win32/FakeCog: A rogue security software family distributed under the names Defense Center, AntiMalware, and many others.

Win32/Fakeinit: A rogue security software family distributed under the names Internet Security 2010, Security Essentials 2010, and others.

Win32/FakeRean: A rogue security software family distributed under a large variety of randomly generated names, including Win 7 Internet Security 2010, Vista Antivirus Pro, XP Guardian, and many others.

Win32/FakeSpypro: A rogue security software family distributed under the names Antivirus System PRO, Spyware Protect 2009, and others.

Win32/FakeVimes: A rogue security software family distributed under the names Ultra Antivir 2009, Extra Antivirus, Virus Melt, and many others.

Win32/FakeXPA: A rogue security software family distributed under the names Antivirus 7, Personal Security, AntiVir2010, Antivirus BEST, Green AV, MaCatte, and many others.

Win32/FakeYak: A rogue security software family distributed under the names Antimalware Doctor and others.

Win32/FlyAgent: A backdoor trojan program that is capable of performing several actions depending on the commands of a remote attacker.

Win32/Frethog: A large family of password-stealing trojans that target confidential data, such as account information, from massively multiplayer online games.

Win32/Hamweq: A worm that spreads through removable drives, such as USB memory sticks. It may contain an IRC-based backdoor enabling the computer to be controlled remotely by an attacker.

Win32/Hotbar: Adware that displays a dynamic toolbar and targeted pop-up ads based on its monitoring of Web-browsing activity.

Win32/Hupigon: A family of trojans that uses a dropper to install one or more backdoor files, and installs sometimes a password stealer or other malicious programs.

Win32/IRCbot: A large family of backdoor trojans that drops other malicious software and connects to IRC servers to receive commands from attackers.

Win32/Koobface: A multi-component family of malware used to compromise computers and use them to perform various malicious tasks. It spreads through the internal messaging systems of popular social networking sites.

Win32/Lethic: A trojan that connects to remote servers, which may lead to unauthorized access to an affected system.

Win32/MoneyTree: A family of software that provides the ability to search for adult content on local disk. It may also install other potentially unwanted software, such as programs that display pop-up ads.

Win32/Nuwar: A family of trojan droppers that install a distributed P2P downloader trojan. This downloader trojan in turn downloads an e-mail worm component.

Win32/Obfuscator: A generic detection for programs that have had their purpose obfuscated to hinder analysis or detection by anti-virus scanners. They commonly employ a combination of methods, including encryption, compression, anti-debugging and anti-emulation techniques.

Win32/Oficla: A family of trojans that attempt to inject code into running processes in order to download and execute arbitrary files. It may download rogue security programs.

Win32/Parite: A family of viruses that infect .exe and .scr executable files on the local file system and on writeable network shares.

Win32/Pdfjsc: A family of specially crafted PDF files that exploit Adobe Acrobat and Adobe Reader vulnerabilities. These files contain malicious JavaScript that executes when the file is opened.

Win32/PrettyPark: A worm that spreads via email attachments. It allows backdoor access and control of an infected computer.

Win32/Prorat: A trojan that opens random ports that allow remote access from an attacker to the affected computer. This backdoor may download and execute other malware from predefined websites and may terminate several security applications or services.

Win32/Pushbot: A detection for a family of malware that spreads via MSN Messenger, Yahoo! Messenger and AIM when commanded by a remote attacker. It contains backdoor functionality that allows unauthorized access and control of an affected machine.

Win32/Randex: A worm that scans randomly generated IP addresses to attempt to spread to network shares with weak passwords. After the worm infects a computer, it connects to an IRC server to receive commands from the attacker.

Win32/Rbot: A family of backdoor trojans that allows attackers to control the computer through an IRC channel.

AutoIt/Renocide: A detection for a worm written in the AutoIt scripting language that exhibits backdoor behavior and attempts to download additional files from remote servers. It spreads via removable drives and network shares.

Win32/Renos: A family of trojan downloaders that install rogue security software.

Win32/Rimecud: A family of worms with multiple components that spreads via fixed and removable drives and via instant messaging. It also contains backdoor functionality that allows unauthorized access to an affected system.

Win32/RJump: A worm that attempts to spread by copying itself to newly attached media, such as USB memory devices or network drives. It also contains backdoor functionality that allows an attacker unauthorized access to an affected machine.

Win32/Rugzip: A trojan that downloads other malware from predefined Web sites. Rugzip may itself be installed by other malware. Once it has performed its malicious routines, it deletes itself to avoid detection.

Win32/Rustock: A multi-component family of rootkit-enabled backdoor trojans that were first developed around 2006 to aid in the distribution of spam email.

Win32/Sality: A family of polymorphic file infectors that target executable files with the extensions .scr or .exe. They may execute a damaging payload that deletes files with certain extensions and terminates security-related processes and services.

Win32/SDBot: A family of backdoor Trojans that allows attackers to control infected computers over an IRC channel.

Win32/Sdbot: A family of backdoor trojans that allows attackers to control infected computers. After a computer is infected, the trojan connects to an internet relay chat (IRC) server and joins a channel to receive commands from attackers.

Win32/Slenfbot: A family of worms that can spread via instant messaging programs, and may spread via removable drives. They also contain backdoor functionality that allows unauthorized access to an affected machine. This worm does not spread automatically upon installation, but must be ordered to spread by a remote attacker.

Win32/Small: A generic detection for a variety of threats.

Win32/Swif: A trojan that exploits a vulnerability in Adobe Flash Player to download malicious files. Adobe has published security bulletin APSB08-11 addressing the vulnerability.

Win32/Taterf: A family of worms that spread through mapped drives in order to steal login and account details for popular online games.

Win32/Tofsee: A multi-component family of backdoor trojans that act as a spam and traffic relay.

Win32/VBInject: A generic detection for obfuscated malware. The loader is written in Visual Basic and the malicious code, which may have virtually any purpose, is encrypted.

Win32/Virut: A family of file-infecting viruses that target and infect .exe and .scr files accessed on infected systems. Win32/Virut also opens a backdoor by connecting to an IRC server.

Win32/Waledac: A trojan that is used to send spam. It also has the ability to download and execute arbitrary files, harvest email addresses from the local machine, perform denial-of-service attacks, proxy network traffic, and sniff passwords

Win32/Winwebsec: A rogue security software family distributed under the names Winweb Security, System Security, and others.

Win32/Zbot: A family of password stealing trojans that also contains backdoor functionality allowing unauthorized access and control of an affected machine.

Win32/Zwangi: A program that runs as a service in the background and modifies Web browser settings to visit a particular Web site.