

Microsoft Security Intelligence Report

Volume 11

*An in-depth perspective on
software vulnerabilities and exploits,
malicious code threats, and
potentially unwanted software
in the first half of 2011*

Microsoft®

Microsoft Security Intelligence Report

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Copyright © 2011 Microsoft Corporation. All rights reserved.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Authors

Joe Faulhaber
Microsoft Malware Protection
Center

David Felstead
Bing

Paul Henry
Wadeware LLC

Jeff Jones
Microsoft Trustworthy
Computing

Ellen Cram Kowalczyk
Microsoft Trustworthy
Computing

Jimmy Kuo
Microsoft Malware Protection
Center

John Lambert
Microsoft Security
Engineering Center

Marc Lauricella
Microsoft Trustworthy
Computing

Aaron Margosis
Microsoft Public Sector
Services

Michelle Meyer
Microsoft Trustworthy
Computing

Anurag Pandit
Windows Live Safety
Platform

Anthony Penta
Windows Live Safety
Platform

Dave Probert
Microsoft Security
Engineering Center

Tim Rains
Microsoft Trustworthy
Computing

Mark E. Russinovich
Microsoft Technical Fellow

Weijuan Shi
Windows Business Group

Adam Shostack
Microsoft Trustworthy
Computing

Frank Simorjay
Microsoft Trustworthy
Computing

Hemanth Srinivasan
Microsoft Malware Protection
Center

Holly Stewart
Microsoft Malware Protection
Center

Matt Thomlinson
Microsoft Security Response
Center

Jeff Williams
Microsoft Malware Protection
Center

Scott Wu
Microsoft Malware Protection
Center

Terry Zink
Microsoft Forefront Online
Protection for Exchange

Contributors

Roger Capriotti
Windows Live Safety
Platform

Doug Cavit
Microsoft Trustworthy
Computing

**CSS Japan Security
Response Team**
Microsoft Japan

Dave Forstrom
Microsoft Trustworthy
Computing

Eric Foster
Windows Live Safety
Platform

Enrique Gonzalez
Microsoft Malware Protection
Center

Heather Goudey
Microsoft Malware Protection
Center

Vinny Gullotto
Microsoft Trustworthy
Computing

Satomi Hayakawa
CSS Japan Security Response
Team

Forbes Higman
Windows Live Safety
Platform

Yuhui Huang
Microsoft Malware Protection
Center

Aaron Hulett
Microsoft Malware Protection
Center

Hilda Larina Rraggio
Microsoft Malware Protection
Center

Eric Lawrence
Windows Live Safety
Platform

Ken Malcolmson
Microsoft Trustworthy
Computing

Takumi Onodera
Microsoft Premier Field
Engineering, Japan

Daryl Pecelj
Microsoft IT Information
Security and Risk
Management

Kathy Phillips
Microsoft Legal and
Corporate Affairs

Tareq Saade
Microsoft Malware Protection
Center

Richard Saunders
Microsoft Trustworthy
Computing

Jasmine Sesso
Microsoft Malware Protection
Center

Norie Tamura
CSS Japan Security Response
Team

Matt Thomlinson
Microsoft Trustworthy
Computing

Patrik Vicol
Microsoft Malware Protection
Center

Steve Wacker
Wadeware LLC



Table of Contents

About This Report	ix
Trustworthy Computing: Security Engineering at Microsoft	x
Key Findings Summary	xi
Zeroing In on Malware Propagation Methods	1
Background.....	3
Analysis and Results.....	5
A New Method for Classifying Malware Propagation.....	5
Data Used	6
Analytic Methods	7
Results.....	10
Insights	12
User Interaction.....	13
Feature Abuse.....	13
Exploit Age.....	14
Zero-Day Exploits: A Supplemental Analysis	14
Analysis Details.....	17
The Project Broad Street Taxonomy	17
Using the Taxonomy	17
Vulnerability Subprocess	20
Methodology Details	21

Other classifications of malware.....	22
Conclusion	24
Call to Action	24
Advice to IT Professionals on Social Engineering	25
Organizations	25
Software.....	27
People.....	27
Worldwide Threat Assessment	29
Vulnerabilities.....	31
Industry-Wide Vulnerability Disclosures	31
Vulnerability Severity.....	32
Vulnerability Complexity.....	34
Operating System, Browser, and Application Vulnerabilities.....	35
Microsoft Vulnerability Disclosures.....	36
Guidance: Developing Secure Software	37
Exploits.....	38
Java Exploits	40
HTML and JavaScript Exploits	41
Document Parser Exploits.....	42
Microsoft Office File Format Exploits	43
Operating System Exploits.....	45
Adobe Flash Player Exploits	47
Malware and Potentially Unwanted Software.....	49
CCM Calculation Changes	49
Global Infection Rates	51
Regional Effective Practices.....	56
Operating System Infection Rates	57

Threat Categories	60
Threat Categories By Location	61
Threat Families	63
Rogue Security Software.....	64
Home and Enterprise Threats	66
Guidance: Defending Against Malware	70
Email Threats.....	71
Spam Messages Blocked.....	71
Spam Types.....	73
Guidance: Defending Against Threats in Email	75
Malicious Websites	76
Phishing Sites	77
Target Institutions	79
Global Distribution of Phishing Sites	81
Malware Hosting Sites	83
Malware Categories.....	84
Global Distribution of Malware Hosting Sites	87
Drive-By Download Sites	89
Guidance: Protecting Users from Unsafe Websites	91
 Managing Risk	 93
Protecting Organizations, Software, and People	95
Advanced Malware Cleaning Techniques for the IT Professional.....	96
Step 1: Disconnect from the Network.....	97
Step 2: Identify Malicious Processes and Drivers	97
Using Process Explorer	98
Tracing Malware	106
Step 3: Terminate Malicious Processes.....	108

Step 4: Identify and Delete Malware Autostarts	109
Using Autoruns	109
Step 5: Delete Malware Files.....	111
Steps 6 and 7: Reboot and Repeat.....	111
Conclusion	112
Promoting Safe Browsing	113
SmartScreen Filter	114
ActiveX Filtering.....	115
Cross-site scripting filter.....	115
Other browser defenses.....	115
Group Policy and the Security Compliance Manager	117
Appendixes	119
Appendix A: Threat Naming Conventions.....	120
Appendix B: Data Sources.....	122
Appendix C: Worldwide Infection Rates.....	124
Appendix D: Microsoft Office Vulnerabilities Encountered in 1H11	131
Glossary	132
Threat Families Referenced in This Report	137

About This Report

Scope

The *Microsoft® Security Intelligence Report (SIR)* focuses on software vulnerabilities, software vulnerability exploits, malicious and potentially unwanted software, and security breaches. 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 organizations, software, and users.

Reporting Period

This volume of the *Microsoft Security Intelligence Report* focuses on the first and second quarters of 2011, with trend data for the last several years presented on a quarterly basis. Because vulnerability disclosures can be highly inconsistent from quarter to quarter and often occur disproportionately at certain times of the year, statistics about vulnerability disclosures are presented on a half-yearly basis, as in previous volumes of the report.

Throughout the report, half-yearly and quarterly time periods are referenced using the *nHyy* or *nQyy* formats, respectively, where *yy* indicates the calendar year and *n* indicates the half or quarter. For example, 1H11 represents the first half of 2011 (January 1 through June 30), and 2Q11 represents the second quarter of 2011 (April 1 through June 30). To avoid confusion, please 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.

Trustworthy Computing: Security Engineering at Microsoft

Amid the increasing complexity of today's computing threat landscape and the growing sophistication of criminal attacks, enterprise organizations and governments are more focused than ever on protecting their computing environments so that they and their constituents are safer online. With more than a billion systems using its products and services worldwide, Microsoft collaborates with partners, industry, and governments to help create a safer, more trusted Internet.

Trustworthy Computing (TwC), formed in 2002, is Microsoft's commitment to creating and delivering secure, private, and reliable computing experiences based on sound business practices. Most of the intelligence provided in this report comes from Trustworthy Computing security centers—the Microsoft Malware Protection Center (MMPC), Microsoft Security Response Center (MSRC), and Microsoft Security Engineering Center (MSEC)—which deliver in-depth threat intelligence, threat response, and security science. Additional information comes from product groups across Microsoft and from Microsoft IT (MSIT), the group that manages global IT services for Microsoft. The report is designed to give Microsoft customers, partners, and the software industry a well-rounded understanding of the threat landscape so that they will be in a better position to protect themselves and their assets from criminal activity.



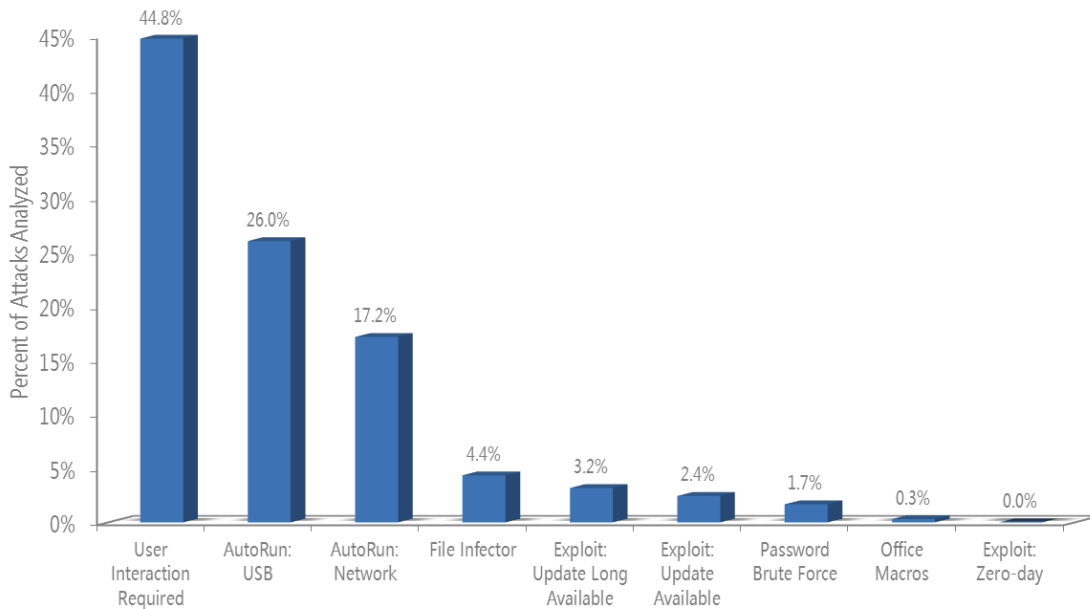
Key Findings Summary



Zeroing in on Malware Propagation Methods

Microsoft conducted an analysis to better understand the frequency of zero-day exploitation and the risks customers face from it. This analysis was created to give security professionals information they can use to prioritize their concerns and effectively manage risks. Like everyone else, IT departments face constraints of time, budget, personnel, and resources when planning and performing their work. Having accurate, up-to-date information about the threat landscape enables security professionals to effectively prioritize their defenses and help keep their networks, software, and people safe.

For the analysis, threats detected by the Malicious Software Removal Tool (MSRT) during the first half of 2011 (1H11) were classified by the means of propagation that each threat family has been documented to use to infect victims. If the threat was reported as using multiple vectors to infect users, then the number of infections reported by the MSRT for that family were divided and attributed equally to each vector. The figure on the next page shows the results of that analysis.

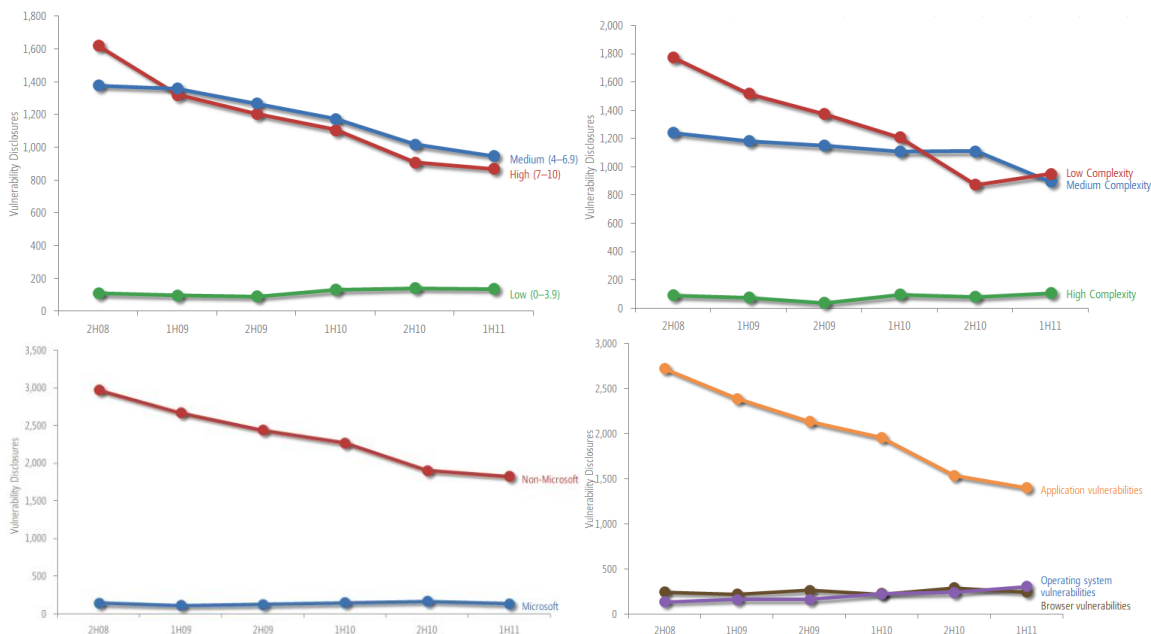


- The different malware threat propagation methods referenced in the figure are described as follows:
 - **User Interaction Required.** When a user has to perform an action for the computer to be compromised. In this usage, “action” means an intentional action that is in some way distinguished from typical use of the computer.
 - **AutoRun: USB.** The threat takes advantage of the AutoRun feature in Windows to infect USB storage devices and other removable volumes.
 - **AutoRun: Network.** The threat takes advantage of the AutoRun feature to infect network volumes mapped to drive letters.
 - **File Infector.** The threat spreads by modifying files, often with .exe or .scr extensions, by rewriting or overwriting some code segments.
 - **Exploit: Update Long Available.** The vendor released a security update to address the vulnerability more than a year before the attack.
 - **Exploit: Update Available.** The vendor released a security update to address the vulnerability less than a year before the attack.
 - **Exploit: Zero-day.** The vendor had not released a security update to address the vulnerability at the time of the attack.

- **Password Brute Force.** The threat spreads by attempting brute force password attacks on available volumes, as with the `net use` command.
- **Office Macros.** The threat spreads by infecting Microsoft Office documents with malicious Visual Basic® for Applications (VBA) macros.
- More than a third of malware detections that were analyzed were attributed to malicious software that misused the AutoRun feature in Windows®.
 - Threats that misused AutoRun were split between those that spread via removable volumes (26 percent of the total) and those that spread via network volumes (17 percent).
 - To combat these threats, Microsoft took several steps to help protect customers, including releasing an automatic update for the Windows XP and Windows Vista® platforms in February 2011 to make the Autorun feature more secure, as it is by default in Windows 7.
- About six percent of the MSRT detections that were analyzed were attributed to *exploits*—malicious code that attempts to exploit vulnerabilities in applications or operating systems.
- None of the top families in the MSRT were documented as using zero-day exploits in 1H11.
- Out of all the vulnerability exploitation detected by the MMPC, less than one percent was zero-day exploit activity.

Worldwide Threat Assessment

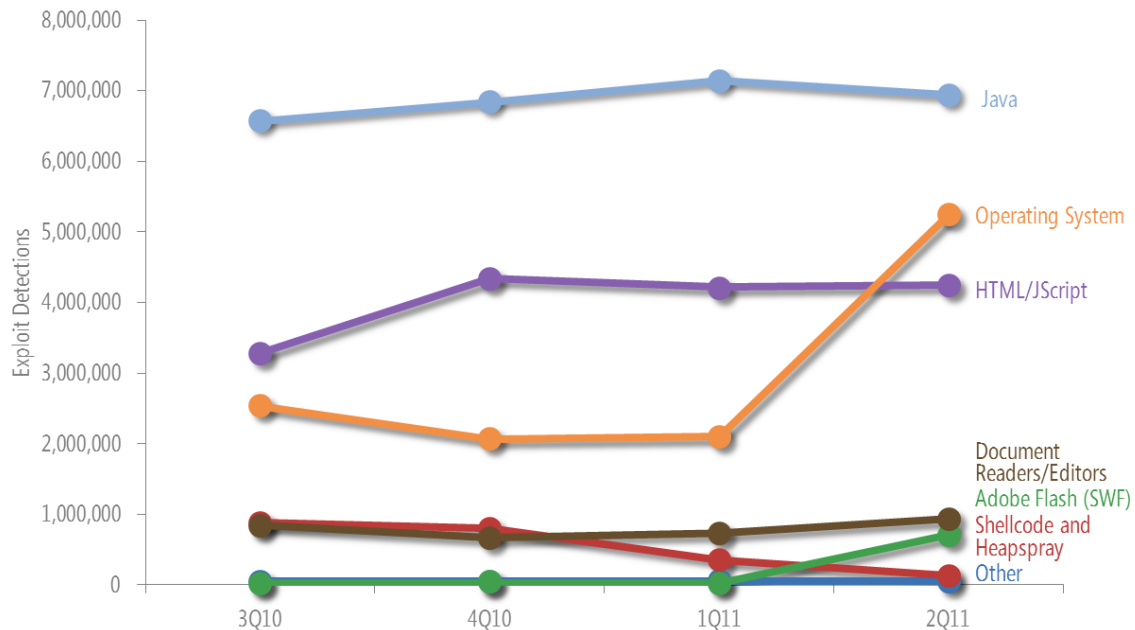
Vulnerability Disclosures



- The overall vulnerability severity trend (as determined by Common Vulnerabilities and Exposures, or CVE, number) has been a positive one. Medium and High severity vulnerabilities disclosed in 1H11 were down 6.8 percent and 4.4 percent from 2H10, respectively.
- Low complexity vulnerabilities—the easiest ones to exploit—were down 41.2 percent from the prior 12-month period.
- Operating system and browser vulnerability disclosures have been mostly stable for several years, accounting for 12.7 percent and 15.7 percent of all vulnerabilities disclosed in 1H11, respectively.
- Vulnerabilities in Microsoft products accounted for 6.9 percent of all vulnerabilities disclosed in 1H11, down from 8.2 percent in 2H10.

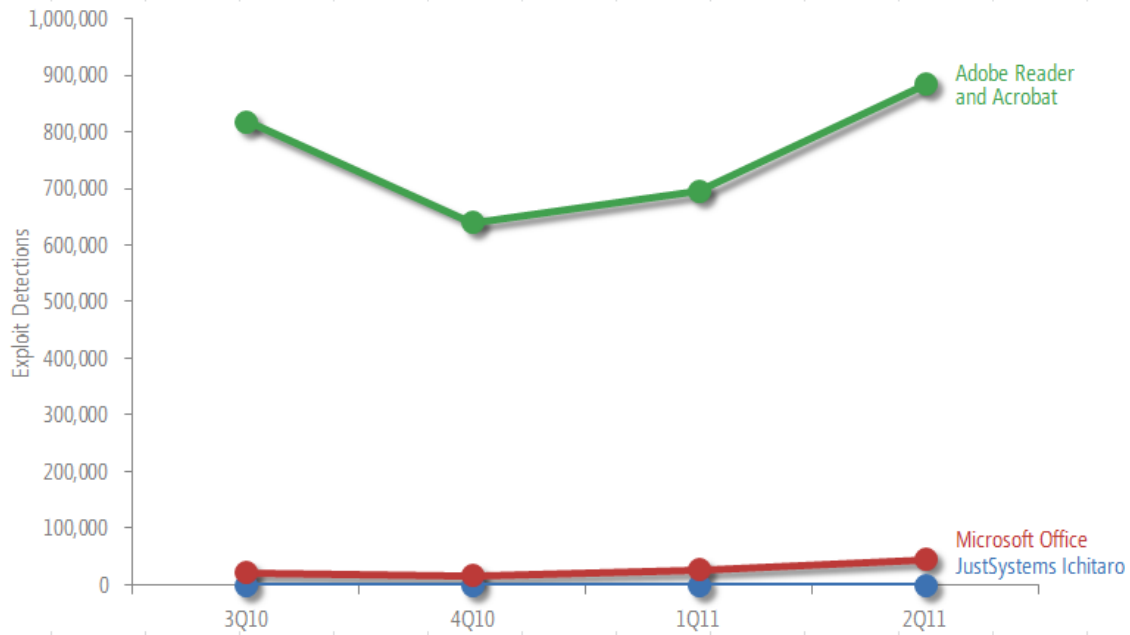
Exploits

The next figure shows the prevalence of different types of exploits for each quarter between 3Q10 and 2Q11.



- The most commonly observed types of exploits in 1H11 were those targeting vulnerabilities in the Oracle (formerly Sun) Java Runtime Environment (JRE), Java Virtual Machine (JVM), and Java SE in the Java Development Kit (JDK). Java exploits were responsible for between one-third and one-half of all exploits observed in each of the four most recent quarters.
- Detections of operating system exploits increased dramatically in 2Q11 because of increased exploitation of vulnerability [CVE-2010-2568](#).
- Detections of exploits targeting Adobe Flash, although uncommon in comparison to some other types of exploits, increased in 2Q11 to more than 40 times the volume seen in 1Q11 because of exploitation of a pair of newly-discovered vulnerabilities.
- Exploits that target [CVE-2010-2568](#), a vulnerability in Windows Shell, increased significantly in 2Q11, and were responsible for the entire 2Q11 increase in operating system exploits. The vulnerability was first discovered being used by the family [Win32/Stuxnet](#) in mid-2010.

Document Exploits



- Exploits that affected Adobe Acrobat and Adobe Reader accounted for most document format exploits detected in the first half of 2011. Almost all of these exploits involved the generic exploit family [Win32/Pdfjsc](#).
- More than half of Microsoft Office exploits involved [CVE-2010-3333](#), a vulnerability in the Rich Text Format (RTF) parser in versions of Microsoft Word.

Malware and Potentially Unwanted Software

Except where specified, the information in this section was compiled from telemetry data that was generated from more than 600 million computers worldwide and some of the busiest online services on the Internet. Infection rates are given in *computers cleaned per mille (CCM)*, or thousand, and represent the number of reported computers cleaned in a quarter for every 1,000 executions of the Malicious Software Removal Tool. See the “[Malware](#)” section of the *Microsoft Security Intelligence Report* website for more information about the CCM metric.

Operating System Infection Rates

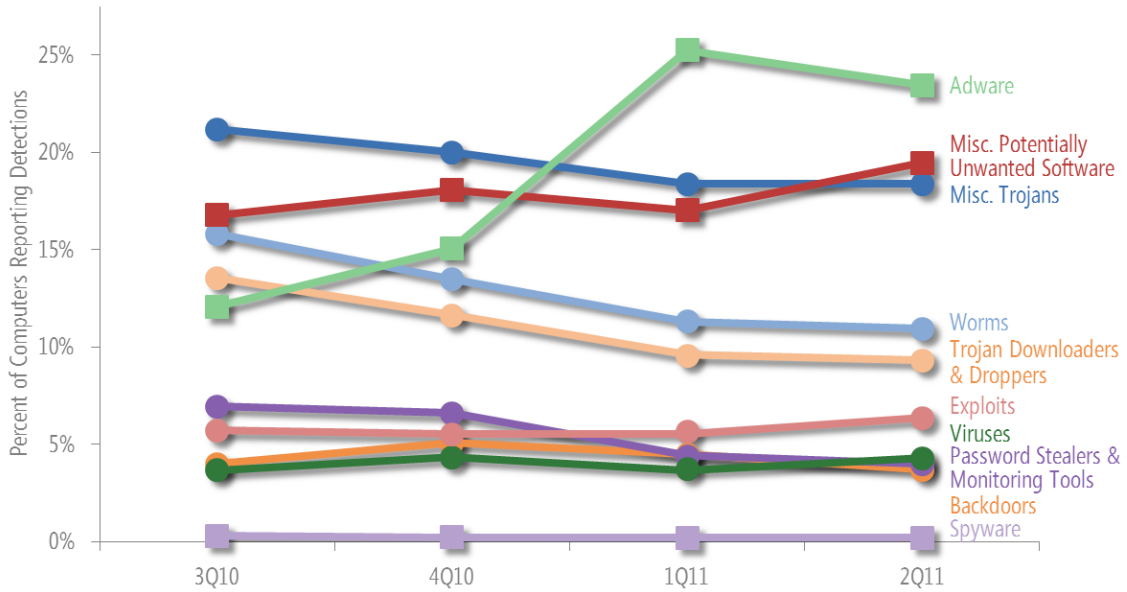


“32” = 32-bit edition; “64” = 64-bit edition. SP = Service Pack. Supported operating systems with at least 0.1 percent of total executions in 2Q11 shown...

- As in previous periods, infection rates for more recently released Microsoft operating systems and service packs are consistently lower than older ones, for both client and server platforms. Windows 7 and Windows Server® 2008 R2, the most recently released Windows client and server versions, respectively, have the lowest infection rate, as shown in the figure.
- Infection rates for Windows XP SP3 and Windows Vista declined following the February 2011 release of an automatic update that changed the way the AutoRun feature works on those platforms to match its functionality in Windows 7. The impact of this change can be seen in the infection statistics

for Win32/Rimecud, the ninth most commonly detected threat family worldwide in 1H11 and one of the top abusers of the AutoRun feature.

Threat Families and Categories



Round markers indicate malware categories; square markers indicate potentially unwanted software categories.

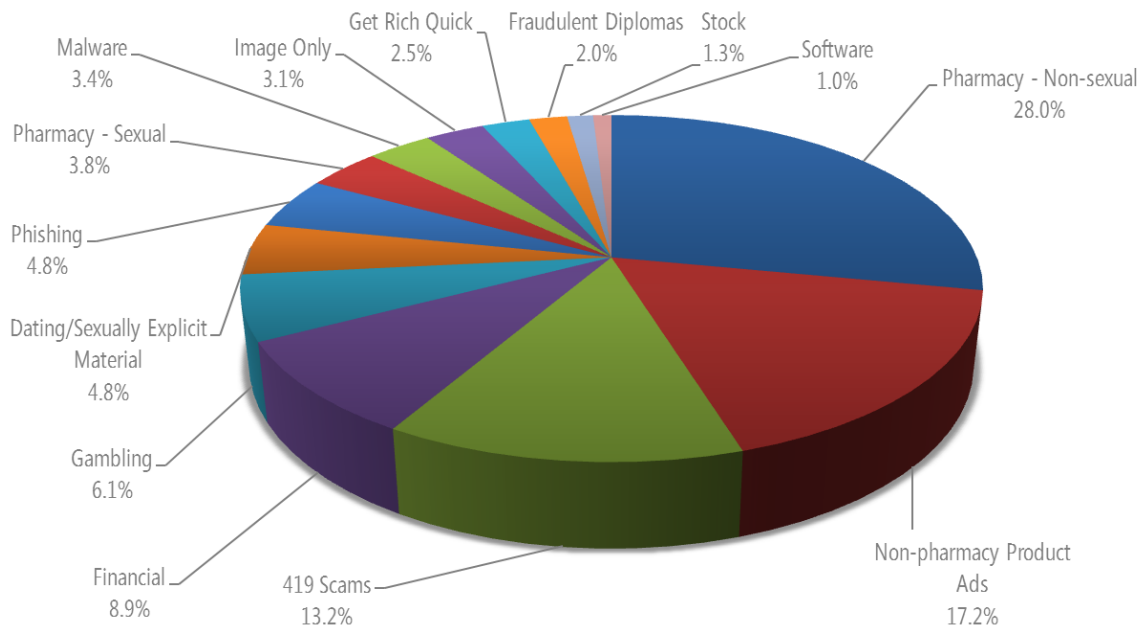
- [Win32/OpenCandy](#) was the most commonly detected threat family in 1H11 overall. OpenCandy is an adware program that might be bundled with certain third-party software installation programs.
- [JS/Pornpop](#), the second most commonly detected threat family in 1H11 overall, is a detection for specially crafted JavaScript-enabled objects that attempt to display pop-under advertisements in users' web browsers
- [Win32/Hotbar](#), the most commonly detected threat family in 2Q11 and the third most commonly detected family in 1H11, is adware that installs a browser toolbar that displays targeted pop-up ads based on its monitoring of web browsing activities.
- Detections of [Win32/FakeRean](#) increased more than 300 percent from 1Q11 to 2Q11 to become the most commonly detected rogue security software family of the second quarter.

Enterprise Threats

- Worm families accounted for the three most common malware families detected on domain-joined computers, which are more common in enterprise environments than in home environments.
- Malware families that are significantly more prevalent on domain-joined computers include [Win32/Conficker](#) and the potentially unwanted software program [Win32/RealVNC](#). RealVNC is a program that enables a computer to be controlled remotely, similar to Remote Desktop Services. It has a number of legitimate uses, but attackers have also used it to gain control of users' computers for malicious purposes.
- The virus family Win32/Sality, which was not among the top 10 families detected on domain-joined computers in 2010, ranks tenth in 1H11.

Email Threats

- The volume of spam blocked by Microsoft Forefront® Online Protection for Exchange (FOPE) decreased dramatically over the past 12 months, from 89.2 billion messages in July 2010 to 25.0 billion in June 2011, primarily because of takedowns of two major botnets: Cutwail, which was shut down in August 2010, and Rustock, which was shut down in March 2011 following a period of dormancy that began in January.
- As in previous periods, advertisements for nonsexual pharmaceutical products (28.0 percent of the total) and nonpharmaceutical product advertisements (17.2 percent) accounted for the majority of the spam messages blocked by FOPE content filters in 1H11.
- Image-only spam messages declined to 3.1 percent of the total in 1H11, down from 8.7 percent in 2010.



Malicious Websites

- Phishers have traditionally targeted financial sites more than other types of sites, but the largest share of phishing impressions in 1H11 was for sites that targeted social networks, reaching a high of 83.8 percent of impressions in April. (A phishing impression is a single instance of a user attempting to visit a known phishing site with Windows Internet Explorer® and being blocked by SmartScreen® Filter. See the “[Malicious Websites](#)” section of the Microsoft Security Intelligence Report website for more information.) Overall, impressions that targeted social networks accounted for 47.8 percent of all impressions in 1H11, followed by those that targeted financial institutions at 35.0 percent.
- By contrast, phishing sites that targeted financial institutions accounted for an average of 78.3 percent of active phishing sites tracked each month in 1H11, compared to just 5.4 percent for social networks. Financial institutions targeted by phishers can number in the hundreds, and customized phishing approaches are required for each one. The number of popular social networking sites is much smaller, so phishers who target social networks can effectively target many more people per site. Still, the potential for direct illicit access to victims’ bank accounts means that financial institutions remain perennially popular phishing targets, and they continue to receive the largest or second-largest number of impressions each month.

- This phenomenon also occurs on a smaller scale with online services and gaming sites. A small number of online services account for the majority of traffic to such sites, so phishing sites that targeted online services garnered 11.0 percent of impressions with just 3.6 percent of sites. Online gaming traffic tends to be spread out among a larger number of sites, so phishing sites that targeted online gaming destinations accounted for 8.9 percent of active sites but gained just 4.3 percent of impressions.
- Phishing sites that targeted e-commerce were responsible for just 3.8 percent of active sites and 1.9 percent of impressions, which suggests that phishers have not found e-commerce sites to be particularly profitable targets.

Information on [Protecting Your Organization, Software, and People](#) can be found in the “Managing Risk” section of the *Microsoft Security Intelligence Report* website (www.microsoft.com/sir).



Zeroing In on Malware Propagation Methods



Background

Among the array of technical and non-technical mechanisms that malicious parties have at their disposal for attacking computers and stealing data, the *zero-day vulnerability*—a software vulnerability that is successfully exploited before the software vendor has published a security update to address it—is especially significant for security professionals and attackers alike. Zero-day vulnerabilities—according to conventional wisdom, at least—cannot be effectively defended against, and can arise at any time, leaving even security-conscious IT administrators essentially at their mercy. Although technologies such as Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR) have been introduced to make it more difficult to reliably exploit software, and processes such as the Secure Development Lifecycle (SDL) have been shown to reduce the incidence of software vulnerabilities, zero-day vulnerabilities continue to capture the imagination.

The zero-day vulnerability is especially alarming for consumers and IT professionals, and for good reason—it combines fear of the unknown and an inability to fix the vulnerability, which leaves users and administrators feeling defenseless. It's no surprise that zero-day vulnerabilities often receive considerable coverage in the press when they arise, and can be treated with the utmost level of urgency by the affected vendor and the vendors' customers.

Despite this level of concern, there has been little measurement of the zero-day threat in the context of the broader threat landscape. This section of the *Microsoft Security Intelligence Report* presents such an analysis, along with details of the methodology used, a discussion of the insights gained from it, and some information about what's been done with those insights.

This analysis approaches its subject in two ways. First, it establishes a method to estimate how malware propagates, including the use of zero-day exploits. Second, it measures the amount of zero-day exploitation in comparison with overall vulnerability exploitation. In other words, what are the relative proportions of exploitation before and after the update?

This analysis was undertaken for a number of reasons. Microsoft is always seeking better statistics about the frequency of zero-day exploitation and the risk

customers face from it. Also, Microsoft frequently fields questions about zero-day vulnerabilities from a variety of interested parties, ranging from journalists to IT security professionals. It is important to provide timely and accurate answers for such questions, but also help put them in perspective relative to other threats in the greater security landscape. In a more general sense, it serves everyone—IT and security professionals as well as consumers—to have realistic models of the way malware spreads in today’s world. At a time when effective cooperation and coordination of security efforts across corporate and political borders is as important as it has ever been, it is only through an accurate shared understanding of the threats all users face that IT and security pros can create the most effective defense.

One important goal of this analysis is to provide security professionals with information they can use to prioritize their concerns and effectively manage risks. Like everyone else, IT departments face constraints of time, budget, personnel, and resources when planning and performing their work. Having accurate, up-to-date information about the threat landscape enables security professionals to effectively prioritize their defenses and help keep their networks, software, and people safe.

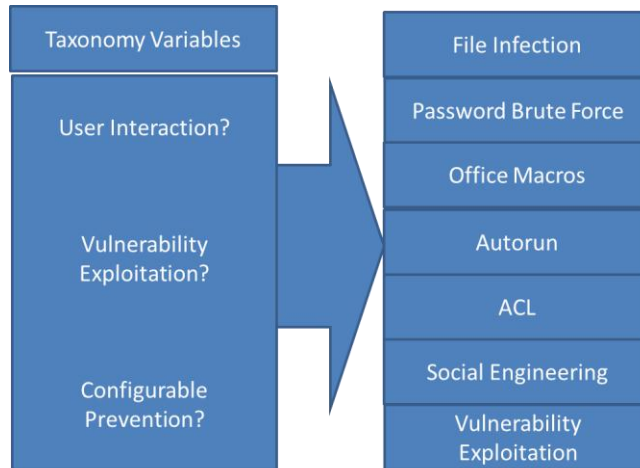
Analysis and Results

To better understand the landscape, Microsoft researchers have drawn on current information about trends and developments in malware creation and distribution to develop a new taxonomy for classifying malware according to the methods it uses to spread. Applying this taxonomy to telemetry data generated by security products has provided insights into the ways attackers distribute malware.

A New Method for Classifying Malware Propagation

The analysis presented here is in part an effort to start a conversation within the industry about the current state of malware analysis and classification. Many of the de facto standards that security professionals use were originally formulated when the threat landscape was very different than it is today. These standards were created when widespread public use of the Internet was nonexistent or very limited, and before malware development and propagation were the domain of professional criminals looking for illegitimate profits. Many of these standards and beliefs evolved chaotically over a period of years, and in some cases terms were never especially well defined. By adding new ways to classify malware and understand how exploitation is measured, security professionals can improve the ways they think and communicate about the threats that modern computer users face. This analysis is not a call to throw away current approaches, but rather a new lens that has been shown to be helpful.

Figure 1. Classifying malware according to propagation methods



The framework sketched in Figure 1 that classifies malware families by the methods—both technical and non-technical—that they use to propagate was developed as part of this analysis. In this context, propagation refers to the crucial moment when the attacker is first running software on a computer. “Insights,” beginning on page 12, provides an overview of this taxonomy; an in-depth explanation begins on page 17.

As with any taxonomy, adaptation is a natural progression. As a lesson learned from past malware categorization, this taxonomy should not be considered definitive. On the contrary, the researchers are enthusiastic about presenting its current form and look forward to the community dialogue that is sure to result as it evolves.

Data Used

To apply this taxonomy to infection data, Microsoft researchers analyzed infections reported by the [Microsoft Malicious Software Removal Tool \(MSRT\)](#) during the first half of 2011. The MSRT is a free tool that Microsoft designed to help identify and remove selected prevalent malware families from Windows-based computers. A new version of the MSRT is released each month and distributed through Windows® Update, Microsoft Update, and the Microsoft Download Center.

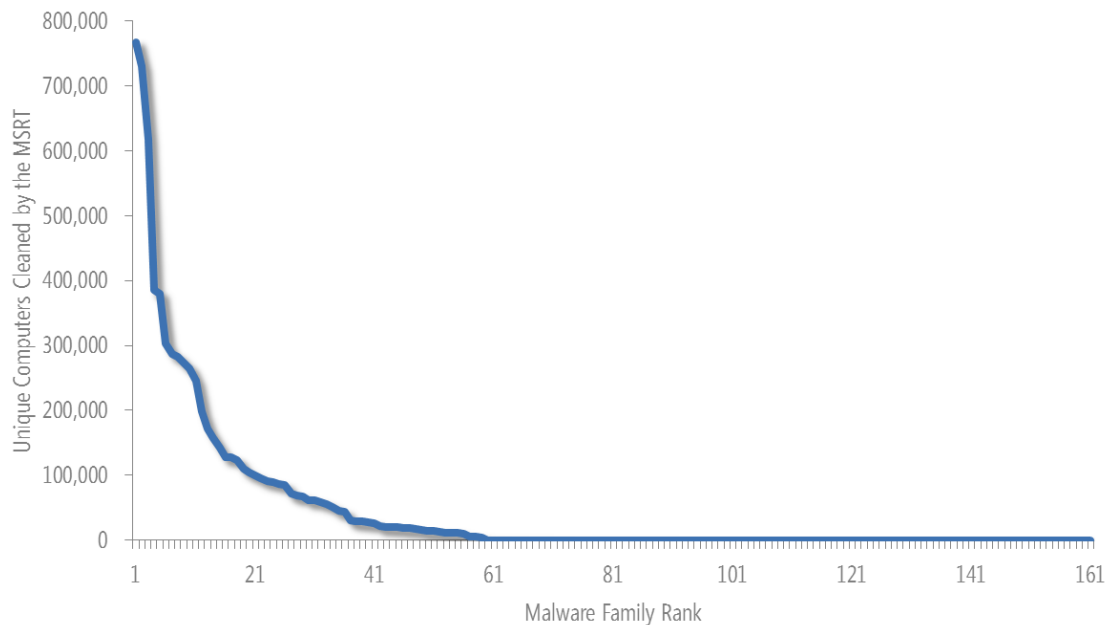
The MSRT was selected as the data source for this exercise for several reasons:

- The MSRT runs on more than 600 million individual computers around the world each month.
- The MSRT specifically targets malware families that present a severe risk to users or are particularly prevalent.
- MSRT data represents infected computers (as opposed to infection attempts that were blocked by real-time protection products).
- Installations of the MSRT are strongly correlated with usage of Windows Update and Microsoft Update, the tool's primary distribution mechanisms, which helps provide a reasonably accurate picture of the risks faced by computers that likely apply regular security updates.

Analytic Methods

Malware infections tend to resemble a power law distribution, as shown in Figure 2, in which a few dozen malware families account for most infections and a “long tail” consisting of a large number of less common families account for the rest.

Figure 2. Malware families detected by the MSRT, ranked by the number of computers each family was removed from in the second quarter of 2011 (“2Q11”)



To allow for a thorough analysis of infection methods for a significant portion of the malware landscape, this analysis focuses on the 27 malware families detected most often by the MSRT in the first half of 2011, which together accounted for a

majority of total MSRT detections.¹ To classify these malware families for analysis, the researchers investigated the mechanisms by which each of the families has been documented to spread, using information from the MMPC malware encyclopedia as well as other sources. Only mechanisms used actively by each family to spread were considered; The mechanisms used by these families were grouped into nine separate categories. (See “Insights” beginning on page 12 for more information about this classification scheme.)

Many families use multiple mechanisms to propagate. When malware is detected on a computer, the actual method of infection is very difficult to determine without performing forensic work on each computer. Therefore, to analyze infections on hundreds of thousands of computers, some assumptions are necessary.

To compensate for the difficulty in determining the exact propagation mechanism used in each case, an “equal buckets” approach was used in which detections of these families were allocated equally among each category in which they were known to spread. For example, [Win32/Conficker](#) spreads by exploiting a vulnerability ([CVE-2008-4250](#), addressed by Security Bulletin [MS08-067](#)), by taking advantage of AutoRun on both mapped drives and removable ones, and by using a password dictionary. Using this approach, 100 Conficker infections is translated into 25 vulnerability-related propagations and 75 in feature abuse (25 each for AutoRun USB, AutoRun network, and password brute force activity).

Families that were determined to spread via exploits were classified according to the age of the security update addressing the vulnerability at the time of analysis:

- **Zero-day.** The exploit is known to have existed in the wild before the vendor could publish a security update to address the related vulnerability. If the exploit was zero-day at any time during the month-long period preceding the release of the MSRT version that detected it, it is considered a zero-day exploit for the purposes of this analysis.
- **Update Available.** The security update that addresses the vulnerability was first issued less than a year before the recorded detection.
- **Update Long Available.** The security update that addresses the vulnerability was first issued more than a year before the recorded detection.

¹The analysis included all malware families detected on computers at least 25,000 times. The families listed here accounted for 83 percent of all MSRT detections for the 6-month period.

For example, security bulletin MS08-067, which addressed the vulnerability exploited by Conficker, was released in October 2008, so Conficker is now listed in the “Update Long Available” category.

Figure 3 lists the malware families included in this analysis and shows how they were classified.

Figure 3. Some of the top malware families detected by the MSRT in 1H11 and their propagation methods

Family	Exploit: Zero-day	Exploit: Update Avail.	Exploit: Update Long Avail.	AutoRun (Net.)	AutoRun (USB)	Office Macro	Passwd. Brute Force	User Interaction	File Infector
Win32/Alureon		•						•	
Win32/Bancos								•	
Win32/Bredolab			•						
Win32/Brontok					•			•	
Win32/Bubnix								•	
Win32/Conficker			•	•	•		•		
Win32/Cutwail								•	
Win32/Cycbot			•					•	
Win32/FakeRean								•	
Win32/FakeSpypro								•	
Win32/FakeXPA								•	
Win32/Frethog				•				•	
Win32/Hamweq					•				
Win32/Jeefo									•
Win32/Lethic								•	
Win32/Parite									•
Win32/Pushbot			•		•			•	
Win32/Ramnit				•	•	•			•
Win32/Randex							•		
Win32/Renocide				•	•			•	

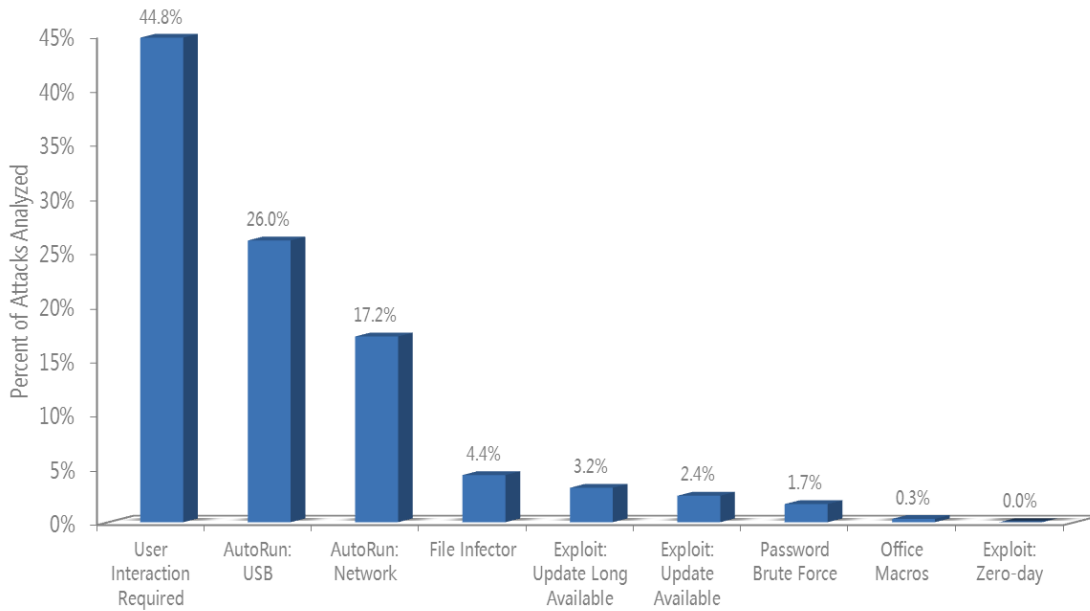
Figure 3 (continued). Some of the top malware families detected by the MSRT in 1H11 and their propagation methods

Family	Exploit: Zero-day	Exploit: Update Avail.	Exploit: Update Long Avail.	AutoRun (Net.)	AutoRun (USB)	Office Macro	Passwd. Brute Force	User Interaction	File Infector
Win32/Renos								•	
Win32/Rimecud				•	•			•	
Win32/Sality				•					•
Win32/Taterf				•	•				
Win32/Vobfus			•	•	•				
Win32/Yimfoca								•	
Win32/Zbot		•	•					•	

Results

Figure 4 shows the results of this analysis.

Figure 4. Malware detected by the MSRT in 1H11, by means of propagation ability



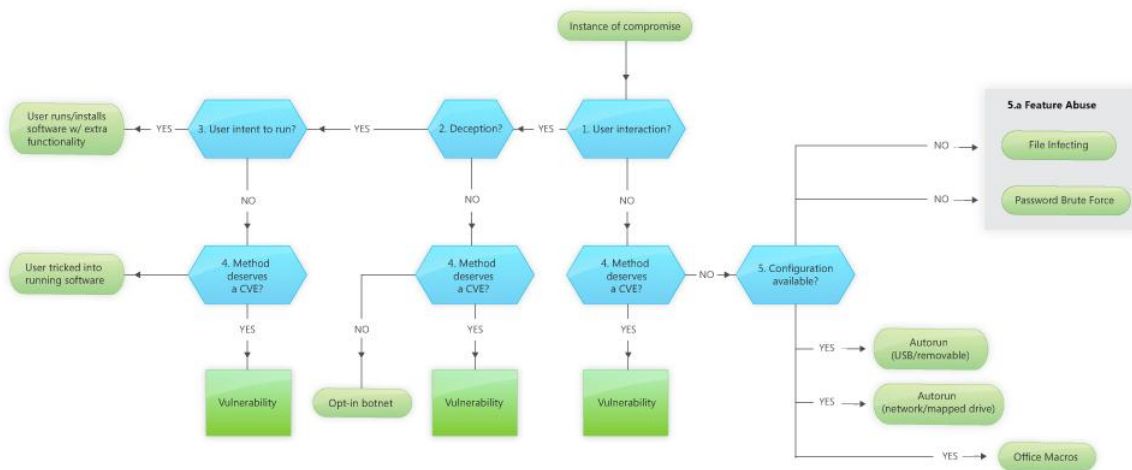
- Threats that are documented as relying on user interaction to spread account for 45 percent of attacks analyzed.

- More than a third of the detections that were analyzed were caused by malicious software that misused the AutoRun feature in Windows. Analyzed threats were split between USB AutoRun threats (26 percent of the total) and network volume AutoRun threats (17 percent).
- About 6 percent of the MSRT detections analyzed were likely caused by exploits. Of these, the majority had had security updates available for more than a year at the time of detection (classified as “Update Long Available”), with the remainder involving exploits for vulnerabilities for which security updates had been released less than a year before detection (classified as “Update Available”).
- File infectors, or viruses, accounted for 4 percent of detections.
- The password brute force and Office macro behaviors were each identified in just one of the families examined in this exercise, and accounted for 2 percent and 0.3 percent of the total, respectively.

Insights

The taxonomy introduced on page 5, codenamed “Broad Street,” organizes the categories used in this exercise according to propagation behavior, as shown in Figure 5.

Figure 5. The project Broad Street taxonomy, version 2.6



Vulnerability Subprocess



User Interaction

The first distinction shown in Figure 5 is between threats that require user interaction to compromise a computer and threats that do not. Threats that require user interaction can be further subdivided according to whether they require deception, and whether they require the user to make an explicit decision to install software. (An example of a mechanism that requires user interaction but not deception would be an opt-in botnet, such as [Java/Loic](#); see page 18 for more information.)

A typical example of a user interaction that isn't considered an installation decision would be a user following a hyperlink on a webpage or in an email message that leads to a page that attempts to use browser vulnerabilities to install malware. (See "Drive-By Download Sites" on page 89 for more information.)

Feature Abuse

Among threats that don't require user interaction, another fundamental distinction exists between threats that exploit vulnerabilities in software and threats that don't. The latter group includes file infecting viruses and threats that misuse legitimate features or functionality for malicious purposes.

Detections of threats that abuse features—including AutoRun threats, malicious scripts and macros, viruses, and password cracking—are increasing; the project Broad Street analysis attributes almost two-thirds of MSRT detections in 1H11 to a variety of feature abuses. This increase may be caused in part by an increase in the detection of threats that take advantage of the AutoRun feature in Windows. These threats spread by creating or modifying the autorun.inf file on mounted volumes in an effort to cause the computer to execute a malicious program whenever the volume is connected. Some of these threat families display an extra "Open folder to view files" entry in the AutoPlay dialog that appears by default in some versions of Windows when a network or removable volume is connected. Selecting this option would install the malware.

Microsoft introduced a change in the way the AutoRun feature works in Windows 7 and Windows Server® 2008 R2 in an effort to help protect users from such threats. In these versions of Windows, the AutoRun task is disabled for all volumes except optical drives such as CD-ROM and DVD-ROM drives, which have historically not been used to transmit AutoRun malware. In November 2009, Microsoft published a set of updates to the Microsoft Download Center that backported this change to Windows XP, Windows Server 2003, Windows Vista®, and

Windows Server 2008. As a result of data obtained through this exercise, these updates have been published as important updates through the Windows Update and Microsoft Update services since February 2011, and have been installed by more than 500 million computers since then.

The publication of these updates on Windows Update has had a significant effect on the ability of malware to use AutoRun to replicate. Between January and May of 2011, the MSRT reported decreases in detections of AutoRun-abusing families of between 62 and 82 percent on supported versions of Windows XP and Windows Vista. For more information, see the entry “[Autorun-abusing malware \(Where are they now?\)](#)” (June 14, 2011) in the Microsoft Malware Protection Center (MMPC) blog at blogs.technet.com/mmpc.

Exploit Age

When compared to the other categories of threats identified for the project Broad Street analysis, exploits are relatively rare, and exploits that target recently disclosed vulnerabilities are rarer still. Of the attacks attributed to exploits in the 1H11 MSRT data, less than half of them targeted vulnerabilities disclosed within the previous year, and none targeted vulnerabilities that were zero-day during the first half of 2011. (Because Microsoft usually releases security updates and the MSRT at the same time, the analysis considers a vulnerability zero-day for the entire month that an update is released. For example, if a malware family only uses a particular exploit in January, and Microsoft releases an update to fix the vulnerability in January, all February cleans of that family are counted as zero-day. This choice was made to avoid under-counting zero-days.)

Zero-Day Exploits: A Supplemental Analysis

However, if one considers exploits that are not associated with families detected by the MSRT, a small number of vulnerabilities did have zero-day exploits in 1H11. To assess the impact of these zero-day exploits compared to exploits of vulnerabilities for which security updates were available, the researchers conducted a supplemental analysis that used data from all Microsoft security products. (See “Appendix B: Data Sources” on page 122 for more information about the products and services that provided data for this report.)

The MMPC tracks vulnerability exploitation attempts using more than 3,000 signatures. Although some generic signatures may detect a zero-day exploit before the vulnerability has been disclosed, in most cases a signature update is required to detect or to single out one vulnerability exploit from another. Given these

constraints, some small-scale, targeted attacks using zero-day exploits may escape detection briefly, and such attacks would not be reflected in the data presented here. In general, though, when attacks involving an undisclosed vulnerability occur in significant volume, they are noticed quickly; security vendors respond by providing detection signatures and protection, and the affected software vendor publishes security updates to address the vulnerability.

In this supplemental analysis, zero-day exploitation accounted for about 0.12 percent of all exploit activity in 1H11, reaching a peak of 0.37 percent in June. Two vulnerabilities accounted for the bulk of zero-day exploit activity: [CVE-2011-0611](#), disclosed in April 2011, and [CVE-2011-2110](#), disclosed in June 2011. Both vulnerabilities affect Adobe Flash Player. (See “Adobe Flash Player Exploits” on page 47 for more information about these two exploits.)

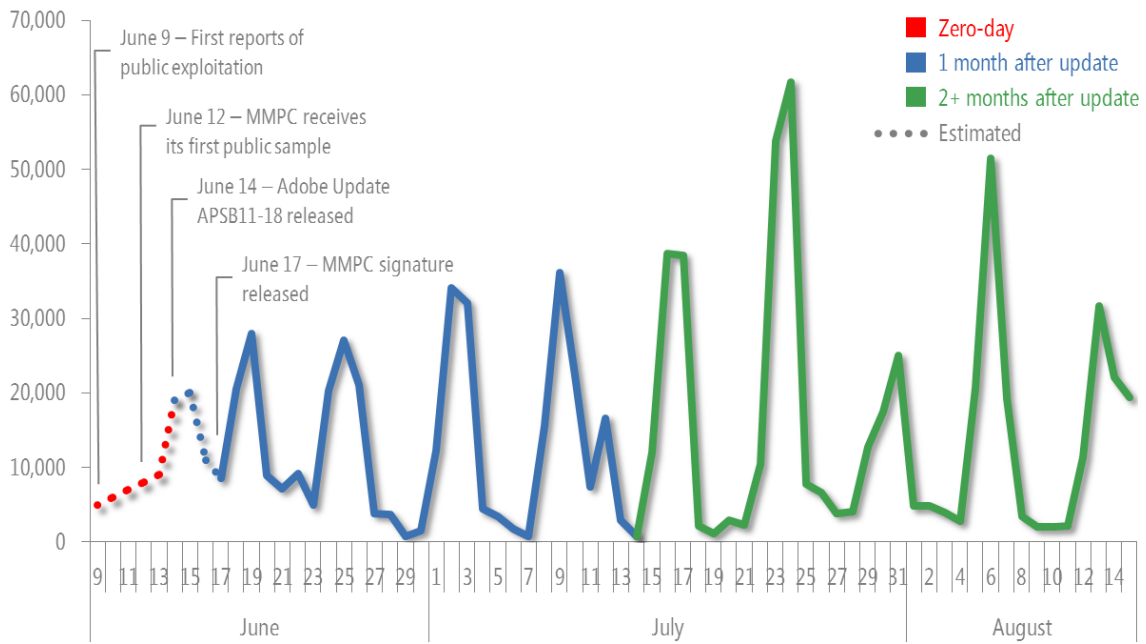
In the case of CVE-2011-0611, Adobe Systems released [Security Bulletin APSB11-07](#) for Adobe Flash Player on April 15, 2011, less than a week after the first reports of public exploitation. [Security Bulletin APSB11-08](#) for Adobe Reader and Adobe Acrobat was released the following week, on April 21, to address exploits involving malicious Flash files embedded in PDF documents. (Exploits using the PDF vector were only detected in a handful of samples before April 21, and the first real surge of activity using PDFs did not occur until May 13, a few weeks after the update had been released.)

Figure 6. Detections of exploits targeting CVE-2011-0611, April–July, 2011



For CVE-2011-2110, Adobe released an update on June 14, 2011 in response to targeted attacks that were reported to have been occurring since around June 9. The MMPC received its first exploit sample on June 12, two days before the release of the update. Microsoft released a generic signature, Exploit:SWF/ShellCode.A (subsequently redesignated [Exploit:SWF/CVE-2011-2110.A](#)), on June 17 to detect and remove the exploit.

Figure 7. Detections of exploits targeting CVE-2011-2110, June–August, 2011



In total, an estimated 0.04 percent of the CVE-2011-0611 attacks and 8.9 percent of the CVE-2011-2110 attacks came before the applicable security updates were released.

Analysis Details

The Project Broad Street Taxonomy

The following analysis uses a new taxonomy that was designed to classify propagation vectors. To create the taxonomy, researchers examined the documented propagation methods used by each of the malware families studied in the analysis. Successful malware propagation reflects a failure of the defensive systems that are in place to prevent attacks; consequently, focusing on means of propagation can help security professionals hone their defenses.²

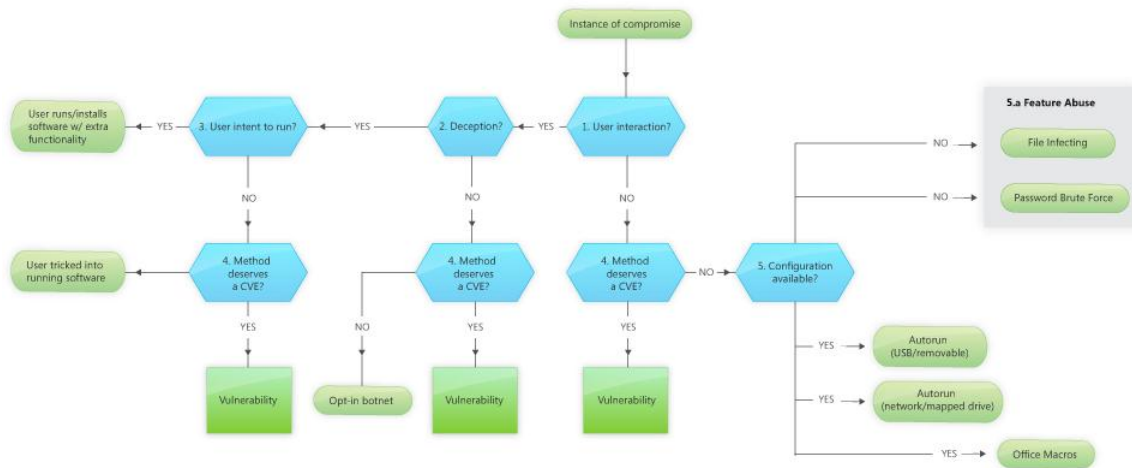
The taxonomy focuses on built-in malware propagation methods. The goal is to assess what percentage of malware succeeds by taking advantage of each vector to provide actionable data to the industry about what can be done to make it harder for malware to succeed using that vector in the future.

Using the Taxonomy

Figure 8 is a reprint of the project Broad Street taxonomy, first shown in Figure 5. The question boxes (diamonds) are numbered to make it easier to reference them in the text.

² This analysis intentionally focuses on *propagation* from computer to computer, rather than on malware *distribution*. File infection propagation from computer to computer occurs via shared or removable drives.

Figure 8. The project Broad Street taxonomy



User interaction required? (question 1) The first question the taxonomy poses is whether the user has to perform some action that results in a compromise. If the answer is Yes, the flow proceeds to question 2; if No, question 2 is skipped and the flow proceeds to question 4.

Deception? (2) The second question is one of deception. Deception often entails convincing someone that they will get some benefit from the action, or suffer some penalty if they don't do it, using any of a variety of social engineering techniques. Examples of deception might include a website telling people that they need to install a codec to watch a video, or an email message that claims to be from the tax authorities.

In some cases, users choose to install software that is designed to perform malicious actions. This classification includes scenarios involving **opt-in botnets**, in which the user chooses to give partial control of the computer to another party, who intends to use it to conduct activities such as denial-of-service (DoS) attacks. This category includes [Floder:Java/Loic](#), an open-source network attack tool designed to perform DoS attacks. Decentralized groups of protesters or vigilantes sometimes distribute software such as Java/Loic to users who wish to participate in DoS attacks on specific political or commercial targets.

If propagation requires deceiving the user, the flow proceeds to question 3. If it doesn't, question 3 is skipped and the flow proceeds to question 4.

User intent to run? (3) If user interaction is required, is the user aware that the action they are taking will involve running or installing software? If the answer is Yes, the flow terminates in an endpoint:

- **User runs/installs software with extra functionality.** The user runs the software, which performs malicious actions in addition to or instead of the software's desired function. A significant overlap exists between this kind of threat and the traditional definitions of "Trojan Horse" software. The analogy with the Trojan Horse from Greek mythology refers to the way many trojans gain access to victims' computers by masquerading as something innocuous: malicious executables represented as installers for legitimate security programs, for example, or disguised as documents for common desktop applications. In modern usage, however, most security vendors define *trojan* simply as a program that is unable to spread of its own accord. To avoid confusion, therefore, this analysis avoids use of the "trojan" or "Trojan Horse" labels.

If the answer is No, the flow proceeds to question 4.

Method deserves a CVE? (4) This question is the same for all three branches of the process flow, and determines whether or not a vulnerability is involved. Because the term "vulnerability" can be open to interpretation, the question asks whether the method used to install the software deserves to be documented in the Common Vulnerabilities and Exposures list (CVE), a standardized repository of vulnerability information maintained at cve.mitre.org. ("Deserves" is used for situations in which the method meets the CVE criteria but has not yet been assigned a CVE number, as with a previously undisclosed vulnerability.)

If the answer is Yes, the flow continues in the vulnerability subprocess, which is documented on page 20.

If the answer is No and user interaction is required to install or run the software, the flow terminates in one of two endpoints, depending on whether deception is involved:

- **User tricked into running software.** This result indicates a "false badging," such as a malicious executable named "document.pdf.exe" with an icon similar or identical to the one used for PDF files in Adobe Reader. The user launches the executable, believing it to be an ordinary PDF file, and it installs malware or takes other malicious actions.
- **Opt-in botnet.** This result indicates that the user has voluntarily installed botnet software.

If the answer is No and user interaction is not required to install or run the software, the flow proceeds to question 5.

Configuration available? (5) Can the attack vector be eliminated through configuration changes, or does it involve intrinsic product features that cannot be disabled through configuration? Configuration options would include things like turning the firewall off, and using a registry change to disable the AutoRun feature.

If the answer is Yes—in other words, if the attack vector can be eliminated through configuration changes—the flow terminates in one of three endpoints:

- **AutoRun (USB/removable).** The threat takes advantage of the AutoRun feature in Windows to propagate on USB storage devices and other removable volumes, as described on page 13.
- **AutoRun (network/mapped drive).** The threat takes advantage of the AutoRun feature to propagate via network volumes mapped to drive letters.
- **Office Macros.** The threat propagates on new computers when users open Microsoft Office documents with malicious Visual Basic® for Applications (VBA) macros.

Feature abuse: (5a) If the answer is No—in other words, if the attack vector uses product features that cannot be turned off via a configuration option—it is considered feature abuse, and the flow terminates in one of three endpoints:

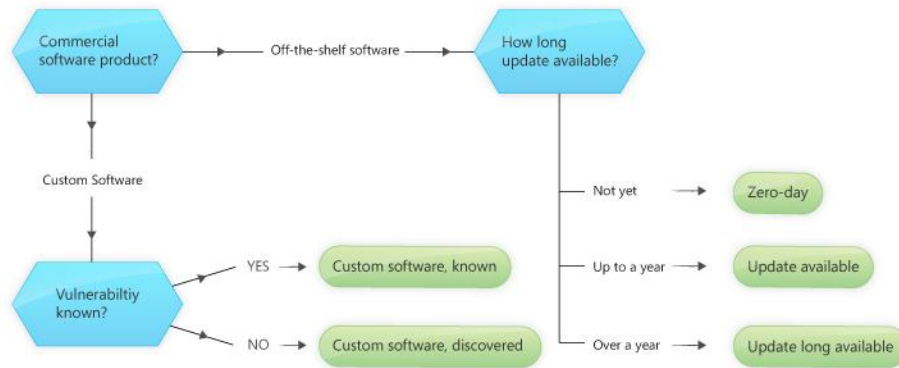
- **File infecting viruses.** The threat spreads by modifying files, often with .exe or .scr extensions, by rewriting or overwriting some code segments. To spread between computers, the virus writes to network drives or removable drives.
- **Password brute force.** The threat spreads by attempting brute force password attacks on available volumes to obtain Write or Execute permissions, as with the `net use` command.

A note on “other”: All taxonomies include either implied or explicit “other” or “unclassified” elements. For simplicity, these are not shown, but one could imagine classifying a threat as “other feature abuse,” “other configuration issue,” or “other ways a user is deceived.”

Vulnerability Subprocess

If the answer to question 4 is Yes—if the method used to install the software has or deserves a CVE entry—the attack is considered an exploit, and the process flow continues in a subprocess, shown in extended form in Figure 9.

Figure 9. The extended vulnerability subprocess of the project Broad Street taxonomy



The first question in the subprocess asks whether the vulnerability affects commercial software or custom software. Vulnerabilities are not unique to commercial software, and other exploit analyses have found that vulnerabilities in custom software, such as website code, account for a significant percentage of exploitation. Exploits of custom software are classified according to whether the vulnerability involved was known to the developers before the attack, or was discovered by the attacker.³

If the vulnerability affects commercial software, the flow terminates in one of three endpoints, according to the amount of time that has elapsed since the release of a security update addressing the vulnerability:

- **Zero-day.** The vendor had not released a security update to address the vulnerability at the time of the attack.
- **Update available.** The vendor released a security update that addressed the vulnerability less than a year before the attack.
- **Update long available.** The vendor released a security update that addressed the vulnerability more than a year before the attack.

Methodology Details

The project Broad Street analysis focuses on successful malware installs. Many other analyses are focused on attacks. Sometimes, attacks that are seen more often will seem more successful, but that may or may not be accurate.

³ The researchers would like to thank the Verizon RISK team for pointing out this extension to the approach.

One might object that only examining computers that are regularly updated would naturally tend to reduce exploit detections of all kinds. In fact, that is a key point: Regularly installing security updates is one of the most fundamental steps that IT departments and individual users can take to reduce their risk from malicious software. IT departments and individual users who are concerned about security—a group that is presumed to include most of those reading this report—are likely to regularly install security updates from Microsoft and other vendors, and to face less risk from older exploits as a result. The project Broad Street analysis, therefore, examines the residual risk faced by hundreds of millions of computers that are already being kept up to date.

Although the MSRT only detects a subset of the malware families recognized by Microsoft antimalware solutions, malware that propagates via exploits, such as “traditional” worms, do not seem to be underrepresented in this subset. Most of the prevalent malware families not detected by the MSRT are adware and other potentially unwanted software families, as shown in Figure 10.

Figure 10. The most commonly detected malware families not detected by the MSRT in 2Q11

	Family	Security Intelligence Report Category
1	Win32/Hotbar	Adware
2	JS/Pornpop	Adware
3	Win32/Autorun	Worms
4	Win32/OpenCandy	Adware
5	Win32/ShopperReports	Adware
6	Win32/Keygen	Miscellaneous Potentially Unwanted Software
7	Win32/ClickPotato	Adware
8	Win32/Zwangi	Miscellaneous Potentially Unwanted Software
9	Win32/Obfuscator	Miscellaneous Potentially Unwanted Software
10	Win32/OfferBox	Adware

Although malware can be distributed by vectors that are extrinsic to the malware, this analysis focuses on the documented ways in which specific forms of malware are installed.

Other classifications of malware

Other malware classification systems use some terms that this malware taxonomy does not, including:

- **Drive-by download.** This term refers to exploits that target vulnerabilities in web browsers, which can lead to computers becoming compromised if users simply browse to the malicious site. The project Broad Street taxonomy presented here does not use this term; it classifies all exploits according to whether a security update that addresses the vulnerability is available and how long ago it was released.
- **Exploit kit.** Exploit kits are collections of exploits that usually target web browsers and plugins in the form of packages that can be deployed on a web server. Project Broad Street sees exploit kits as collections of attacks that exploit vulnerabilities.
- **Pay per install.** This term is used to identify malware that is distributed by other malware as part of an affiliate scheme. This taxonomy is focused on the initial compromise, and does not take economic arrangements into consideration.
- **Bluetooth.** Some security software vendors highlight malware that uses Bluetooth wireless connections to propagate. Analysis of Bluetooth as a propagation mechanism is out of scope for this project, but it seems likely that use of this vector would be classified as either social engineering or exploits, or potentially a new part of the taxonomy.

Conclusion

The intent of this analysis is not to downplay the risks posed by zero-day vulnerabilities, or to encourage software vendors and others to “let their guard down” against them. Rather, it is to provide security professionals with information they can use to prioritize their concerns and respond effectively to threats. Like everyone else, IT departments face constraints of resources such as time, budget, and personnel when planning and performing their work. Having accurate, up-to-date information about the threat landscape is vitally important to security professionals who seek to effectively prioritize their defenses and keep their organizations safe.

Call to Action

- Security professionals, including antivirus/antimalware vendors, penetration testers, incident response analysts, and others can use the project Broad Street taxonomy to talk more clearly about how computers are compromised.
- Test and deploy security updates from all software vendors as quickly as possible. See the [Microsoft Security Update Guide](#), available from the Microsoft Download Center, for guidance and recommendations.
- Ensure that your development team is using the Security Development Lifecycle (SDL) (www.microsoft.com/sdl) or a similar software security assurance process. Using such a methodology can help reduce the number of vulnerabilities in software and help manage vulnerabilities that might be found after deployment.
- Build your defenses against social engineering.

Advice to IT Professionals on Social Engineering

IT professionals are accustomed to thinking about the technical aspects of security. However, as this report has shown, the human element—the techniques that attackers use to trick typical users into helping them—has become just as important for attackers as the technical element, if not more so. By implementing effective technical safeguards, programs, and processes designed to defend against social engineering, you can help your users avoid being taken advantage of by attackers. You can even enlist them as some of your most valuable assets in the fight against security threats.

Organizations

Your network provides the underlying infrastructure in which your applications are deployed. It is important to secure your network as a vital component of your defense-in-depth strategy.

Minimize and Monitor Your Attack Surface

- Limit the number of powerful user accounts in your organization and the level of access they have, because this will help limit the harm a successful social engineering attack can cause.
- Regularly audit your powerful user accounts. Provide them only to those who must have access, and to the specific resources to which they need access.
- Ensure these user accounts have strong authentication (strong passwords and/or two-factor authentication).
- Regularly audit attempts to access sensitive company information—both failed and successful attempts.

Create a Social Engineering Incident Response Plan

- Put in place systems to detect and investigate potential social engineering attacks.
- Create a virtual team to respond to attacks, and consider the following areas:
 - What was or is being attacked, and how.
 - Which resources are threatened or compromised.
 - How to shut down an ongoing attack with the least amount of disruption to the business.
 - How to recover from the attack.
 - How to implement protections against similar attacks.

Create a Plan For Addressing Social Engineering In Your Organization

- Determine which threats have the greatest potential:
 - Determine the resources attackers are most likely to pursue and those most critical to the business.
 - Analyze attacks that have occurred against your organization and those like it.
 - Determine where technology, policies, or company culture creates “soft spots” that are especially vulnerable to social engineering attacks.
- Determine how to address these vulnerable areas:
 - Determine where technology or processes can be altered to reduce or eliminate the threats.
 - Create policies that make it easy for people to perform secure actions without feeling rude.
 - Create awareness training for those vulnerable areas that are most critical, and where technology, process, and policy may not address

the problem sufficiently. Ensure that your guidance fits well within your organizational culture; it should be:

- **Realistic.** Guidance should enable typical people to accomplish their goals without inconveniencing them.
 - **Durable.** Guidance should remain true and relevant, and not be easy for an attacker to use against your people.
 - **Memorable.** Guidance should stick with people, and should be easy to recall when necessary.
 - **Proven Effective.** Guidance should be tested and shown to actually help prevent social engineering attacks.
 - **Concise and Consistent.** The amount of guidance you provide should be minimal, be stated simply, and be consistent within all the contexts in which you provide it.
- More details on how to create a process around social engineering prevention and response can be found in [“How to Protect Insiders from Social Engineering Threats”](#) on Microsoft TechNet.

Software

Many social engineering attacks involve tricking the user into opening a malicious file or browsing to a malicious website that takes advantage of a code vulnerability. As the data presented in this report shows, in many cases these attacks use vulnerabilities for which a security update has already been made available—sometimes quite a while ago. One of the most important things you can do to blunt social engineering attacks is to keep software as up-to-date as possible. The [Microsoft Security Update Guide, Second Edition](#), available from the Microsoft Download Center, provides guidance on how to deliver updates to your users in a timely and effective manner, in consideration of all of the other challenges in your IT environment.

People

Information security awareness and training are critical for any organization’s information security strategy and for supporting security operations.

In many scenarios, people are an organization's last line of defense against threats such as malicious code, disgruntled employees, and malicious third parties. It is therefore important to educate workers on what your organization considers appropriate security-conscious behavior, and on the security best practices they need to incorporate in their daily business activities.

Drive Awareness and Train Your Organization

- Use creative ways to help your people understand the threat that social engineering imposes, the skill with which attacks are carried out, their role in protecting the organization, and the advice that will enable them to resist these attacks.
- Provide a regular rhythm of updated information and refresher courses to keep employees aware of the risks involved in relaxing security.
- Keep the message fresh so people don't lose sight of its meaning and importance.

Encourage the Behavior You Want and Enforce Where Necessary

- Many social engineering attacks take advantage of the positive qualities of people and social norms. Find ways to encourage behavior that allows for questioning of why someone needs information or access, such that it becomes socially acceptable to push back or say "No."
- When enforcement is necessary, set policies to require realistic safe behavior. Ensure that users understand why such measures are necessary to protect the organization as well as the consequences of not following the policy.



Worldwide Threat Assessment



Vulnerabilities

Vulnerabilities are weaknesses in software that enable an attacker to compromise the integrity, availability, or confidentiality of that software or the data it processes. Some of the worst vulnerabilities allow attackers to exploit the compromised system by causing it to run arbitrary code without the user's knowledge.

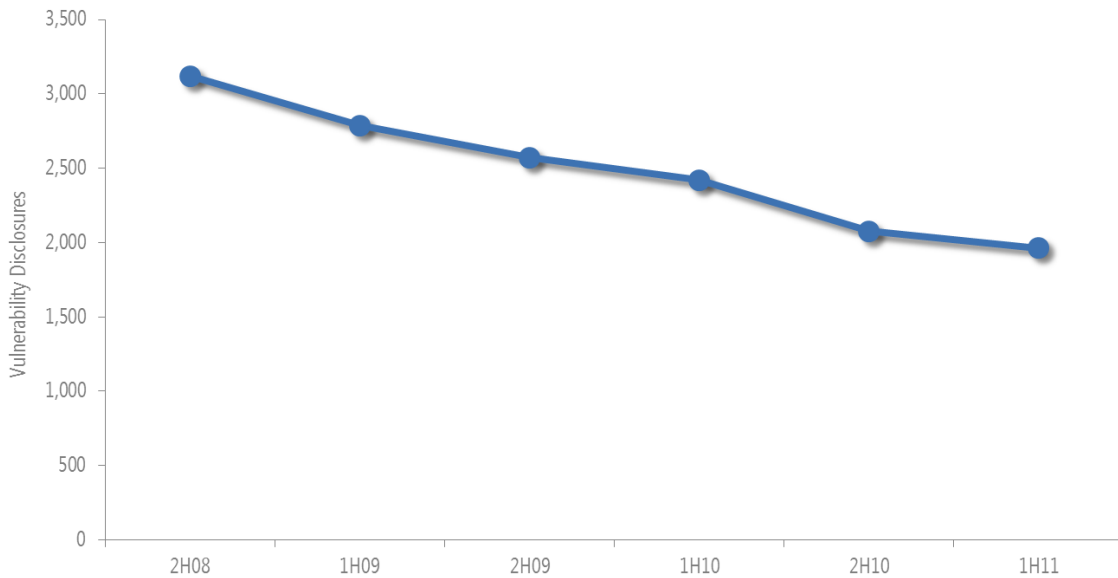
Industry-Wide Vulnerability Disclosures

A *disclosure*, as the term is used in the *Microsoft Security Intelligence Report*, is the revelation of a software vulnerability to the public at large. It does not refer to any type of private disclosure or disclosure to a limited number of people. Disclosures can come from a variety of sources, including the software vendor, security software vendors, independent security researchers, and even malware creators.

The information in this section is compiled from vulnerability disclosure data that is published in the National Vulnerability Database (<http://nvd.nist.gov>), the U.S. government repository of standards-based vulnerability management. It represents all disclosures that have a CVE (Common Vulnerabilities and Exposures) number.

Figure 11 illustrates the number of vulnerability disclosures across the software industry for each half-year period since 2H08. (See “About This Report” on page ix for an explanation of the reporting period nomenclature used in this report.)

Figure 11. Industry-wide vulnerability disclosures, 2H08–1H11

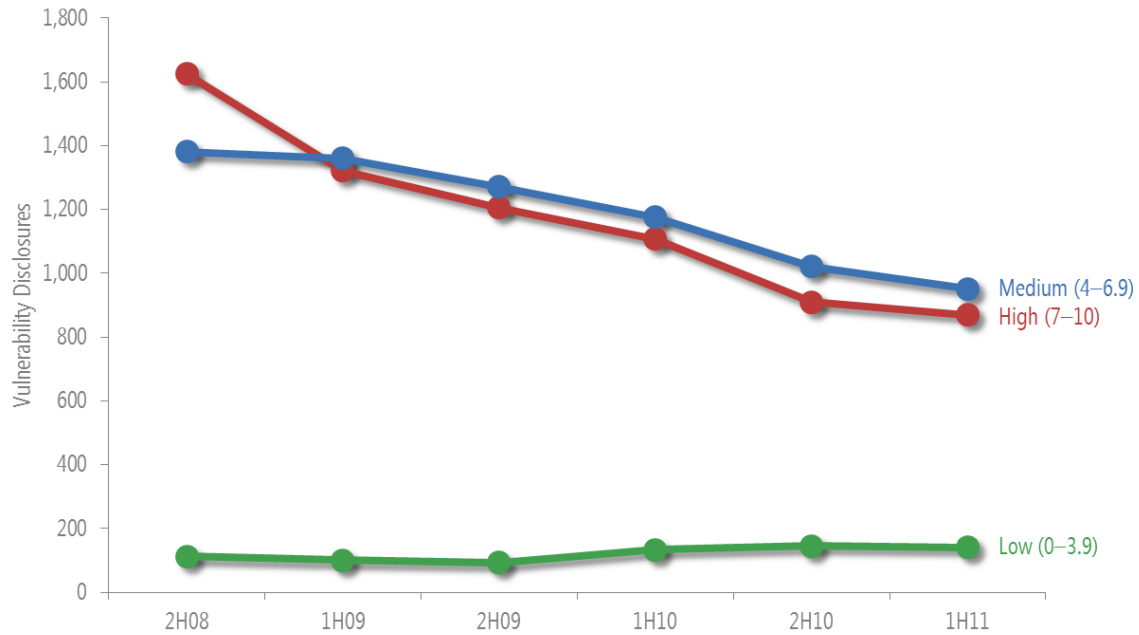


- Vulnerability disclosures across the industry in 1H11 were down 5.5 percent from 2H10, and down 37.1 percent from 2H08.
- This decline continues an overall trend of moderate declines since 2006. This trend is likely because of better development practices and quality control throughout the industry, which results in more secure software and fewer vulnerabilities. (See [Protecting Your Software](#) in the “Managing Risk” section of the *Microsoft Security Intelligence Report* website for additional details and guidance about secure development practices.)

Vulnerability Severity

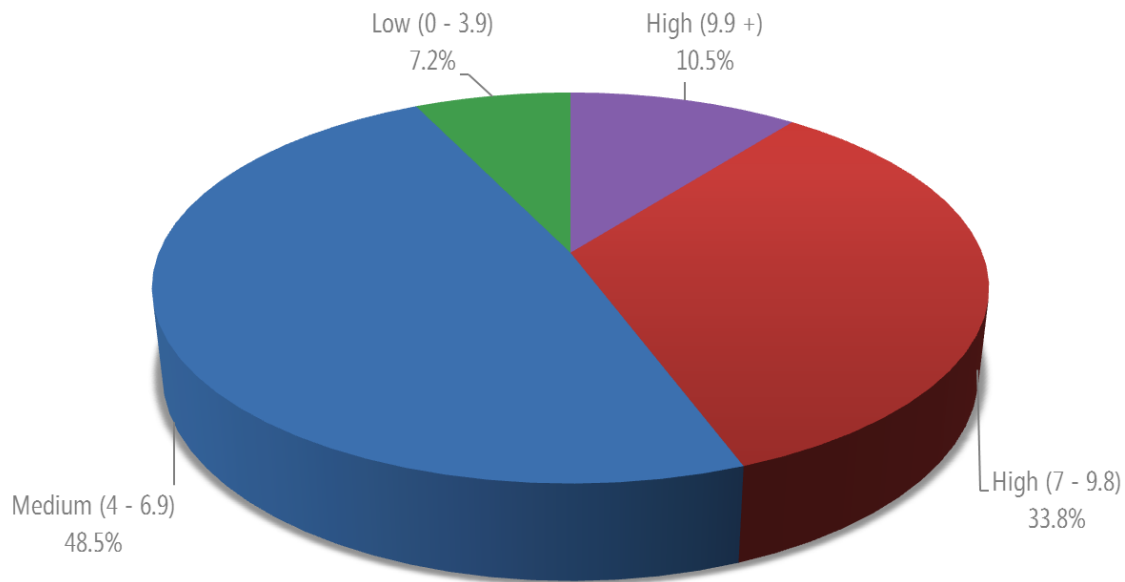
The Common Vulnerability Scoring System (CVSS) is a standardized, platform-independent scoring system for rating IT vulnerabilities. The CVSS assigns a numeric value between 0 and 10 to vulnerabilities according to severity, with higher scores representing greater severity. (See [Vulnerability Severity](#) at the *Microsoft Security Intelligence Report* website for more information.)

Figure 12. Industry-wide vulnerability disclosures by severity, 2H08–1H11



- The overall vulnerability severity trend has been a positive one. Medium and High severity vulnerabilities disclosed in 1H11 were down 6.8 percent and 4.4 percent from 2H10, respectively.
- Even as fewer vulnerabilities are being disclosed overall, the number of Low severity vulnerabilities being disclosed has increased slightly. Low severity vulnerabilities accounted for 7.2 percent of all vulnerabilities disclosed in 1H11.
- Mitigating the most severe vulnerabilities first is a security best practice. High severity vulnerabilities that scored 9.9 or greater represent 10.5 percent of all vulnerabilities disclosed in 1H11, as Figure 13 illustrates.

Figure 13. Industry-wide vulnerability disclosures in 1H11, by severity

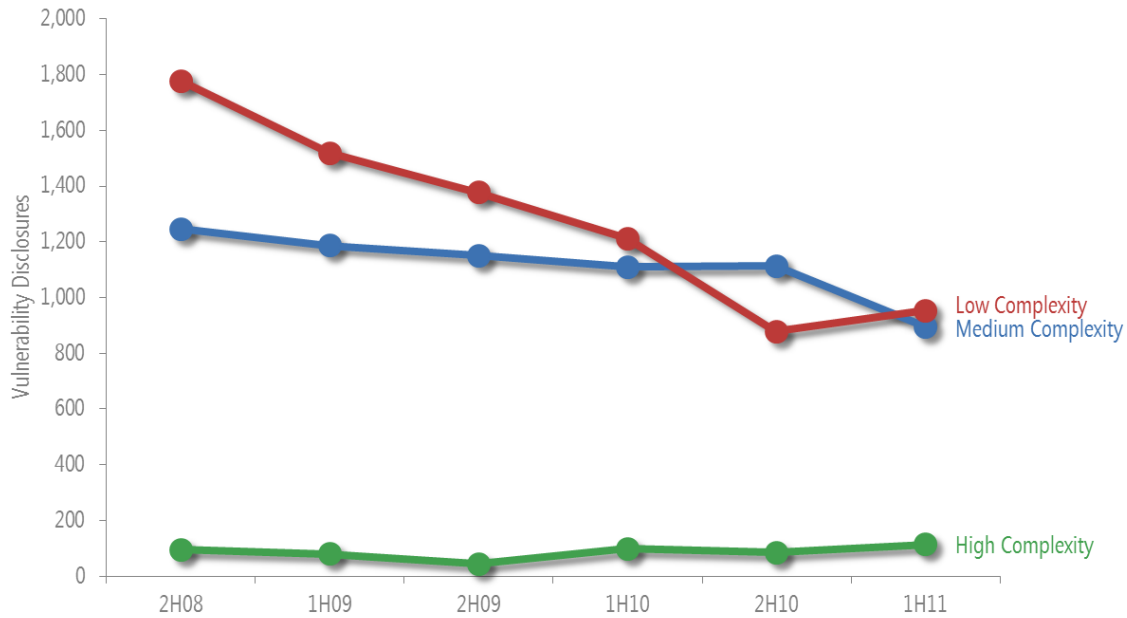


Vulnerability Complexity

Some vulnerabilities are easier to exploit than others, and vulnerability complexity is an important factor to consider in determining the magnitude of the threat that a vulnerability poses. A High severity vulnerability that can only be exploited under very specific and rare circumstances might require less immediate attention than a lower severity vulnerability that can be exploited more easily.

The CVSS gives each vulnerability a complexity ranking of Low, Medium, or High. (See [Vulnerability Complexity](#) at the *Microsoft Security Intelligence Report* website for more information about the CVSS complexity ranking system.) Figure 14 shows complexity trends for vulnerabilities disclosed since July 2006. Note that Low complexity indicates greater danger, just as High severity indicates greater danger in Figure 12.

Figure 14. Industry-wide vulnerability disclosures by access complexity, 2H08–1H11

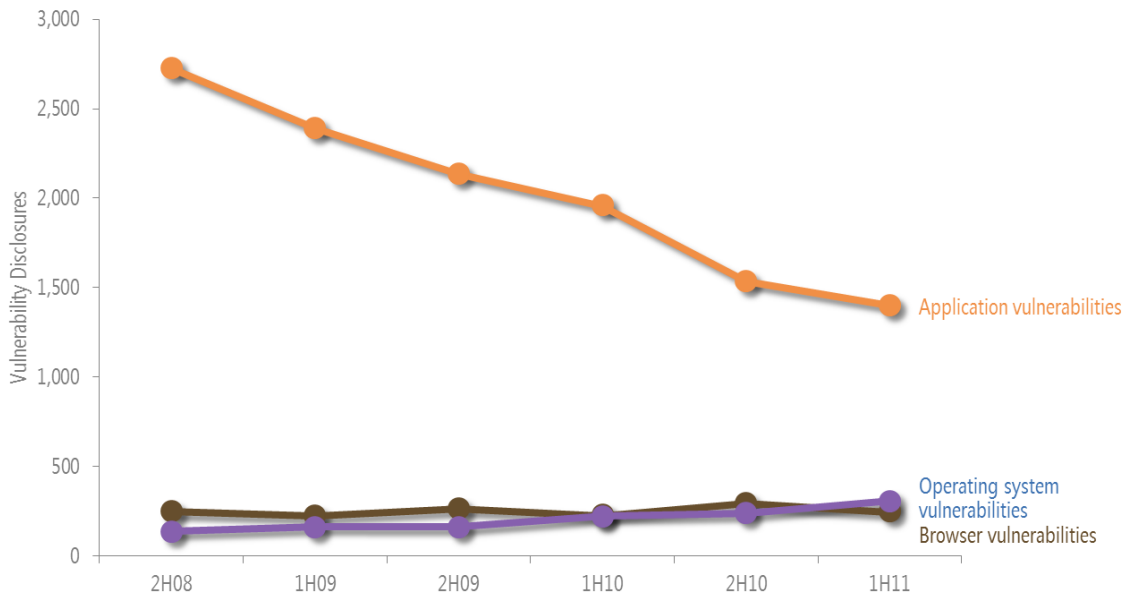


- As with vulnerability severity, the trend here is a positive one, with Low complexity vulnerabilities—the easiest ones to exploit—down 41.2 percent from the prior 12-month period.
- High complexity vulnerability disclosures, meanwhile, have increased slightly. They accounted for 4.9 percent of all vulnerabilities disclosed between July 2010 and June 2011, up from 2.8 percent in the prior 12-month period.

Operating System, Browser, and Application Vulnerabilities

Figure 15 shows industry-wide vulnerabilities for operating systems, browsers, and applications since July 2006. (See [Operating System, Browser, and Application Vulnerabilities](#) at the *Microsoft Security Intelligence Report* website for an explanation of how operating system, browser, and application vulnerabilities are distinguished.)

Figure 15. Industry-wide operating system, browser, and application vulnerabilities, 2H08–1H11

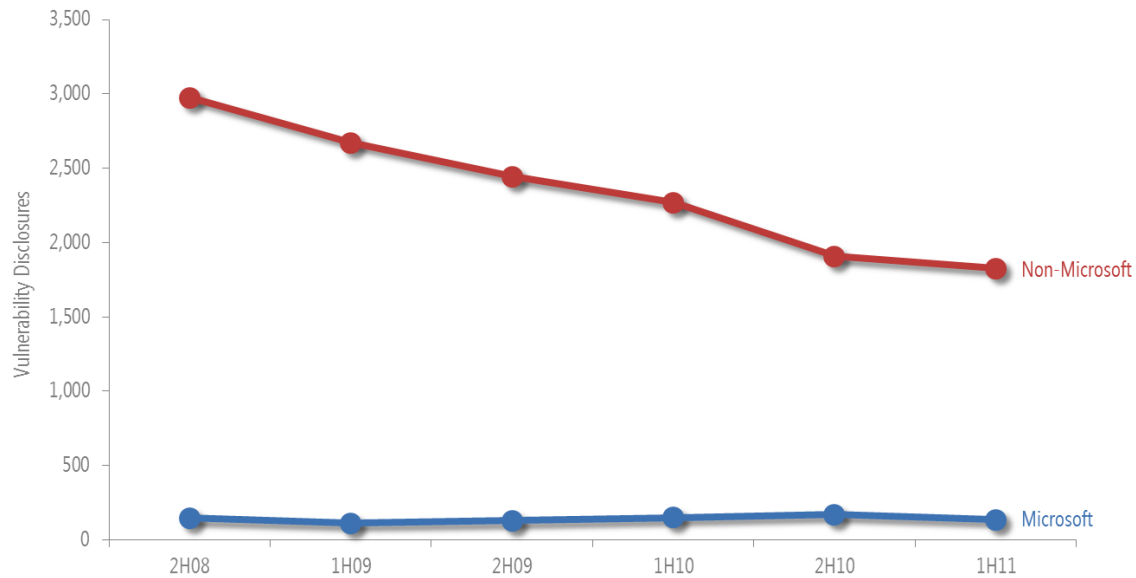


- As Figure 15 shows, most of the industry-wide decline in vulnerability disclosures over the past several years has been caused by a decrease in application vulnerabilities, which were down 8.8 percent from 1H11.
- Despite this decline, application vulnerabilities still accounted for 71.5 percent of all vulnerabilities disclosed in 1H11.
- Operating system and browser vulnerability disclosures have been mostly stable for several years, accounting for 12.7 percent and 15.7 percent of all vulnerabilities disclosed in 1H11, respectively.

Microsoft Vulnerability Disclosures

Figure 16 charts vulnerability disclosures for Microsoft and non-Microsoft products since 2H08.

Figure 16. Vulnerability disclosures for Microsoft and non-Microsoft products, 2H08–1H11



- Vulnerabilities in Microsoft products accounted for 6.9 percent of all vulnerabilities disclosed in 1H11, down from 8.2 percent in 2H10.
- Vulnerability disclosures for Microsoft products have generally remained stable over the past several periods, though the percentage of all disclosures industry-wide that affect Microsoft products has increased slightly, primarily because of the overall decline in vulnerability disclosures across the industry.

Guidance: Developing Secure Software

The Security Development Lifecycle (www.microsoft.com/sdl) is a software development methodology that embeds security and privacy throughout all phases of the development process with the goal of protecting software users. Using such a methodology can help reduce vulnerabilities in the software and help manage vulnerabilities that might be found after deployment. (For more in-depth information about the SDL and other techniques developers can use to secure their software, see [Protecting Your Software](#) in the “Managing Risk” section of the *Microsoft Security Intelligence Report* website.)

Exploits

An *exploit* is malicious code that takes advantage of software vulnerabilities to infect, disrupt, or take control of a computer without the user's consent and usually without the user's knowledge. Exploits target vulnerabilities in operating systems, web browsers, applications, or software components that are installed on the computer. In some scenarios, targeted components are add-ons that are pre-installed by the computer manufacturer before the computer is sold. A user may not even use the vulnerable add-on or be aware that it is installed. Some software has no facility for updating itself, so even if the software vendor publishes an update that fixes the vulnerability, the user may not know that the update is available or how to obtain it, and therefore remains vulnerable to attack.

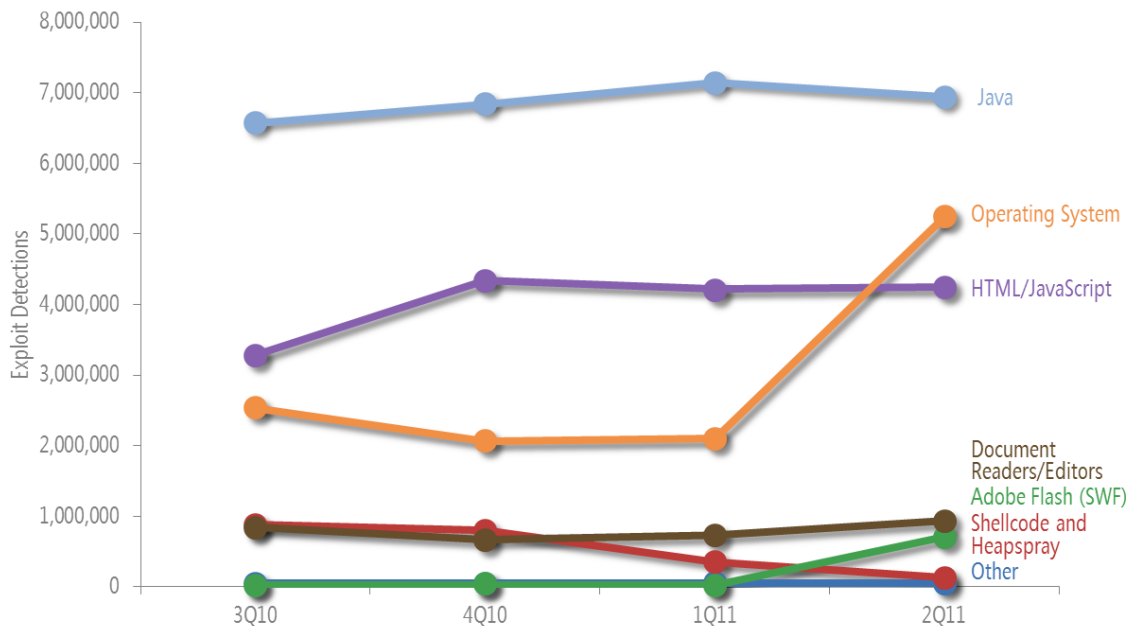
Software vulnerabilities are enumerated and documented in the Common Vulnerabilities and Exposures list (CVE) (<http://cve.mitre.org>), a standardized repository of vulnerability information. Here and throughout this report, exploits are labeled with the CVE identifier that pertains to the affected vulnerability, if applicable. In addition, exploits that affect vulnerabilities in Microsoft software are labeled with the Microsoft Security Bulletin number that pertains to the vulnerability, if applicable.⁴

Note that most of the charts in the “Exploits” section, with the exception of Figure 25 on page 47, show individual attack counts rather than unique computers affected.

Figure 17 shows the prevalence of different types of exploits for each quarter between 3Q10 and 2Q11.

⁴ See www.microsoft.com/technet/security/Current.aspx to search and read Microsoft Security Bulletins.

Figure 17. Exploits detected and blocked by Microsoft antimalware products, 3Q10–2Q11, by targeted platform or technology

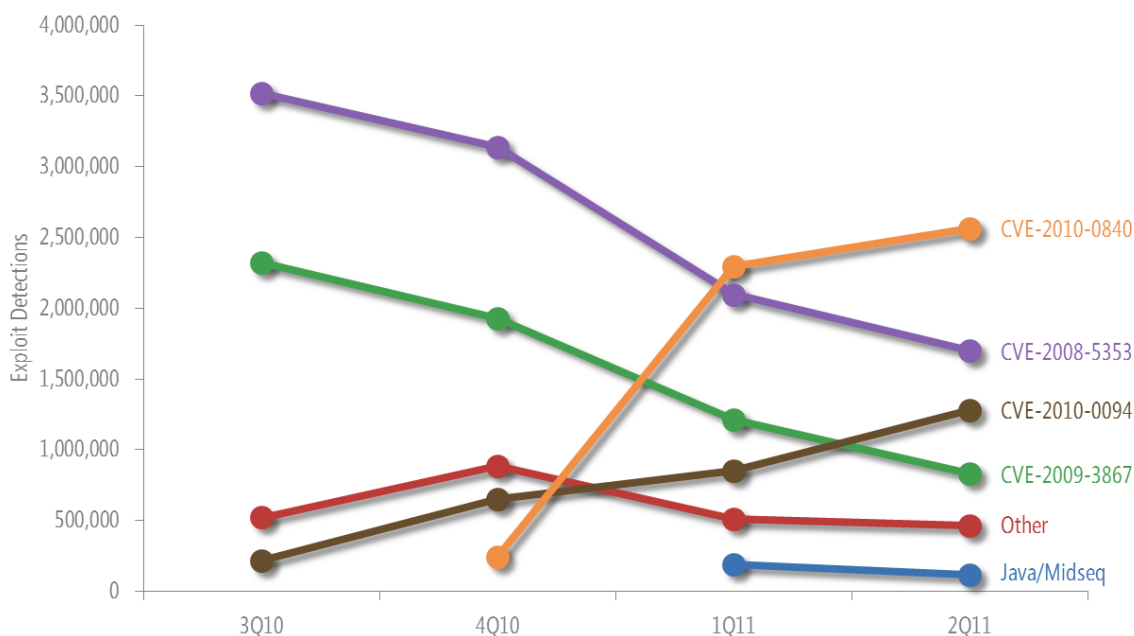


- The most commonly observed type of exploits in 1H11 were those targeting vulnerabilities in the Oracle (formerly Sun) Java Runtime Environment (JRE), Java Virtual Machine (JVM), and Java SE in the Java Development Kit (JDK). Java exploits were responsible for between one-third and one-half of all exploits observed in each of the four most recent quarters.
- Detections of operating system exploits increased dramatically in 2Q11 because of increased exploitation of vulnerability [CVE-2010-2568](#). (See “Operating System Exploits” on page 45 for more information.)
- Detections of exploits targeting Adobe Flash, although uncommon in comparison to some other types of exploits, increased in 2Q11 to more than 40 times the volume seen in 1Q11 because of exploitation of a pair of newly-discovered vulnerabilities. (See “Adobe Flash Player Exploits” on page 47 for more information about these vulnerabilities.)
- The web is the most common vector by which exploits are delivered. Java and HTML/JavaScript exploits are usually delivered through the web, as are large percentages of other types of exploits. Malicious documents that contain exploits are sometimes delivered over the web, but are also often sent directly to prospective victims as files attached to email messages. Similarly, Flash exploits are often delivered over the web, but are sometimes embedded in malicious documents sent through email.

Java Exploits

Figure 18 shows the prevalence of different Java exploits by quarter.

Figure 18. Java exploits detected and blocked by Microsoft antimalware products, 3Q10–2Q11



- As in previous periods, many of the more commonly exploited Java vulnerabilities are several years old, as are the security updates that have been released to address them.
- The most commonly exploited Java vulnerability in 1Q11 and 2Q11 was [CVE-2010-0840](#), a Java Runtime Environment (JRE) vulnerability first disclosed in March 2010 and addressed with an [Oracle security update](#) the same month. Exploitation of the vulnerability was first detected at a low level in 4Q10 before increasing tenfold in 1Q11.
- [CVE-2008-5353](#), the second most commonly exploited Java vulnerability in 1Q11 and 2Q11, was first disclosed in December 2008. This vulnerability affects JVM version 5 up to and including update 22, and JVM version 6 up to and including update 10. It allows an unsigned Java applet to gain elevated privileges and potentially have unrestricted access to a host system, outside its “sandbox” environment. Sun Microsystems released a security update that addressed the vulnerability on December 3, 2008.
- [CVE-2010-0094](#), the fourth most commonly exploited Java vulnerability in 1Q11 and the third in 2Q11, was first disclosed in December 2009. The

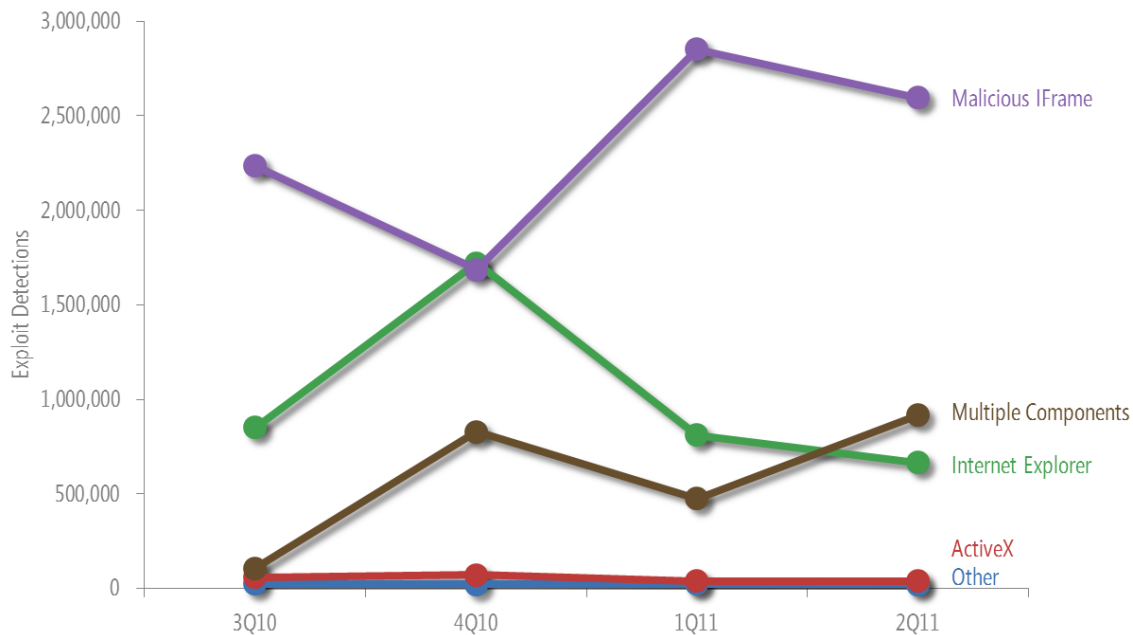
vulnerability affects JRE versions up to and including update 18 of version 6. It allows an unsigned Java applet to gain elevated privileges and potentially have unrestricted access to a host system, outside its sandbox environment. Oracle released a [security update](#) that addressed the vulnerability in March 2010.

- [CVE-2009-3867](#), the third most commonly exploited Java vulnerability in 1Q11 and the fourth in 2Q11, was first disclosed in November 2009. The vulnerability affects JVM version 5 up to and including update 21, and JVM version 6 up to and including update 16. When an applet that exploits the vulnerability is loaded by a computer with a vulnerable version of Java, security checks may be bypassed, allowing the execution of arbitrary code. Sun Microsystems released a security update that addressed the vulnerability on November 3, 2009.

HTML and JavaScript Exploits

Figure 19 shows the prevalence of different types of HTML and JavaScript exploits during each of the four most recent quarters.

Figure 19. Types of HTML and JavaScript exploits detected and blocked by Microsoft antimalware products, 3Q10–2Q11



- Most of the exploits observed involved malicious HTML inline frames (IFrames). These exploits are typically generic detections of inline frames that

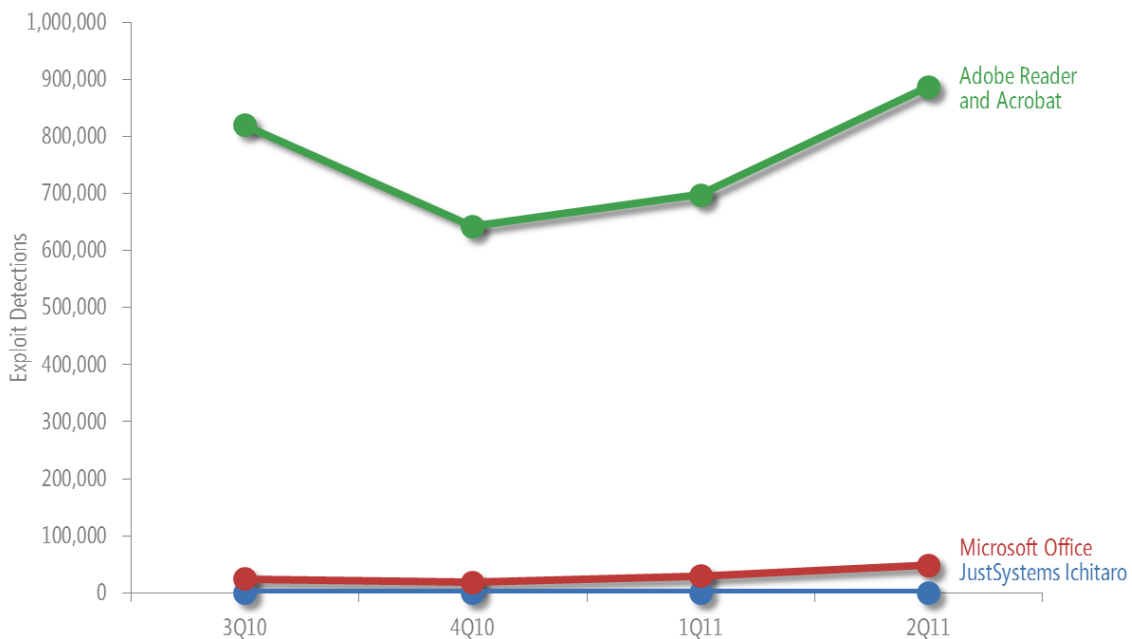
are embedded in web pages and link to other pages that host malicious web content. These malicious pages use a variety of techniques to exploit vulnerabilities in browsers and plugins, with the only commonality being that the exploit can be delivered through an inline frame. The exact exploit delivered and detected by one of these signatures may be changed frequently.

- After peaking in 4Q10, exploits that target Windows Internet Explorer® returned to a more typical level in 1Q11 and stayed at the lower level in 2Q11. The 4Q10 peak largely involved exploits targeting [CVE-2010-0806](#), a vulnerability in versions 6 and 7 of Internet Explorer. Microsoft released security bulletin [MS10-018](#) in March 2010 to address the vulnerability.

Document Parser Exploits

Document parser exploits are those that target vulnerabilities in the way a document editing or viewing application processes, or parses, a particular file format. Figure 20 shows the prevalence of different types of document parser exploits during each of the four most recent quarters.

Figure 20. Types of document parser exploits detected and blocked by Microsoft antimalware products, 3Q10–2Q11



- Exploits that affect Adobe Acrobat and Adobe Reader accounted for most document format exploits detected throughout the last four quarters. Most of

these exploits were detected as variants of the generic exploit family [Win32/Pdfjsc](#).

- Exploits that affect Microsoft Office and Ichitaro, a Japanese-language word processing application published by JustSystems, accounted for a small percentage of exploits detected during the period. (See the following section for more information about Office exploits.)

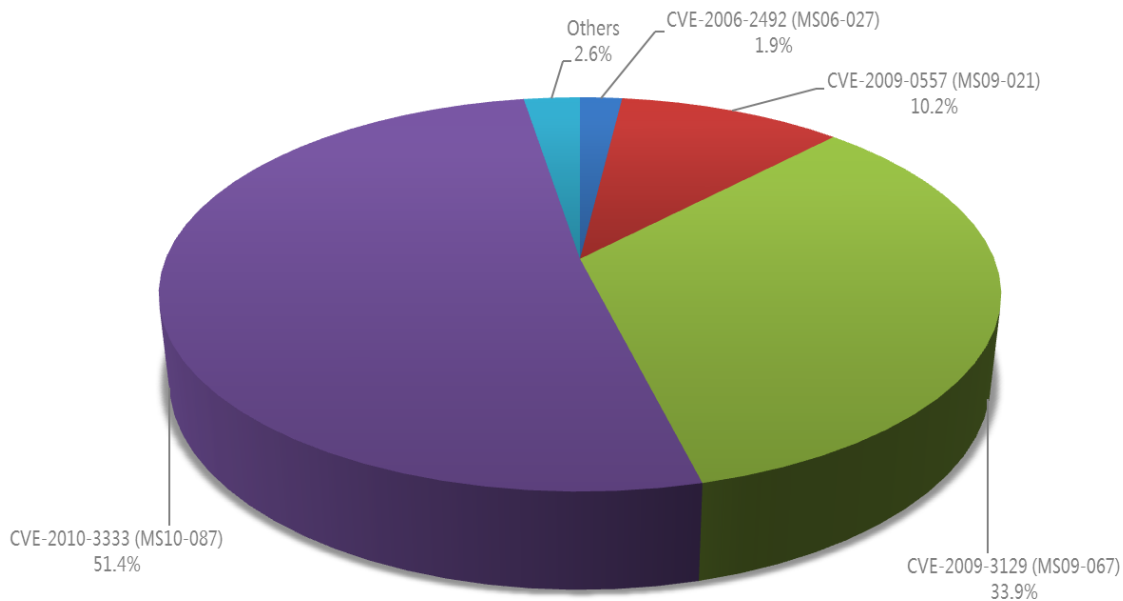
Microsoft Office File Format Exploits

To assess the use of Microsoft Office system file formats as an attack vector, Microsoft analyzed a sample set of several hundred files that were used for successful attacks in 1H11. The data set was taken from submissions of malicious code sent to Microsoft from customers worldwide.

Figure 21. Vulnerabilities exploited in Microsoft Office file formats in 1H11

CVE	Vulnerability	Bulletin	Release Date
CVE-2006-2492	Word Malformed Object Pointer Vulnerability	MS06-027	June 2006
CVE-2006-0022	PowerPoint® Remote Execution Via a Malformed Record Vulnerability	MS06-028	June 2006
CVE-2006-6456	Word Remote Execution Vulnerability	MS07-014	February 2007
CVE-2007-0671	Excel® Malformed Record Vulnerability	MS07-015	February 2007
CVE-2008-0081	Macro Validation Vulnerability	MS08-014	March 2008
CVE-2009-0238	Excel Memory Corruption Vulnerability	MS09-009	April 2009
CVE-2009-0557	Excel Object Record Corruption Vulnerability	MS09-021	June 2009
CVE-2009-3129	Excel Record Memory Corruption	MS09-067	November 2009
CVE-2010-3333	Word RTF File Parsing Stack Buffer Overflow Vulnerability	MS10-087	November 2010
CVE-2011-0979	Excel Parsing Vulnerability allows Remote Code Execution	MS11-021	April 2011

Figure 22. Microsoft Office file format exploits encountered in 1H11, by percentage

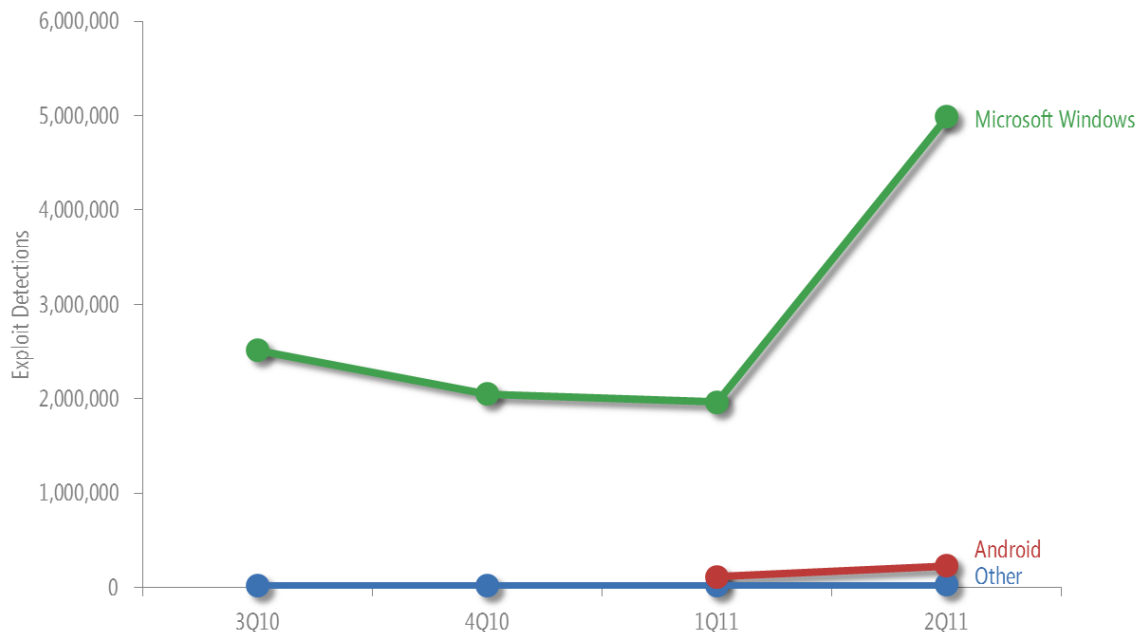


- In total, exploits for 10 vulnerabilities were identified in the sample set, as shown in Figure 21. All 10 of these vulnerabilities had security updates available at the time of the attack. The affected users were exposed because they had not applied the updates.
- More than half of the exploits involved [CVE-2010-3333](#), a vulnerability in the Rich Text Format (RTF) parser in versions of Microsoft Word that was addressed by [Security Bulletin MS10-087](#) in November 2010.
- Most of the other exploits in the sample involved [CVE-2009-3129](#), a vulnerability in Microsoft Excel that was addressed by [Security Bulletin MS09-067](#) in November 2009. Installing these two security updates would have protected users from 85.3 percent of the attacks in the sample set.
- None of the encountered exploits are effective in Office 2010 applications running in their default configurations on Windows Vista or Windows 7. All of the exploits take advantage of techniques that are blocked by address space layout randomization (ASLR) or Data Execution Prevention (DEP), two security-related technologies included in recent versions of Windows. ASLR and DEP are both enabled by default in Office 2010. DEP is available in Windows XP SP3, Windows Vista, and Windows 7; ASLR is available in Windows Vista and Windows 7. (See Appendix D on page 131 for a table of Office versions and their level of exposure to the exploits encountered in 1H11.)

Operating System Exploits

Although most operating system exploits detected by Microsoft security products are designed to affect the platforms on which the security products run, computer users sometimes download malicious or infected files that affect other operating systems. Figure 23 shows the prevalence of different operating system exploits detected and removed by Microsoft security products during each of the past four quarters.

Figure 23. Types of operating system exploits detected and blocked by Microsoft antimalware products, 3Q10–2Q11

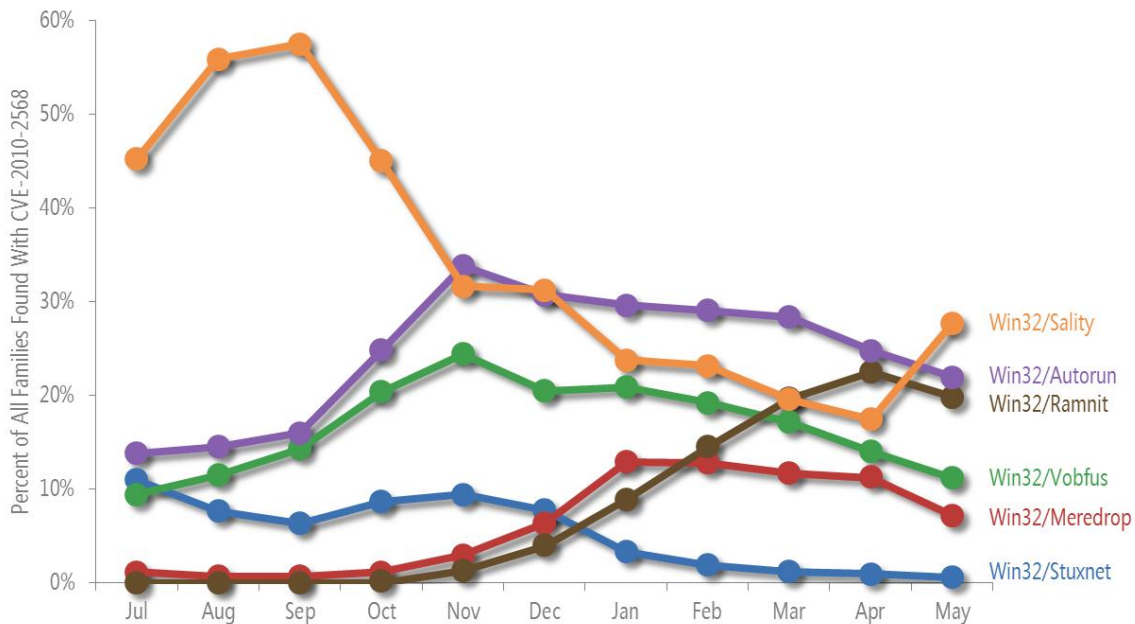


- Detection totals for Windows are inflated by detections of [CVE-2010-2568](#), which is often detected repeatedly on the same computer because of the mechanism it uses to spread. (See page 47 for more information.)
- Exploits that target [CVE-2010-2568](#), a vulnerability in Windows Shell, increased significantly in 2Q11, and were responsible for the entire 2Q11 increase in Windows exploits shown in Figure 23. Microsoft issued [Security Bulletin MS10-046](#) in August 2010 to address the vulnerability.

An attacker exploits [CVE-2010-2568](#) by creating a malformed shortcut file that forces a vulnerable computer to load a malicious file when the shortcut icon is displayed in Windows Explorer. The vulnerability was first discovered being used by the malware family [Win32/Stuxnet](#) in mid-2010, and it has since been exploited by a number of pre-existing families, many of which had

been designed to spread using malicious shortcut files or by abusing the AutoRun feature in Windows. The CVE-2010-2568 attack mechanism is similar to the techniques already in use by these families, which may explain why their authors chose to incorporate the exploit into new variants.

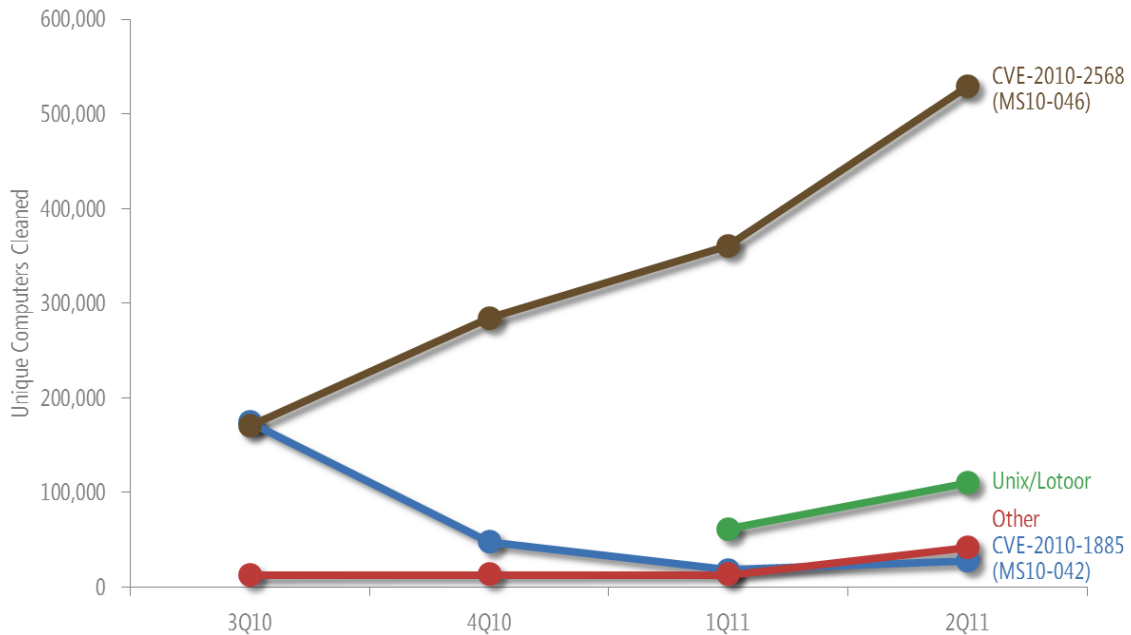
Figure 24. Families commonly found with CVE-2010-2568, July 2010–June 2011



- Exploits that affect the Android mobile operating system published by Google and the Open Handset Alliance have been detected in significant volume beginning in 1H11. Microsoft security products detect these threats when Android users download infected or malicious programs to their computers before transferring the software to their devices. The increase in Android-based threats has been driven primarily by the exploit family [Unix/Lotoor](#), the second most commonly detected operating system exploit in 1Q11 and 2Q11. Lotoor is used to attack vulnerable devices by the trojan family [AndroidOS/DroidDream](#), which often masquerades as a legitimate Android application, and can allow a remote attacker to gain access to the mobile device. Google published a [security update](#) in March 2011 that addressed the vulnerability.

For another perspective on these exploits and others, Figure 25 shows trends for the individual exploits most commonly detected and blocked or removed in 1H11.

Figure 25. Individual operating system exploits detected and blocked by Microsoft antimalware products, 3Q10–2Q11, by number of unique computers exposed to the exploit

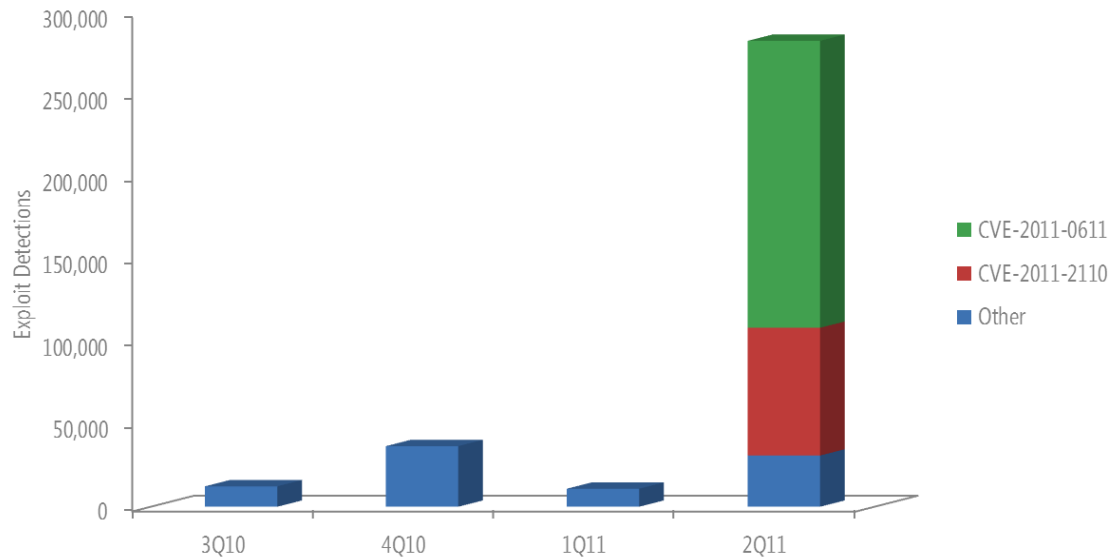


- Unlike the other charts in this section, Figure 25 shows the number of unique computers affected by each exploit, rather than the number of individual attacks detected. [CVE-2010-2568](#) exploits have a tendency to be reported by the same computer many times (eight on average, although some computers report thousands of attack attempts), because of the way the exploit technique works, which could give a misleading impression of the exploit’s impact.
- [CVE-2010-1885](#), a vulnerability that affects the Windows Help and Support Center in Windows XP and Windows Server 2003, was a dominant exploit in 2010, but declined significantly in 1H11. Microsoft issued [Security Bulletin MS10-042](#) in July 2010 to address the issue.

Adobe Flash Player Exploits

Figure 26 shows the prevalence of different Adobe Flash exploits by quarter.

Figure 26. Adobe Flash Player exploits detected and blocked by Microsoft antimalware products, 3Q10–2Q11



- Exploitation of Adobe Flash Player increased dramatically in 2Q11 with the disclosure of two new vulnerabilities, [CVE-2011-0611](#) and [CVE-2011-2110](#).
- [CVE-2011-0611](#) was discovered in April 2011 when it was observed being exploited in the wild, typically in the form of malicious .zip files attached to spam email messages that purported to contain information about the Fukushima Daiichi nuclear disaster in Japan. Adobe Systems released [Security Bulletin APSP11-07](#) on April 15 and [Security Bulletin APSP11-08](#) on April 21 to address the issue. On the same day the security update was released, attacks that targeted the vulnerability skyrocketed and remained high for several days, most of which were detected on computers in Korea. About a month later, a second increase in attacks was observed, affecting multiple locations.
- [CVE-2011-2110](#) was discovered in June 2011, and Adobe released [Security Bulletin APSP11-18](#) on June 15 to address the issue. As with CVE-2011-0611, attacks that targeted the vulnerability spiked just after the security update was released, again with most of the targeted computers located in Korea.
- See page 15 for more information about these two vulnerabilities, as well as the following posts on the MMPC blog (blogs.technet.com/mmpc):
 - [Analysis of the CVE-2011-0611 Adobe Flash Player vulnerability exploitation](#) (April 12, 2011)
 - [Exploits for CVE-2011-2110 focus on Korea](#) (June 21, 2011)

Malware and Potentially Unwanted Software

Except where specified, the information in this section was compiled from telemetry data that was generated from more than 600 million computers worldwide and some of the busiest Internet online services. (See “Appendix B: Data Sources” on page 122 for more information about the telemetry used in this report.)

CCM Calculation Changes

This volume of the *Microsoft Security Intelligence Report (SIR)* introduces a significant change in the way location is determined for computers whose administrators have opted into providing telemetry data to Microsoft. In previous volumes of the report, Windows-based computers reporting information were classified by countries and regions according to the administrator-specified setting under the Location tab or menu in Region and Language in Control Panel. Beginning with this volume of the report, location is determined by geolocation of the IP address used by the computer submitting the telemetry data. (For more information about how location data is collected and used, see “Appendix B: Data Sources” on page 122.)⁵

Using IP addresses to determine the location of systems sharing telemetry instead of using the administrator-specified Location setting of the computer creates slight differences in the trends observed in most countries/regions reported in the SIR. In a few cases, the reported infection rate has changed significantly. Figure 27 and Figure 28 show trends for the locations with the largest CCM decreases and increases caused by the switch to IP geolocation. (CCM stands for *computers cleaned per mille*, or thousand, and represents the number of reported computers cleaned in a quarter for every 1,000 executions of the Malicious Software Removal Tool (MSRT). For example, if the MSRT has 50,000 executions in a particular

⁵ In addition to the geographic changes described here, Microsoft has corrected an error in data tabulation that had caused the worldwide CCM to be reported inaccurately in previous volumes of this report. See the [Microsoft Security Intelligence Report website](#) for more information about this change.

location in the first quarter of the year and removes infections from 200 computers, the CCM for that location in the first quarter is 4.0, or $200 \div 50,000 \times 1,000$.)

Figure 27. The five locations with the largest CCM decreases caused by the switch to IP geolocation

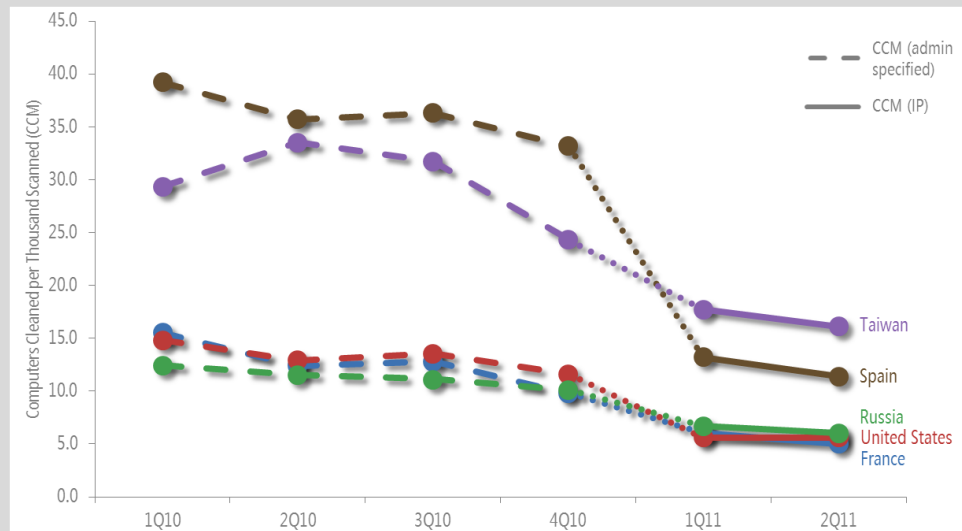
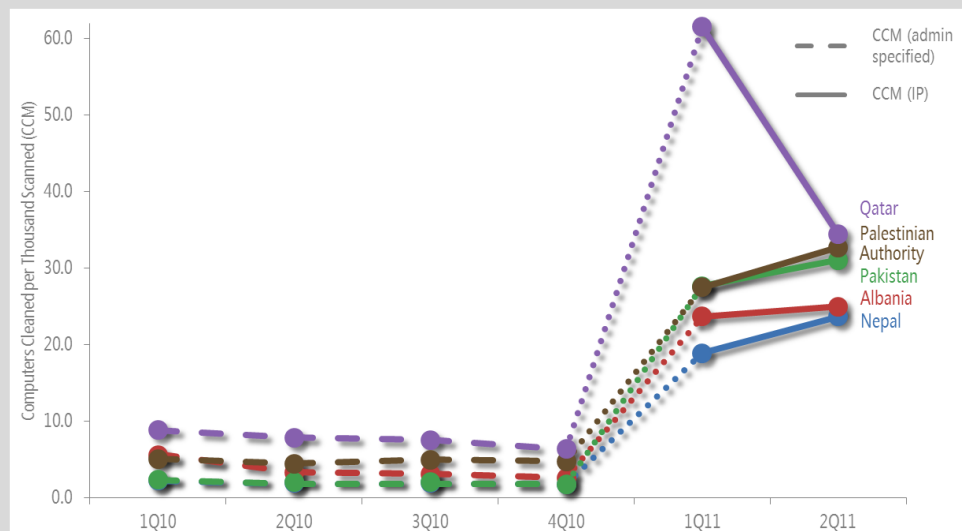


Figure 28. The five locations with the largest CCM increases caused by the switch to IP geolocation



In addition to providing what Microsoft believes will be a more accurate gauge of regional infection rates, this change provides an interesting perspective on computer usage habits around the world.

Very few locations saw their infection rates fall as a result of the switch to IP geolocation—in fact, among locations with at least 100,000 MSRT executions in 1Q11, the five shown in Figure 27 were the only locations that underwent a CCM decrease greater than 1.0 point.

By contrast, there were more than 100 locations whose CCMs rose after applying IP geolocation, with 35 of them moving 10 points or more, and four rising more than 20 points, as shown in Figure 28. In general, most of the locations with significant increases have smaller populations and relatively few reporting computers. The 61.5 CCM for Qatar in 1Q11 is the largest CCM figure ever reported in the *Microsoft Security Intelligence Report*, and is 55.1 points higher than the figure reported for Qatar for 4Q10 using the administrator-configured locale setting to determine location.

Notably, the five locations in which the CCM decreased significantly represent the largest populations using five of the most widely used languages on the Internet: France and French, Spain and Spanish, Russia and Russian, Taiwan and Chinese (Traditional), and the United States and English. This finding suggests that, rather than using the locale settings designated for their country or region, many computer administrators in smaller locations might be using locale settings for larger ones, particularly larger locations in which the dominant language is one spoken by the computer's user. As a result, the reported infection rates were being skewed for some locations. For example, if a Spanish-speaking computer administrator outside Spain configured a computer with the locale settings for Spain, any malware detections on the computer would have been reported for Spain using the previous method for determining location. This factor would have the effect of overreporting malware detections for Spain, and underreporting malware detections for the country or region in which the computer was actually located. Switching to IP address-based geolocation corrects this anomaly and provides more accurate regional infection statistics.

Computer security and response professionals in the more affected locations should consider these findings carefully when developing plans for safeguarding their populations' computers. (See [Managing Risk](#) at the *Microsoft Security Intelligence Report* website for guidance about protecting computers, software, and people from threats.)

Global Infection Rates

The telemetry data generated by Microsoft security products from administrators or users who choose to opt in to data collection includes information about the

location of the computer, as determined by IP geolocation. This data makes it possible to compare infection rates, patterns, and trends in different locations around the world.

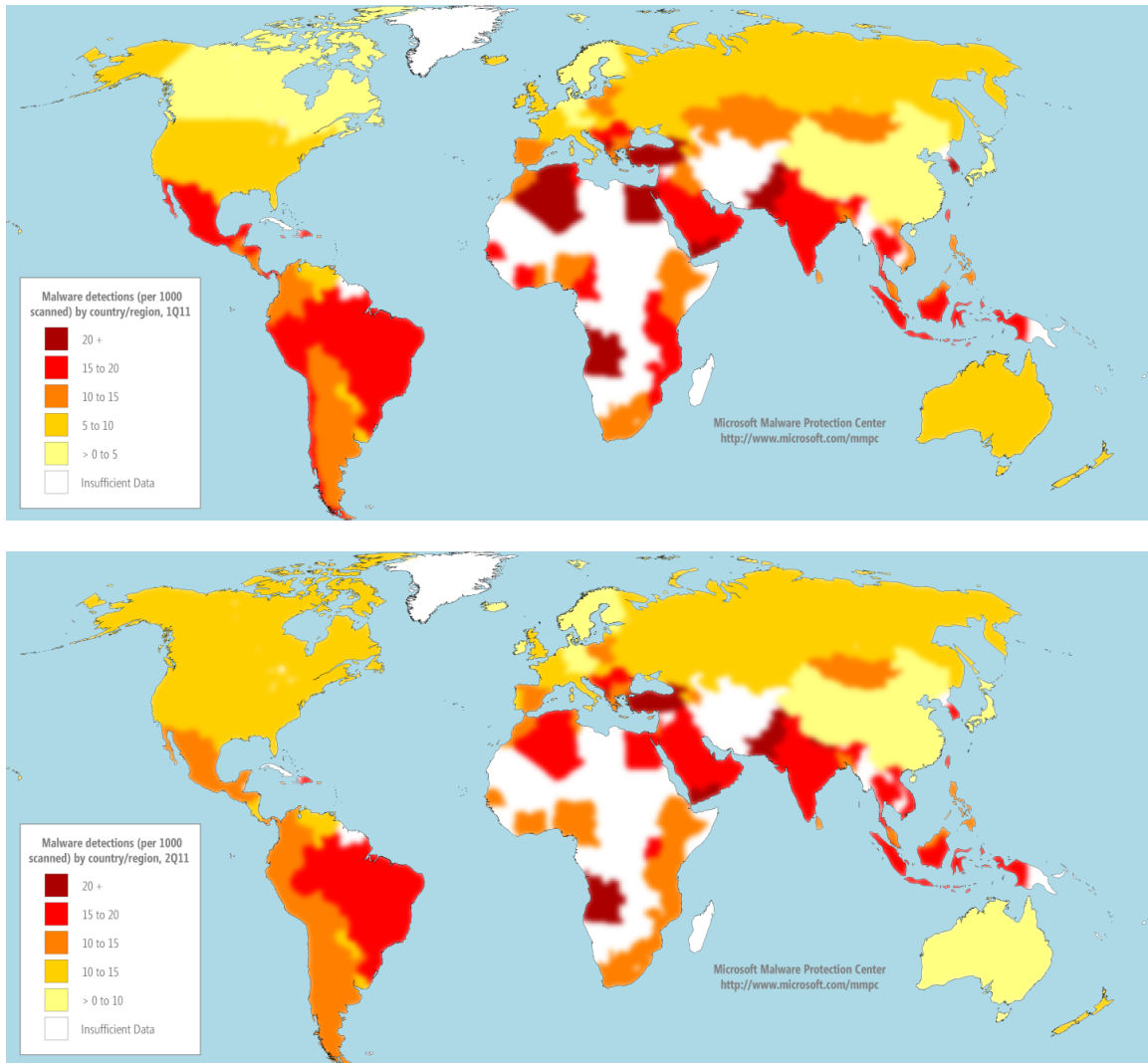
Figure 29. The locations with the most computers reporting detections and removals by Microsoft desktop antimalware products in 1H11

	Country/Region	1Q11	2Q11	Chg. 1Q to 2Q
1	United States	10,727,964	10,471,335	-2.4% ▼
2	Brazil	3,463,973	3,724,844	7.5% ▲
3	France	2,351,941	2,674,775	13.7% ▲
4	United Kingdom	2,175,201	2,089,883	-3.9% ▼
5	China	2,017,682	1,883,578	-6.6% ▼
6	Germany	1,622,081	1,530,551	-5.6% ▼
7	Russia	1,296,208	1,583,857	22.2% ▲
8	Italy	1,358,166	1,509,148	11.1% ▲
9	Canada	1,377,173	1,353,164	-1.7% ▼
10	Turkey	1,248,978	1,359,181	8.8% ▲

- In absolute terms, the locations with the most computers reporting detections tend to be ones with large populations and large numbers of computers.
- Detections in Russia increased 22.2 percent from 1Q11 to 2Q11, mostly because of increased detections of [Win32/Pameseg](#), a potentially unwanted software program with a Russian language user interface.
- Detections in France and Italy both increased significantly in 2Q11 because of increased detections of a number of Adware families, including [Win32/ClickPotato](#), [Win32/Hotbar](#), and [Win32/OfferBox](#).
- Detections in China decreased 6.6 percent, primarily because of steep drops in detections of a pair of malware families, [JS/ShellCode](#) and [Win32/Sogou](#), that have historically been much more common in China than elsewhere.

For a different perspective on infection patterns worldwide, Figure 30 shows the infection rates in locations around the world using CCM.

Figure 30. Infection rates by country/region in 1Q11 (top) and 2Q11 (bottom), by CCM



Detections and removals in individual countries/regions can vary significantly from quarter to quarter. Increases in the number of computers with detections can be caused not only by increased prevalence of malware in that country but also by improvements in the ability of Microsoft antimalware solutions to detect malware. Large numbers of new antimalware installations in a location also typically increase the number of computers cleaned in that location.

The next two figures illustrate infection rate trends for specific locations around the world, relative to the trends for all locations with at least 100,000 MSRT executions each quarter in 1H11.

Figure 31. Trends for the five locations with the highest infection rates in 2Q11, by CCM (100,000 MSRT executions minimum per quarter in 2011)

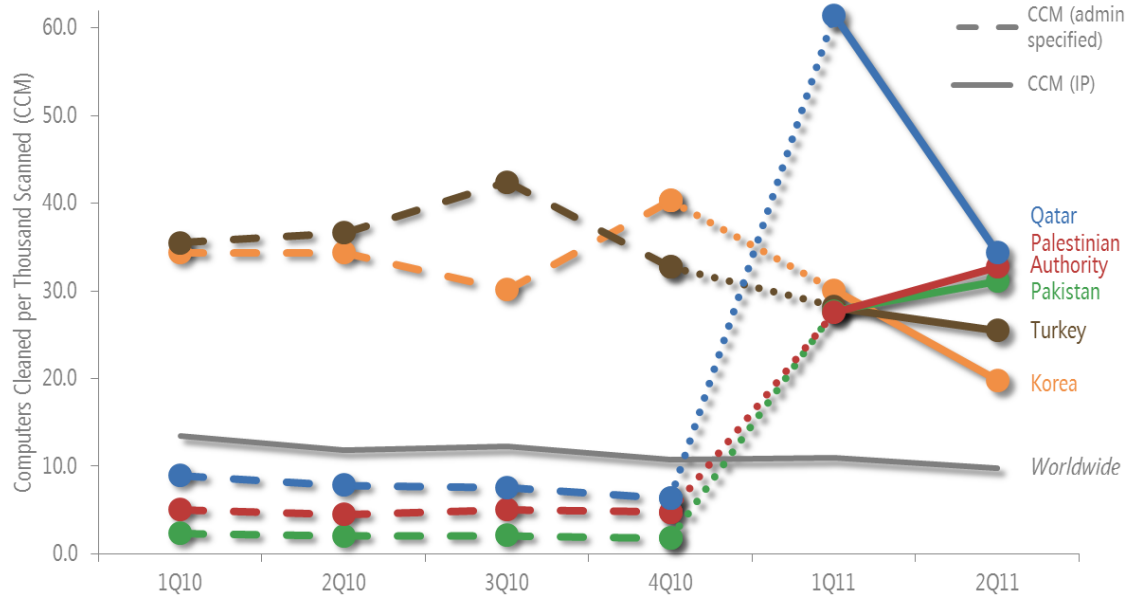
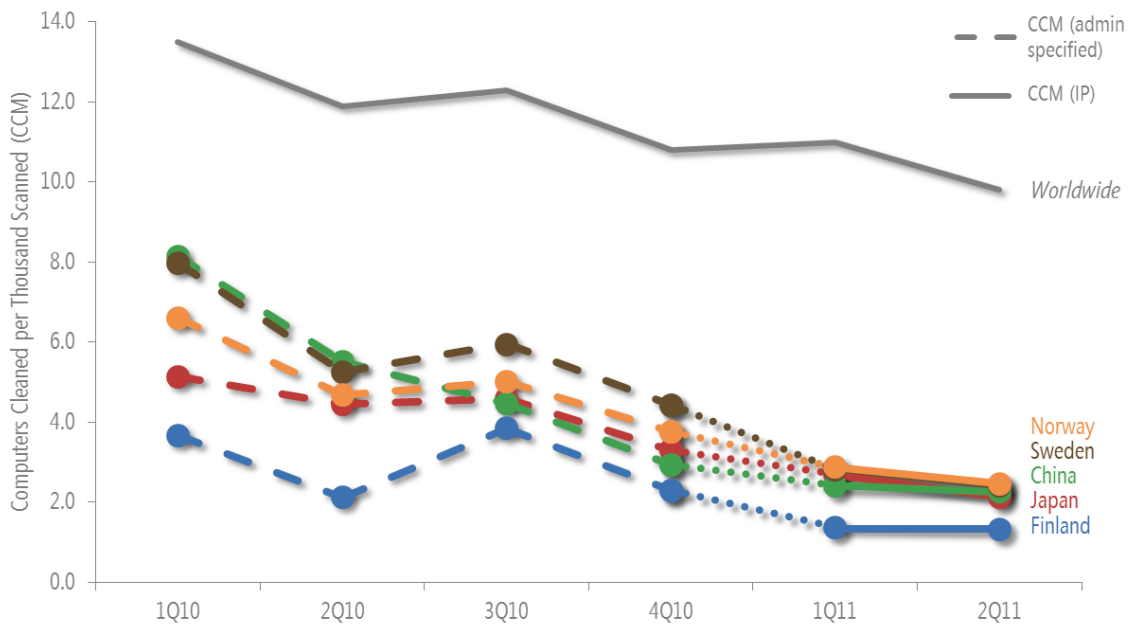


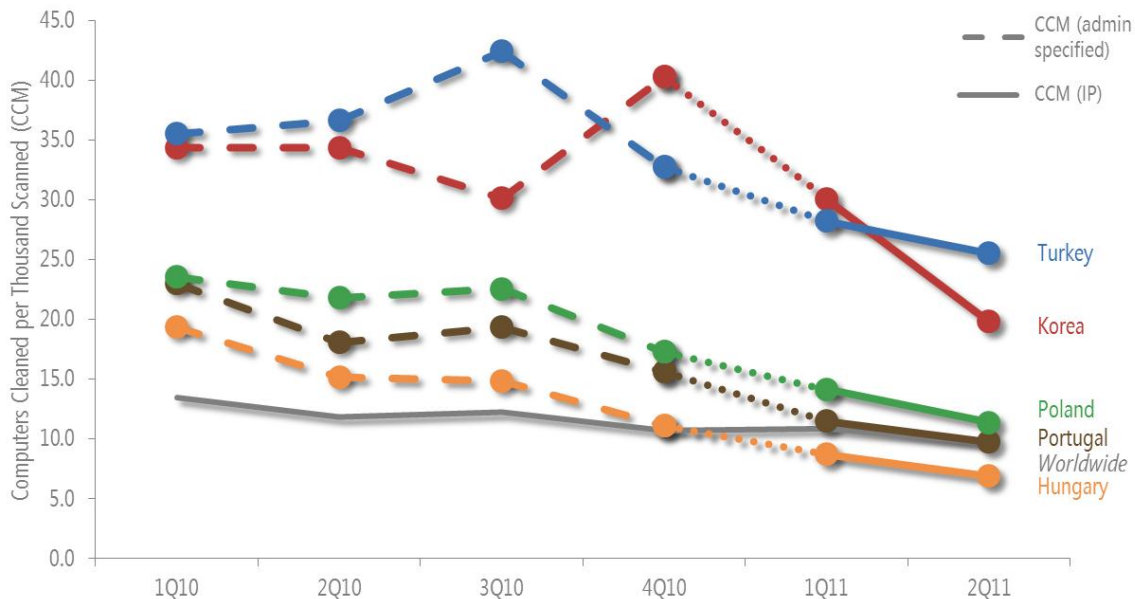
Figure 32. Trends for the five locations with the lowest infection rates in 2Q11, by CCM (100,000 MSRT executions minimum per quarter in 2011)



- The switch from using the administrator-configured location setting to IP address geolocation for classifying computers by country and region (see page 49) is responsible for the significant shifts in Figure 31 between 4Q10 and 1Q11.
- Of the five locations with the highest infection rates in 4Q10—Korea, Spain, Turkey, Taiwan, and Brazil—only Turkey and Korea are on the list for 2Q11. Spain and Taiwan underwent significant decreases with the shift to IP geolocation, and Brazil continued a trend of significant improvement over the last two years.
- Several Nordic countries were among the locations with the lowest infection rates, including Norway, Sweden, and Finland, as shown in Figure 32. Denmark, another Nordic country, had the sixth lowest infection rate in 2Q11.
- Although China is one of the locations with the lowest infection rates worldwide as measured by CCM, a number of factors that are unique to China are important to consider when assessing the state of computer security there. The malware ecosystem in China is dominated by a number of Chinese-language threats that are not prevalent anywhere else. The CCM figures are calculated based on telemetry from the MSRT, which tends to target malware families that are prevalent globally. As a result, many of the more prevalent threats in China are not represented in the data used to calculate CCM. For a more in-depth perspective on the threat landscape in China, see the “[Regional Threat Assessment](#)” section of the *Microsoft Security Intelligence Report* website.

As explained in “CCM Calculation Changes” on page 49, the shift from using administrator-configured location settings to IP address-based geolocation has resulted in significant CCM changes for some countries or regions. To help illustrate which locations improved the most in the first half of 2011, Figure 33 focuses on locations that were not significantly affected by the change. All of the locations shown in Figure 33 are ones in which the 1Q11 infection rate as determined by IP address geolocation differed by less than one percentage point from the 1Q11 infection rate as determined by administrator-configured settings.

Figure 33. Trends for five locations with significant infection rate improvements in 1H11, by CCM (100,000 MSRT executions minimum per quarter in 2011)



Regional Effective Practices

Computer emergency response teams (CERTs) and computer security incident response teams (CSIRTs) around the world work to protect technology users in their regions. Over time, effective practices that help reduce regional malware infections have emerged. Microsoft asked representatives from some of these teams to share insights into their practices:

- In Korea, the Korea Information Security Agency (KISA) has instituted a two-part remediation effort. The first part is a joint malware notification program developed in cooperation with major ISPs in Korea. KISA provides the participating ISPs with information about computers that are determined to be infected with malware families that are widespread within Korea. When the user of an infected computer logs in, a pop-up window displays with a link to a web page that contains instructions for removing the infection.

The second part of the remediation effort consists of a program to develop and distribute free “vaccine” software that targets specific malware families that are widespread in Korea. Responding to a series of serious distributed denial-of-service (DDoS) attacks that have affected Korea recently, KISA contracted with major domestic antivirus (AV) vendors to develop the vaccine, which is available for download from www.boho.or.kr.

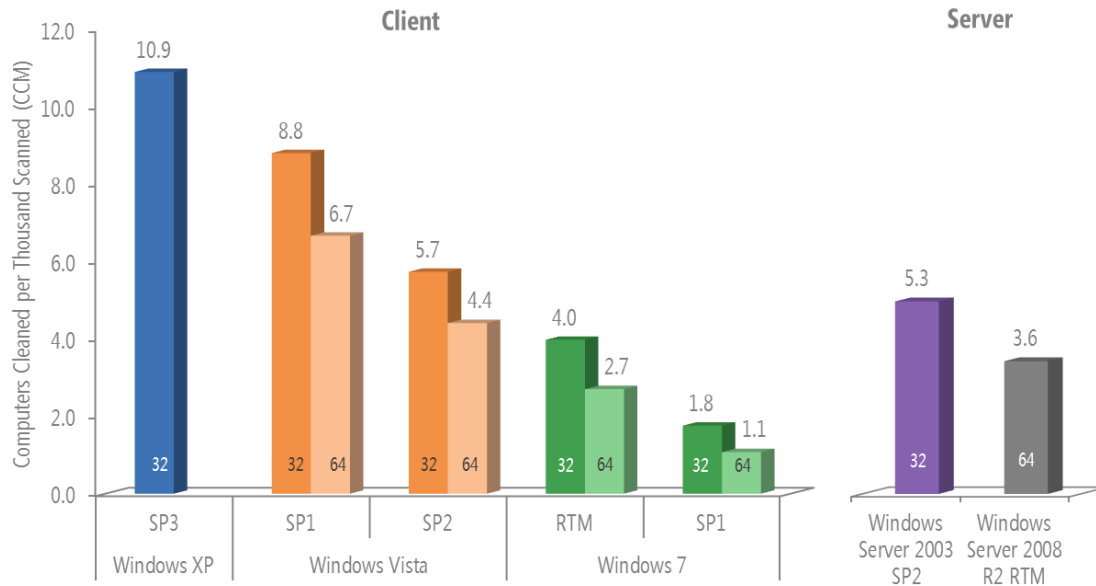
- In Poland, CERT Polska (www.cert.pl) attributes much of the improvement to filtering of port 25, used for Simple Mail Transfer Protocol (SMTP) traffic, by Telekomunikacja Polska, Poland's largest telecommunications provider. SMTP is often abused by malware to send spam and spread infection. Cable Internet providers in Poland have also become more effective at stopping malware and distributing antivirus software to their users. CERT Polska published its annual security report for 2010 at www.cert.pl/PDF/Raport_CP_2010.pdf, and an English-language summary at www.cert.pl/news/3410/langswitch_lang/en.
- In Portugal, infections have decreased significantly since the creation of the National Network of CSIRTs. The Serviço de Resposta a Incidentes de Segurança Informática (CERT.PT) launched the network in 2008 in cooperation with technology companies, telecom providers, and government agencies to address the need for a national response capability for computer security incidents affecting Portugal. As the network has grown and achieved wider recognition, new CSIRTs have been created within ISPs, financial institutions, the Portuguese armed forces, and other companies and agencies.

In 2011, CERT.PT began sending network members a weekly digest of infected systems within their networks, using data from a range of sources including honeynets, the Shadowserver Foundation, and telemetry provided by Microsoft related to the Rustock botnet. (See *Battling the Rustock Threat*, available from the Microsoft Download Center, for more information about Rustock and Microsoft efforts to fight the botnet.)

Operating System Infection Rates

The features and updates that are 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 for the different versions and service packs. Figure 34 shows the infection rate for each currently supported Windows operating system/service pack combination that accounted for at least 0.1 percent of total MSRT executions in 2Q11.

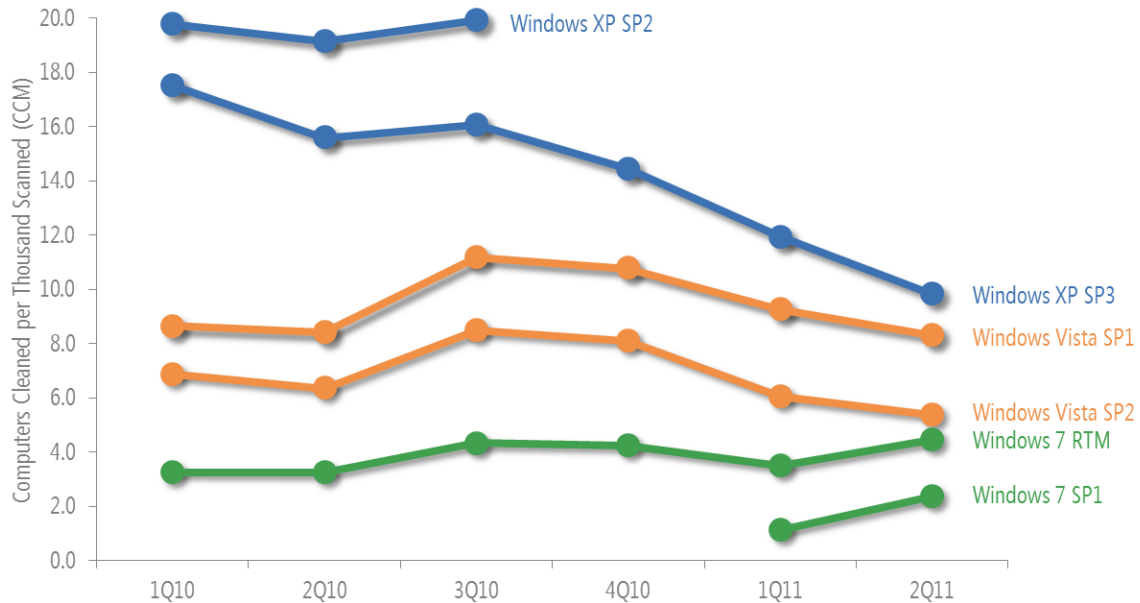
Figure 34. Infection rate (CCM) by operating system and service pack in 2Q11



"32" = 32-bit edition; "64" = 64-bit edition. SP = Service Pack. Supported operating systems with at least 0.1 percent of total executions in 2Q11 shown.

- This data is normalized: the infection rate for each version of Windows is calculated by comparing an equal number of computers per version (for example, 1,000 Windows XP SP3 computers to 1,000 Windows 7 RTM computers).
- As in previous periods, infection rates for more recently released operating systems and service packs are consistently lower than earlier ones, for both client and server platforms. Windows 7 and Windows Server 2008 R2, the most recently released Windows client and server versions, respectively, have the lowest infection rates on the chart.
- Infection rates for the 64-bit versions of Windows Vista and Windows 7 are lower than for the corresponding 32-bit versions of those operating systems. One reason might be that 64-bit versions of Windows still appeal to a more technically savvy audience than their 32-bit counterparts, despite increasing sales of 64-bit Windows versions among the general computing population. Kernel Patch Protection (KPP), a feature of 64-bit versions of Windows that protects the kernel from unauthorized modification, might also contribute to the discrepancy by preventing certain types of malware from functioning.

Figure 35. CCM trends for currently and recently supported 32-bit versions of Windows XP, Windows Vista, and Windows 7, 1Q10–2Q11



- Newer operating systems and service packs consistently have lower infection rates than their older counterparts, with Windows 7 having the lowest infection rates of any client version of Windows.
- Infection rates for Windows XP SP3 and Windows Vista declined following the February 2011 release of a security update that changed the way the AutoRun feature works on those platforms to match its functionality in Windows 7. (See page 13 for more information about this change.) The impact of this change can be seen in the infection statistics for [Win32/Rimecud](#), the ninth most commonly detected family worldwide in 1H11 and one of the top abusers of the AutoPlay feature.

Figure 36. Increase or decrease of Win32/Rimecud detections with different operating system/service pack combinations

Platform	CCM Change
Windows XP SP3	-2.7 ▼
Windows Vista SP1	-1.3 ▼
Windows Vista SP2	-2.2 ▼
Windows 7	-0.1 ▼

Windows XP SP3 and the two supported Windows Vista service packs received the AutoRun update, and detections of Rimecud on those platforms

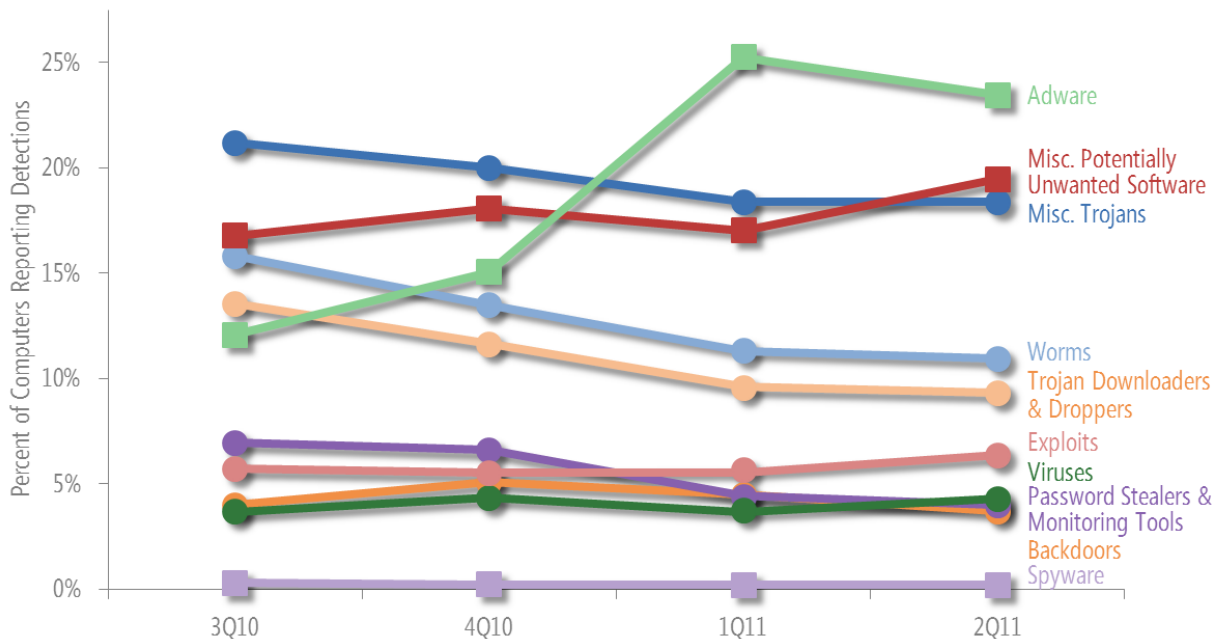
went down by an average of 2.1 computers cleaned per 1000 scanned by the MSRT. Windows 7 already included the more secure AutoPlay functionality; consequently, detections of Rimecud were nearly unchanged.

- Infection rates for Windows 7 RTM and SP1 were higher in 2Q11, primarily because of increased detections of a number of virus and worm families, notably [Win32/Sality](#), [Win32/Ramnit](#), [Win32/Brontok](#), and [Win32/Nuqel](#). Detections of most of these families also increased on Windows XP and Windows Vista, although the infection rates for those platforms decreased overall because of the AutoPlay change discussed earlier.

Threat Categories

The Microsoft Malware Protection Center (MMPC) classifies individual threats into 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 *Microsoft Security Intelligence Report* groups these types into 10 categories based on similarities in function and purpose.

Figure 37. Detections by threat category 3Q10–2Q11, by percentage of all computers reporting detections



Round markers indicate malware categories; square markers indicate potentially unwanted software categories.

- Totals for each time period may exceed 100 percent because some computers report more than one category of threat in each time period.
- Adware rose to become the most commonly detected category in 1Q11 and 2Q11, primarily because of a pair of new families, [Win32/OpenCandy](#) and [Win32/ShopperReports](#), and large increases in detections of a number of older families. See “Threat Families” on page 63 for more information.
- A small increase in detections of Miscellaneous Potentially Unwanted Software families, notably [Win32/Keygen](#), made it the second most commonly detected category in 2Q11, just ahead of Miscellaneous Trojans.
- Worms and Trojan Downloaders & Droppers were two of the more significant categories in 2010, but declined to 10.9 percent and 9.3 percent of detections by 2Q11, respectively. A change in the functionality of the AutoRun feature in older versions of Windows implemented in February 2011 was followed by drops in detections of a number of worm families, contributing to the decline seen here. (See page 13 for more information about the AutoRun change.)

Threat Categories By Location

There are significant differences in the types of threats that affect users in different parts of the world. The spread of malware and its effectiveness are highly dependent on language and cultural factors, in addition to the methods used for distribution. Some threats are spread using techniques that target people who speak a particular language or who use online services that are local to a specific geographic region. Other threats target vulnerabilities or operating system configurations and applications that are unequally distributed around the globe.

Figure 38 shows the relative prevalence of different categories of malware and potentially unwanted software in several locations around the world in 2Q11.

Figure 38. Threat category prevalence worldwide and in 10 individual locations, 2Q11

Category	World	US	Brazil	Fr.	UK	China	Ger.	Russ.	Italy	Can.	Tur.
Adware	37.0%	39.7%	26.1%	72.4%	49.1%	5.3%	44.1%	9.7%	60.0%	45.8%	37.7%
Misc. Potentially Unwanted Software	30.6%	22.1%	35.2%	27.7%	27.9%	48.8%	26.5%	60.3%	26.1%	26.7%	34.7%
Misc. Trojans	28.9%	38.9%	22.6%	12.1%	31.9%	36.6%	25.4%	34.1%	15.5%	36.2%	41.9%
Worms	17.2%	6.3%	24.2%	7.3%	5.9%	14.0%	8.6%	19.9%	11.9%	5.0%	31.3%
Trojan Downloaders & Droppers	14.7%	17.8%	21.0%	7.0%	13.8%	20.4%	13.4%	9.7%	9.1%	17.4%	13.5%
Exploits	10.0%	14.4%	16.3%	2.7%	10.5%	15.0%	7.9%	7.1%	4.0%	13.1%	3.4%
Viruses	6.7%	2.0%	10.1%	1.2%	3.4%	8.0%	2.9%	8.4%	1.7%	2.0%	17.7%
Password Stealers & Monitoring Tools	6.3%	2.9%	18.9%	2.4%	3.9%	4.8%	6.8%	5.1%	4.2%	2.8%	7.8%
Backdoors	5.8%	4.8%	7.7%	3.3%	3.9%	8.4%	5.8%	6.3%	7.1%	4.6%	5.4%
Spyware	0.3%	0.4%	0.1%	0.1%	0.2%	1.8%	0.2%	0.3%	0.1%	0.3%	0.1%

Totals for each location may exceed 100 percent because some computers reported threats from more than one category.

- Within each row of Figure 38, a darker color indicates that the category is more prevalent in the specified location than in the others, and a lighter color indicates that the category is less prevalent.
- The United States and the United Kingdom, two predominantly English-speaking locations that also share a number of other cultural similarities, have similar threat mixes in most categories.
- Although France had lower than average detection rates in most categories, adware was found on 72.4 percent of computers reporting detections, a rate nearly twice as high as the worldwide average. The top 6 families detected in France in 2Q11 were adware families, with all other categories far behind. (See the [Microsoft Security Intelligence Report website](#) for additional details.)
- Italy experienced a rise in Adware detections similar to that of France, because of increased detections of many of the same families. A new family, [Adware:Win32/OfferBox](#), was the top family in both France and Italy in 2Q11.
- Brazil has long had higher-than-average detections of Password Stealers & Monitoring Tools because of the prevalence of [Win32/Bancos](#), which targets customers of Brazilian banks. Detections of Password Stealers & Monitoring Tools are still high, but a number of other categories have also increased to significantly above average because of increased detections of families such as [JS/Pompop](#), [HTML/IframeRef](#), and [Win32/OpenCandy](#).

- China has a relatively high concentration of Miscellaneous Potentially Unwanted Software, Backdoors, and Spyware, and a relatively low concentration of Adware. China routinely exhibits a threat mix that is much different than those of other large countries and regions, featuring a number of Chinese-language families like [Win32/BaiduSobar](#) that are uncommon elsewhere. The most commonly detected families in China also include an exploit, [JS/CVE-2010-0806](#), that is less prevalent elsewhere.

See “Appendix C: Worldwide Infection Rates” on page 124 for more information about malware around the world.

Threat Families

Figure 39 lists the top 10 malware and potentially unwanted software families that were detected on computers by Microsoft antimalware desktop products in the first half of 2011.

Figure 39. Quarterly trends for the top 10 malware and potentially unwanted software families detected by Microsoft antimalware desktop products in 1Q11 and 2Q11, shaded according to relative prevalence

Family	Category	3Q10	4Q10	1Q11	2Q11
Win32/Hotbar	Adware	997,111	1,661,747	3,149,677	4,411,501
JS/Pornpop	Adware	2,659,054	3,666,856	4,706,968	4,330,510
Win32/Autorun	Worms	2,454,708	2,624,241	3,718,690	3,677,588
Win32/OpenCandy	Adware	—	—	6,797,012	3,652,658
Win32/ShopperReports	Adware	—	—	3,348,949	2,902,430
Win32/Keygen	Misc. Potentially Unwanted Software	981,051	1,402,417	2,299,870	2,680,354
Win32/ClickPotato	Adware	451,407	2,074,751	4,694,442	2,592,125
Win32/Zwangi	Misc. Potentially Unwanted Software	1,637,316	2,236,990	2,785,111	2,586,630
Win32/Rimecud	Misc. Trojans	1,673,312	1,872,449	2,123,298	1,818,530
Win32/Conficker	Worm	1,648,481	1,636,201	1,859,498	1,790,035

- [Win32/OpenCandy](#) was the most commonly detected family in 1H11 overall. OpenCandy is an adware program that may be bundled with certain third-party software installation programs, for which detection was first added in February 2011. Some versions of the OpenCandy program send user-specific information without obtaining adequate user consent, and these versions are detected by Microsoft antimalware products.

- [JS/Pornpop](#), the second most commonly detected family in 1H11 overall, is a detection for specially crafted JavaScript-enabled objects that attempt to display pop-under advertisements in users' web browsers. Initially, JS/Pornpop appeared exclusively on websites that contained adult content; however, it has since been observed to appear on websites that may contain no adult content whatsoever. First detected in August 2010, it grew quickly to become one of the most prevalent families in the world.
- [Win32/Hotbar](#), the most commonly detected family in 2Q11 and the third most commonly detected family in 1H11, is adware that installs a browser toolbar that displays targeted pop-up ads based on its monitoring of web browsing activities. Hotbar has existed for several years, but has increased significantly in prevalence beginning in 1Q11.
- [Win32/Autorun](#), the fourth most commonly detected family in 1H11, is a generic detection for worms that spread between mounted volumes using the AutoRun feature of Windows. AutoRun detections had been increasing steadily for several quarters before declining slightly in 2Q11, following the February release of a security update that changed the way the AutoPlay feature works in Windows XP and Windows Vista. (See page 13 for more information about this change.)
- The adware family [Win32/ClickPotato](#), the fifth most commonly detected family in 1H11, was first detected in August 2010 and rose quickly to occupy the third spot in 1Q11 before rapidly declining in 2Q11. ClickPotato is a program that displays pop-up and notification-style advertisements based on the user's browsing habits.

Rogue Security Software

Rogue security software has become one of the most common methods that attackers use to swindle money from victims. Rogue security software, also known as *scareware*, is software that appears to be beneficial from a security perspective but provides limited or no security, generates erroneous or misleading alerts, or attempts to lure users into participating in fraudulent transactions. These programs typically mimic the general look and feel of legitimate security software programs and claim to detect a large number of nonexistent threats while urging users to pay for the "full version" of the software to remove the threats. Attackers typically install rogue security software programs through exploits or other malware, or use social engineering to trick users into believing the programs are legitimate and useful. Some versions emulate the appearance of the Windows Security Center or unlawfully use trademarks and icons to misrepresent themselves. (See www.microsoft.com/security/antivirus/rogue.aspx for an

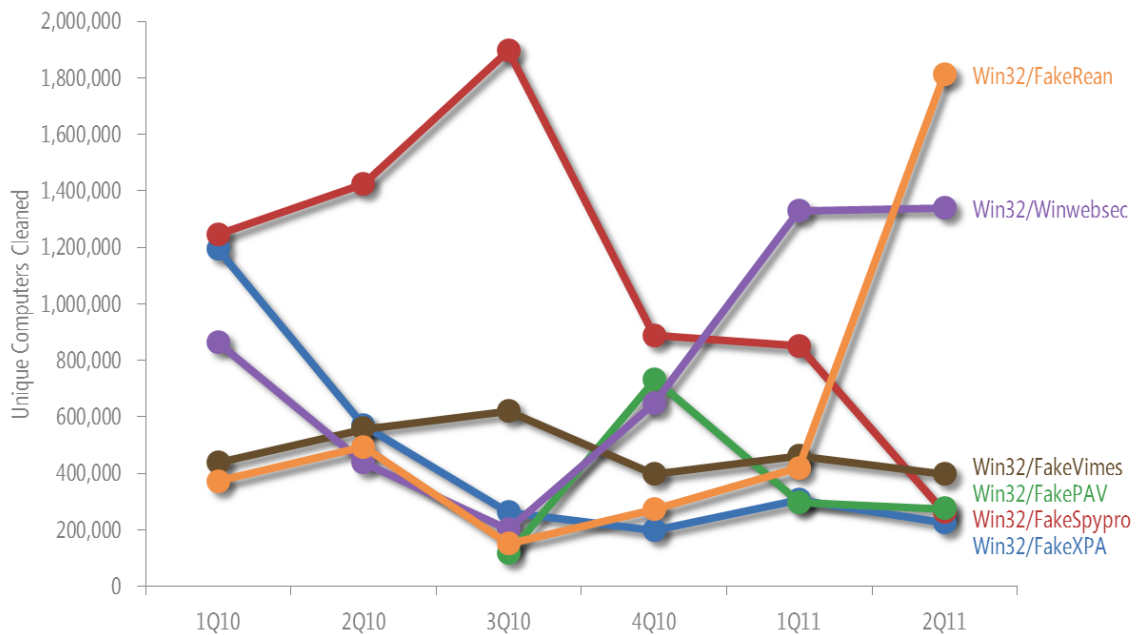
informative series of videos designed to educate a general audience about rogue security software.)

Figure 40. "Brands" used by a number of commonly detected rogue security software programs



Figure 41 shows detection trends for the most common rogue security software families detected in 1H11.

Figure 41. Trends for the most common rogue security software families detected in 1H11, by quarter



- Detections of Win32/FakeRean increased more than 300 percent from 1Q11 to 2Q11 to become the most commonly detected rogue security software family of the second quarter. As with a number of other rogue security

software families, FakeRean distributors sometimes concentrate their distribution efforts into discrete “campaigns,” which can lead to sudden spikes in detections like the one observed in 2Q11.

FakeRean has been distributed with several different names. The user interface and some other details vary to reflect each variant’s individual branding. Current variants of FakeRean choose a name at random, from a number of possibilities determined by the operating system of the affected computer. Detections for FakeRean were added to the MSRT in August 2009.

For more information about FakeRean, see the following entries in the MMPC blog (blogs.technet.com/mmpc):

- [Win32/FakeRean and MSRT](#) (August 11, 2009)
- [Win32/FakeRean is 33 rogues in 1](#) (March 9, 2010)
- As with FakeRean, detections of [Win32/Winwebsec](#) increased significantly in 2011, making it the second most commonly detected rogue security software family of 2Q11. Winwebsec has also been distributed under many names, with the user interface and other details varying to reflect each variant’s individual branding. These different distributions of the trojan use various installation methods, with filenames and system modifications that can differ from one variant to the next. The attackers behind Winwebsec are also believed to be responsible for [MacOS_X/FakeMacdef](#), the highly publicized “Mac Defender” rogue security software program for Apple Mac OS X that first appeared in May 2011. Detections for Winwebsec were added to the MSRT in May 2009.

For more information about the connection between Winwebsec and FakeMacdef, see the entry “[Winwebsec gang responsible for Fakemacdef?](#)” (May 17, 2011) in the MMPC blog.

- [Win32/FakeSpypro](#), the most commonly detected rogue security software family in 2010 by a wide margin, declined steeply beginning in 4Q10 to become only the fifth most prevalent rogue security software family in 2Q11. Names under which FakeSpypro is distributed include AntispywareSoft, Spyware Protect 2009, and Antivirus System PRO. Detections for FakeSpypro were added to MSRT in July 2009.

Home and Enterprise Threats

The usage patterns of home users and enterprise users tend to be very different. Enterprise users typically use computers to perform business functions while

connected to a network, and may have limitations placed on their Internet and email usage. Home users are more likely to connect to the Internet directly or through a home router and to use their computers for entertainment purposes, such as playing games, watching videos, shopping, and communicating with friends. These different usage patterns mean that home users tend to be exposed to a different mix of computer threats than enterprise users.

The infection telemetry data produced by Microsoft desktop antimalware products and tools includes information about whether the infected computer belongs to an Active Directory® Domain Services domain. Such domains are used almost exclusively in enterprise environments, and computers that do not belong to a domain are more likely to be used at home or in other non-enterprise contexts. Comparing the threats encountered by domain-joined computers and non-domain computers can provide insights into the different ways attackers target enterprise and home users and which threats are more likely to succeed in each environment.

Figure 42 and Figure 43 list the top 10 families detected on domain-joined and non-domain computers, respectively, in 2Q11.

Figure 42. Top 10 families detected on domain-joined computers, 3Q10–2Q11, by percentage of domain-joined computers reporting detections

	Family	Most Significant Category	3Q10	4Q10	1Q11	2Q11
1	Win32/Conficker	Worm	19.6%	18.9%	17.8%	15.8%
2	Win32/Autorun	Worm	10.0%	10.0%	11.7%	11.1%
3	Win32/Rimecud	Worm	8.0%	8.3%	8.1%	5.8%
4	Win32/OpenCandy	Adware	—	—	8.5%	4.9%
5	Win32/RealVNC	Misc. Potentially Unwanted Software	4.9%	4.3%	4.5%	4.4%
6	JS/Pornpop	Adware	3.4%	4.5%	4.4%	3.9%
7	Win32/Obfuscator	Misc. Trojans	1.9%	1.4%	3.4%	4.4%
8	Win32/Keygen	Misc. Potentially Unwanted Software	1.5%	2.2%	2.9%	3.5%
9	Java/CVE-2010-0840	Exploits	—	—	3.3%	3.1%
10	Win32/Sality	Viruses	2.5%	2.7%	2.7%	2.8%

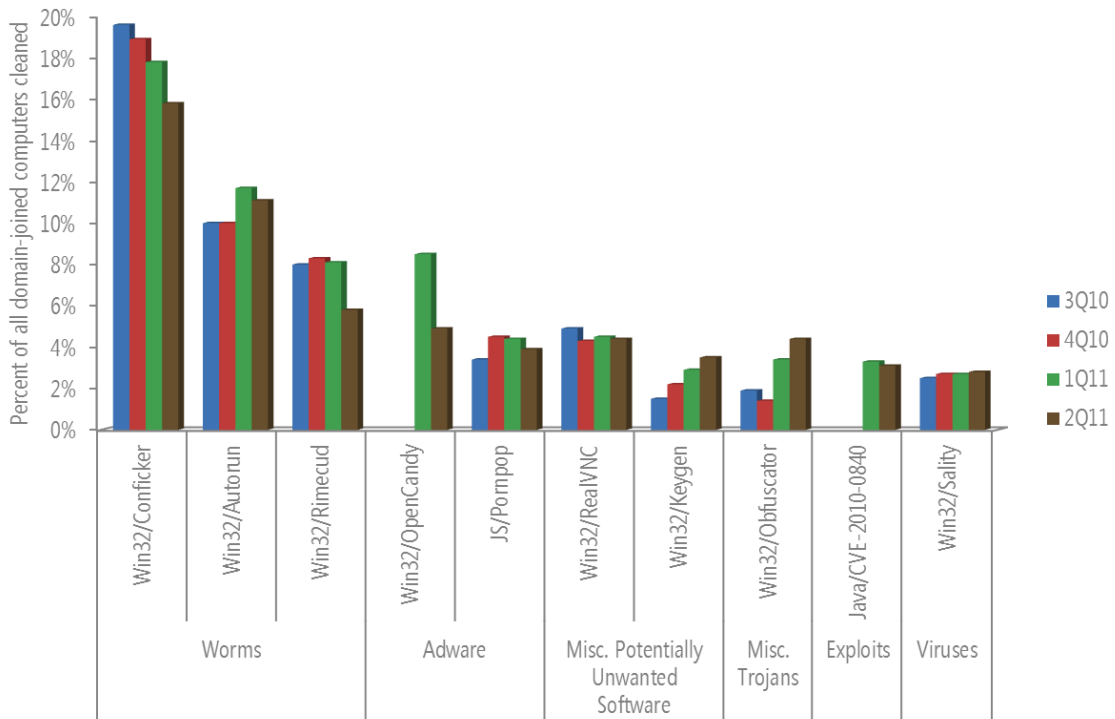
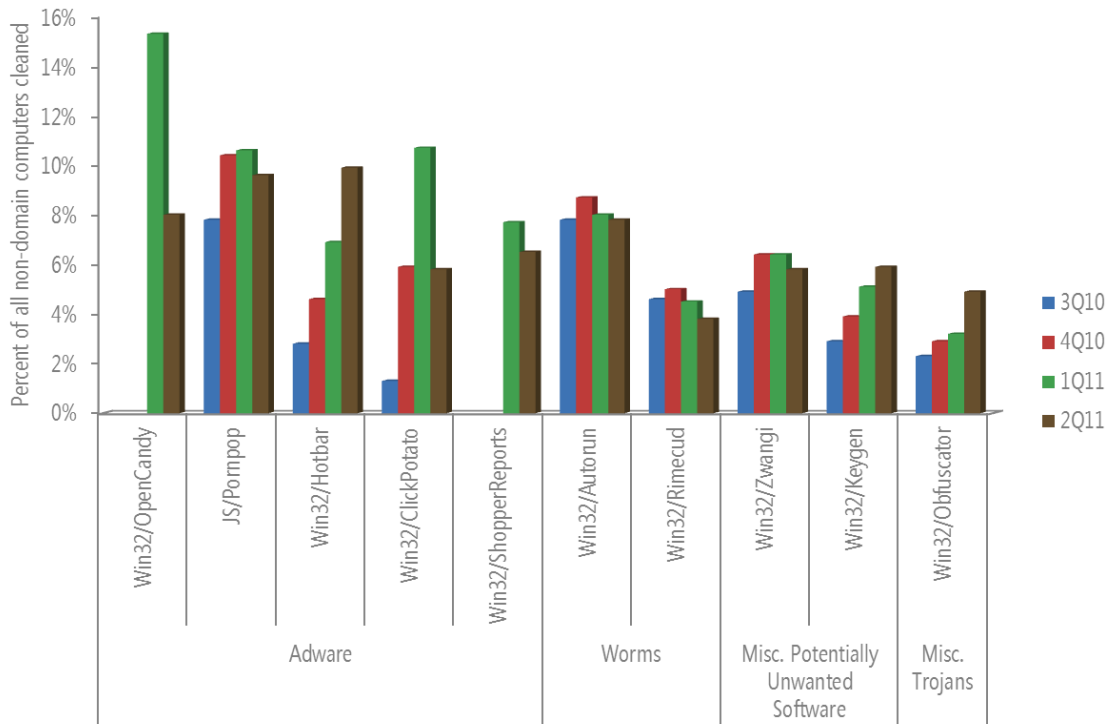


Figure 43. Top 10 families detected on non-domain computers, 3Q10–2Q11, by percentage of non-domain computers reporting detections

	Family	Most Significant Category	3Q10	4Q10	1Q11	2Q11
1	Win32/OpenCandy	Adware	—	—	15.3%	8.0%
2	JS/Pornpop	Adware	7.8%	10.4%	10.6%	9.6%
3	Win32/Hotbar	Adware	2.8%	4.6%	6.9%	9.9%
4	Win32/ClickPotato	Adware	1.3%	5.9%	10.7%	5.8%
5	Win32/Autorun	Worm	7.8%	8.7%	8.0%	7.8%
6	Win32/ShopperReports	Adware	—	—	7.7%	6.5%
7	Win32/Zwangi	Misc. Potentially Unwanted Software	4.9%	6.4%	6.4%	5.8%
8	Win32/Keygen	Misc. Potentially Unwanted Software	2.9%	3.9%	5.1%	5.9%
9	Win32/Rimecud	Worms	4.6%	5.0%	4.5%	3.8%
10	Win32/Obfuscator	Misc. Trojans	2.3%	2.9%	3.2%	4.9%



- Six families are common to both lists, although they are ordered differently and in different proportions. The generic detection [Win32/Autorun](#) and the adware family [Win32/OpenCandy](#) are high on both lists.

- Worms accounted for the top three families detected on domain-joined computers. [Win32/Conficker](#) and [Win32/Rimecud](#), the first and third families on the list, are both designed to propagate via network shares, which are common in domain environments. Conficker has declined slowly over the past four quarters, and dropped 2 percentage points between 1Q11 and 2Q11.
- Adware and potentially unwanted software account for 7 of the top 10 families detected on non-domain computers.
- Families that are significantly more prevalent on domain-joined computers include Conficker and the potentially unwanted software program [Win32/RealVNC](#). RealVNC is a program that enables a computer to be controlled remotely, similar to Remote Desktop Services. It has a number of legitimate uses, but attackers have also used it to gain control of users' computers for malicious purposes.
- [Java/CVE-2010-0840](#), an exploit that targets a vulnerability in older versions of Oracle Java SE and Java for Business, was the ninth most commonly detected threat on domain-joined computers. It is the only exploit to appear on either list. See “Java Exploits” on page 40 for more information about this exploit.
- The virus family [Win32/Sality](#), which was not among the top 10 families detected on domain-joined computers in 2010, ranks tenth in the latest chart. Detections of Sality have not significantly increased over the past four quarters, but significant declines in detections of formerly prevalent families such as [Win32/Taterf](#), [Win32/Hamweq](#), and [Win32/Renos](#) have enabled less common families like Sality to make the list.
- Families that are significantly more prevalent on non-domain computers include the adware families [Win32/Hotbar](#), [JS/Pornpop](#), and [Win32/ClickPotato](#), all of which display pop-up or pop-under advertisements in various contexts that may not be desired.
- As with domain-joined computers, a number of formerly prevalent families no longer appear on the list of the top threats detected on non-domain computers. Among these are the worm families Taterf and Conficker, and the rogue security software family [Win32/FakeSpypro](#).

Guidance: Defending Against Malware

Effectively protecting users from malware requires an active effort on the part of organizations and individuals. For in-depth guidance, see [Protecting Against Malicious and Potentially Unwanted Software](#) in the “Mitigating Risk” section of the *Microsoft Security Intelligence Report* website.

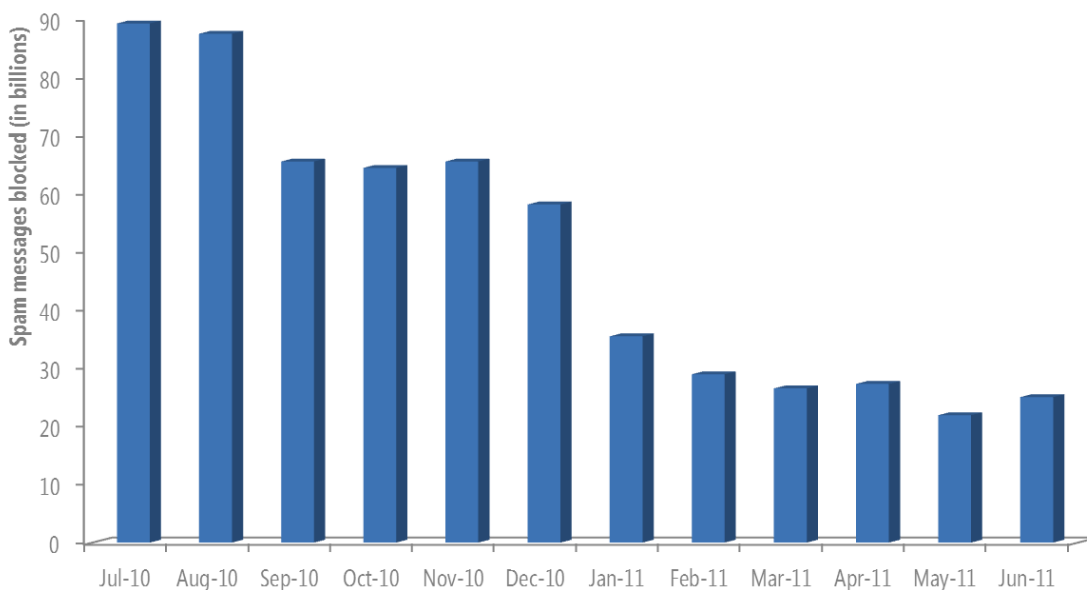
Email Threats

Most of the email messages sent over the Internet are unwanted. Not only does all this unwanted email tax recipients' inboxes and the resources of email providers, but it also creates an environment in which emailed malware attacks and phishing attempts can proliferate. Email providers, social networks, and other online communities have made blocking spam, phishing, and other email threats a top priority.

Spam Messages Blocked

The information in this section of the *Microsoft Security Intelligence Report* is compiled from telemetry data provided by Microsoft Forefront® Online Protection for Exchange (FOPE), which provides spam, phishing, and malware filtering services for thousands of Microsoft enterprise customers that process tens of billions of messages each month.

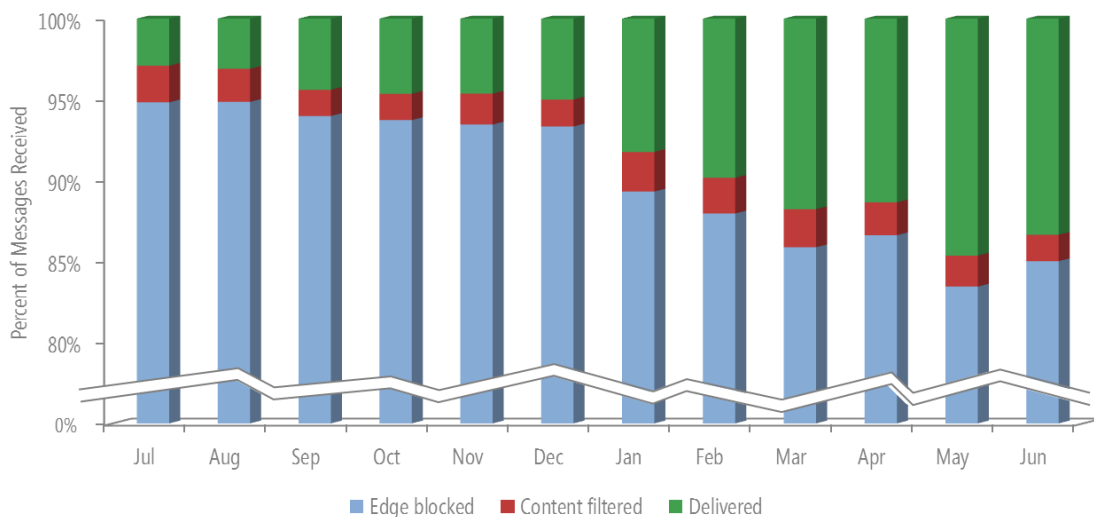
Figure 44. Messages blocked by FOPE each month from July 2010 to June 2011



- The volume of spam blocked by FOPE decreased dramatically over the past 12 months, from a high of 89.2 billion messages in July 2010 to a low of 21.9 billion in May 2011, primarily because of takedowns of two major botnets: Cutwail, which was shut down in August 2010, and Rustock, which was shut down in March 2011 following a period of dormancy that began in January.⁶
- The magnitude of this decrease suggests that coordinated takedown efforts such as the ones directed at Cutwail and Rustock can have a positive effect on improving the health of the email ecosystem.

FOPE performs spam filtering in two stages. Most spam is blocked by servers at the network edge, which use reputation filtering and other non-content-based rules to block spam or other unwanted messages. Messages that are not blocked at the first stage are scanned using content-based rules, which detect and filter many additional email threats, including attachments that contain malware.

Figure 45. Percentage of incoming messages blocked by FOPE using edge-blocking and content filtering from July 2010 to June 2011



- Between 85 and 95 percent of incoming messages were blocked at the network edge each month, which means that only 5 to 15 percent of incoming messages had to be subjected to the more resource-intensive content filtering process.

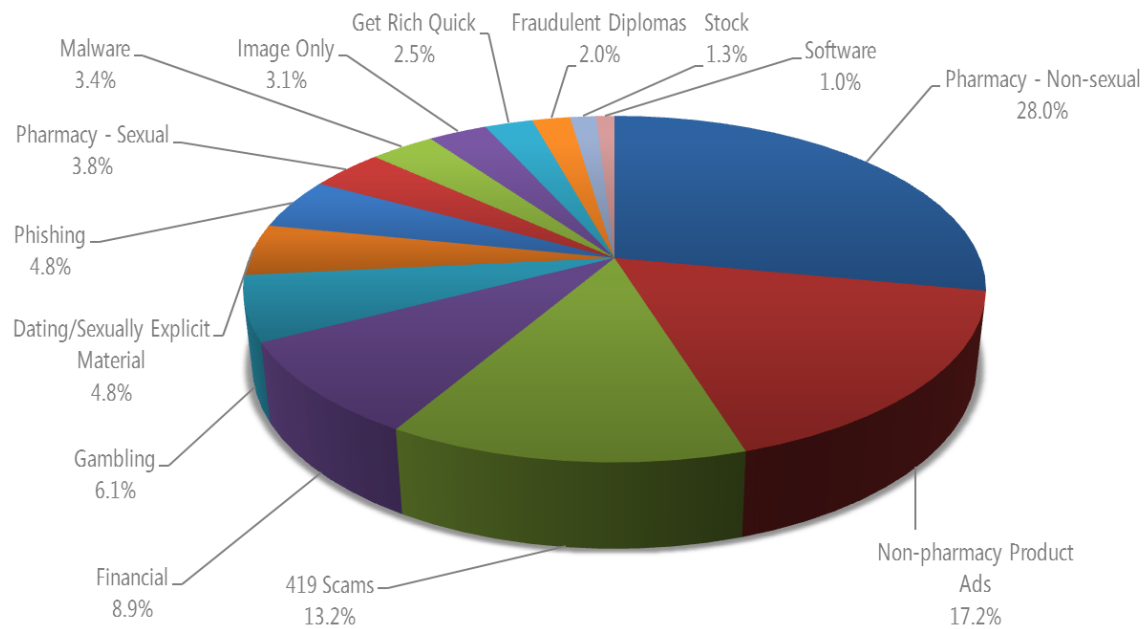
⁶ For more information about the Cutwail takedown, see [Microsoft Security Intelligence Report, Volume 10 \(July-December 2010\)](#). For more information about the Rustock takedown, see ["Battling the Rustock Threat,"](#) available from the Microsoft Download Center.

- The decline in the percentage of messages blocked at the network edge beginning in January was caused by the overall decline in the volume of spam that occurred following the inactivation of the Rustock botnet.

Spam Types

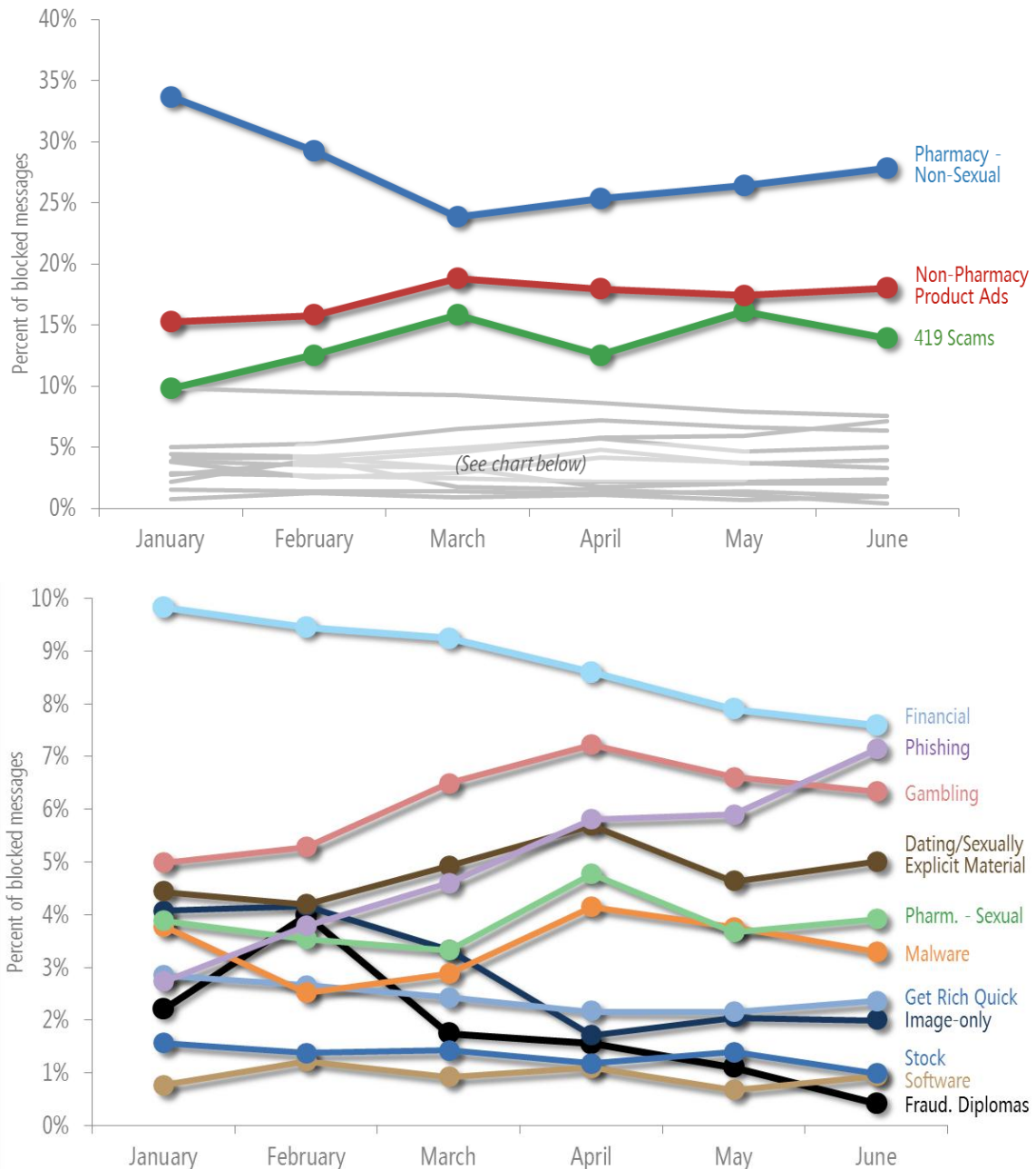
The FOPE content filters recognize several different common types of spam messages. Figure 46 shows the relative prevalence of these spam types in 1H11.

Figure 46. Inbound messages blocked by FOPE filters in 1H11, by category



- As in previous periods, advertisements for nonsexual pharmaceutical products (28.0 percent of the total) and nonpharmaceutical product advertisements (17.2 percent) accounted for the majority of the spam messages blocked by FOPE content filters in 1H11. Together with so-called “419” advance-fee loan scams (13.2 percent), these categories accounted for most of the spam messages that were blocked during the period. (See the [Microsoft Security Intelligence Report website](#) for more information about these scams.)
- In an effort to evade content filters, spammers sometimes send messages that consist only of one or more images, with no text in the body of the message. Image-only spam messages declined to 3.1 percent of the total in 1H11, down from 8.7 percent in 2010.

Figure 47. Inbound messages blocked by FOPE content filters each month in 1H11, by category



- Unlike in some recent periods, which showed evidence of individual spam “campaigns” featuring large volumes of certain types of spam for short periods of time, the increases and decreases of the spam categories tracked by FOPE were much more gradual from month to month. A possible exception involves

spam that advertises fraudulent university diplomas. Typically a low-volume category, fraudulent diploma spam increased to 4.0 percent of the total in February, following a much larger spike in volume that occurred around the same time in 2010.

- Phishing messages increased significantly over the period, going from 2.8 percent of the total in January to 7.2 percent in June. (See “Phishing Sites” on page 77 for more phishing-related statistics.)

Guidance: Defending Against Threats in Email

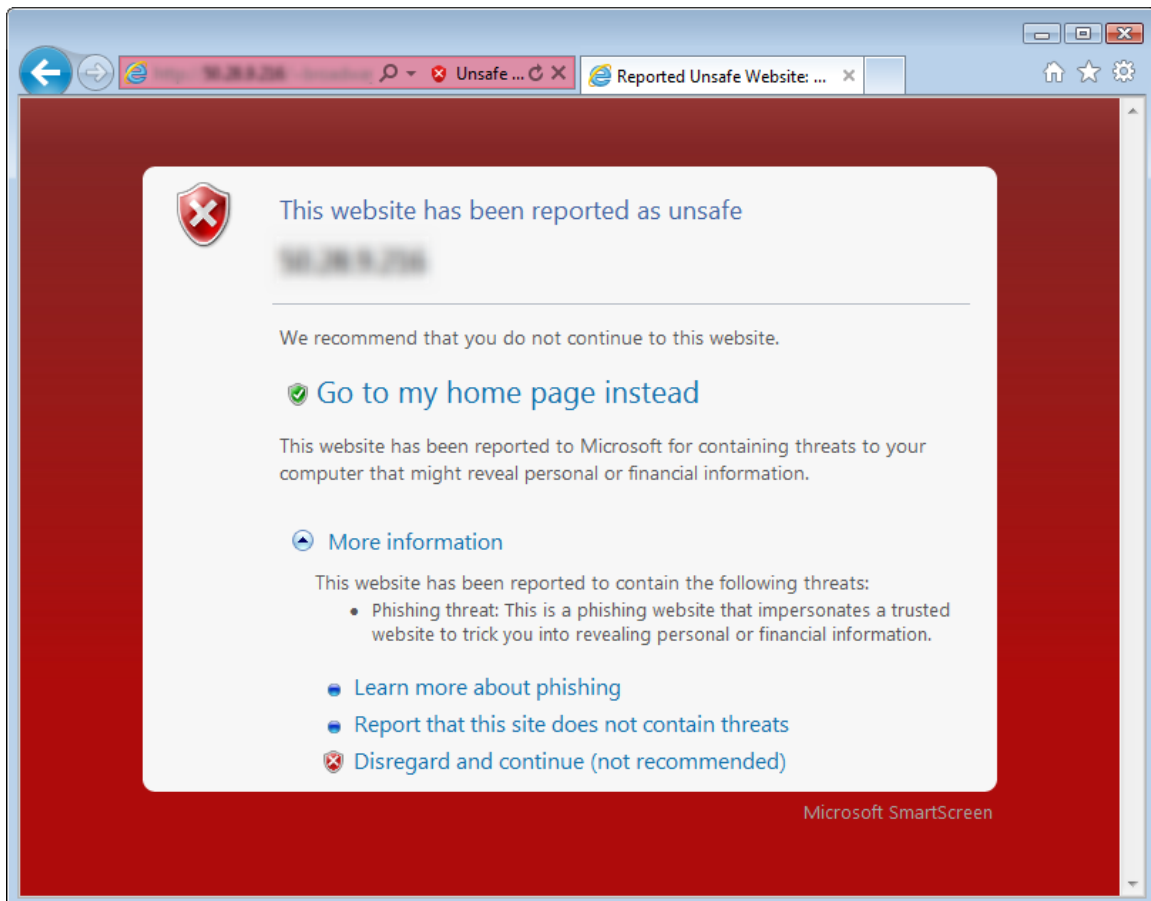
In addition to using a filtering service such as FOPE, organizations can take a number of steps to reduce the risks and inconvenience of unwanted email. Such steps include implementing email authentication techniques and observing best practices for sending and receiving email. For in-depth guidance, see [Guarding Against Email Threats](#) in the “Managing Risk” section of the *Microsoft Security Intelligence Report* website.

Malicious Websites

Attackers often use websites to conduct phishing attacks or distribute malware. Malicious websites typically appear completely legitimate and often provide no outward indicators of their malicious nature, even to experienced computer users. To help protect users from malicious webpages, Microsoft and other browser vendors have developed filters that keep track of sites that host malware and phishing attacks and display prominent warnings when users try to navigate to them.

The information in this section is compiled from a variety of internal and external sources, including telemetry data produced by SmartScreen® Filter (in Windows Internet Explorer 8 and 9), the Phishing Filter (in Internet Explorer 7), from a database of known active phishing and malware hosting sites reported by users of Internet Explorer and other Microsoft products and services, and from malware data provided by Microsoft antimalware technologies. (See “Appendix B: Data Sources” on page 122 for more information about the products and services that provided data for this report.)

Figure 48. SmartScreen Filter in Internet Explorer 8 and 9 blocks reported phishing and malware distribution sites to protect the user



Phishing Sites

Microsoft gathers information about phishing sites and impressions from *phishing impressions* generated by users who choose to enable the Phishing Filter or SmartScreen Filter in Internet Explorer. A phishing impression is a single instance of a user attempting to visit a known phishing site with Internet Explorer and being blocked, as illustrated in Figure 49.

Figure 49. How Microsoft tracks phishing impressions

1. The user views a phishing message, in email or elsewhere, and is tricked into clicking a link that leads to a malicious website.

2. SmartScreen Filter in Internet Explorer checks the Microsoft URL Reputation Service, determines that the website is malicious, and blocks it.

3. The URL Reputation Service records the anonymized details of the incident as a phishing impression.

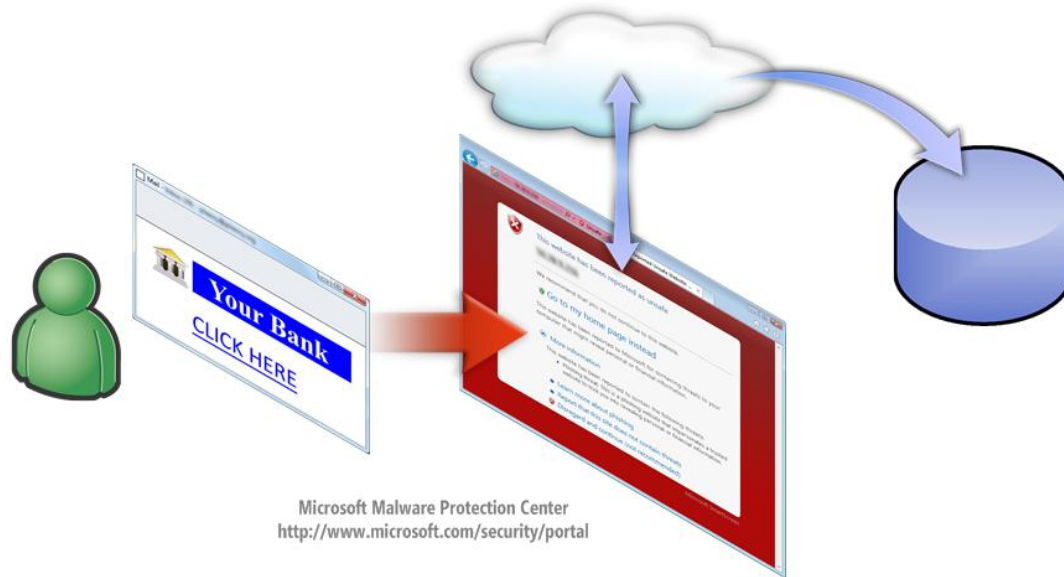
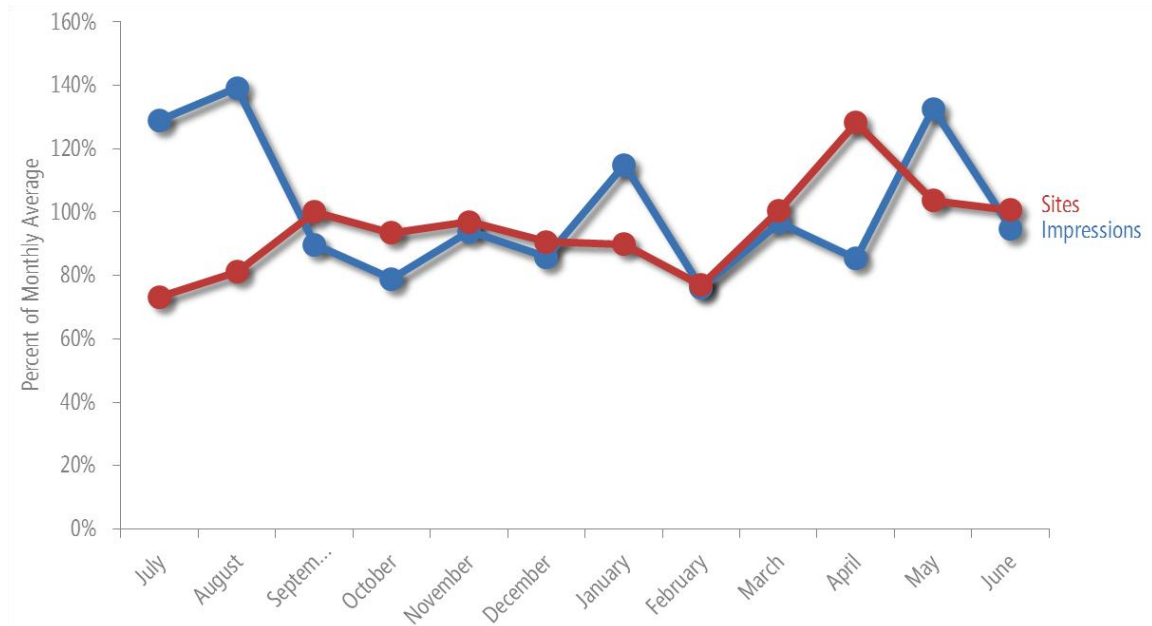


Figure 50 compares the volume of active phishing sites in the Microsoft URL Reputation Service database each month with the volume of phishing impressions tracked by Internet Explorer.

Figure 50. Phishing sites and impressions tracked each month from July 2010 to June 2011 relative to the monthly average for each



- Following a large spike in impressions in June 2010, the figures for both sites and impressions have been mostly stable over the past 12 months. Most phishing sites only last a few days, and attackers create new ones to replace older ones as they are taken offline, so the list of known phishing sites is prone to constant change without significantly affecting overall volume.
- Phishing impressions and active phishing pages rarely correlate strongly with each other. Phishers often engage in discrete campaigns intended to drive more traffic to each phishing page, without necessarily increasing the total number of active phishing pages they maintain at the same time. In August 2010, the month with the highest number of impressions over the past year, the number of active phishing sites tracked was actually near its lowest level for the period.

Target Institutions

Figure 51 and Figure 52 show the percentage of phishing impressions and active phishing sites, respectively, recorded by Microsoft during each month in 1H11 for the most frequently targeted types of institutions.

Figure 51. Impressions for each type of phishing site each month in 1H11, as reported by SmartScreen Filter

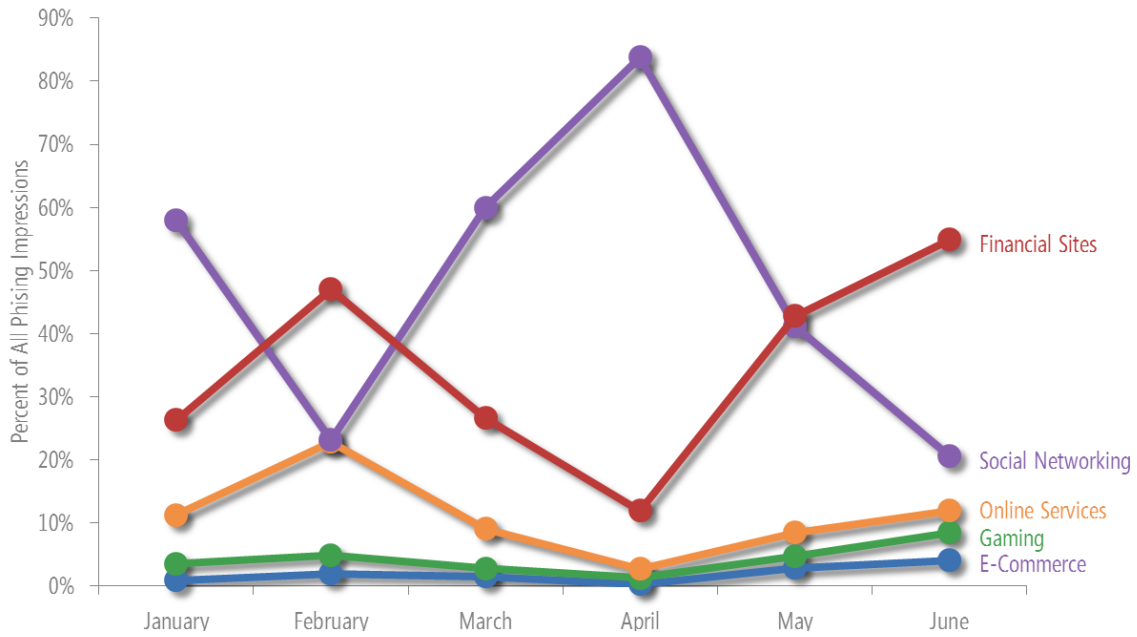
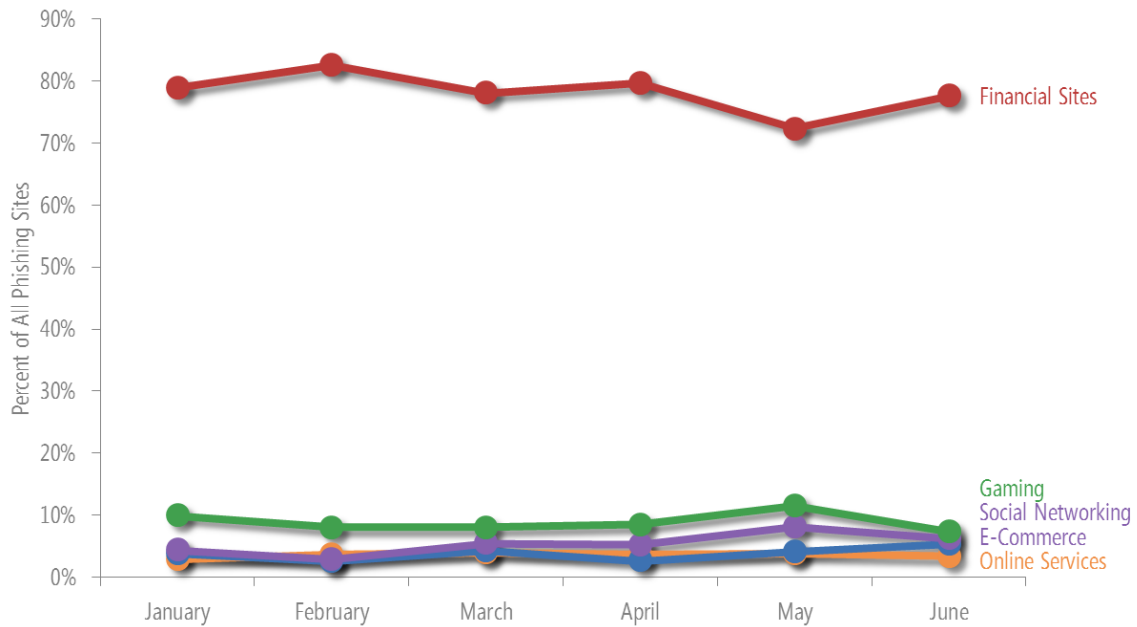


Figure 52. Active phishing sites tracked each month in 1H11, by type of target



- Phishers have traditionally targeted financial sites more than other types of sites, but the largest share of phishing impressions in 1H11 was for sites that

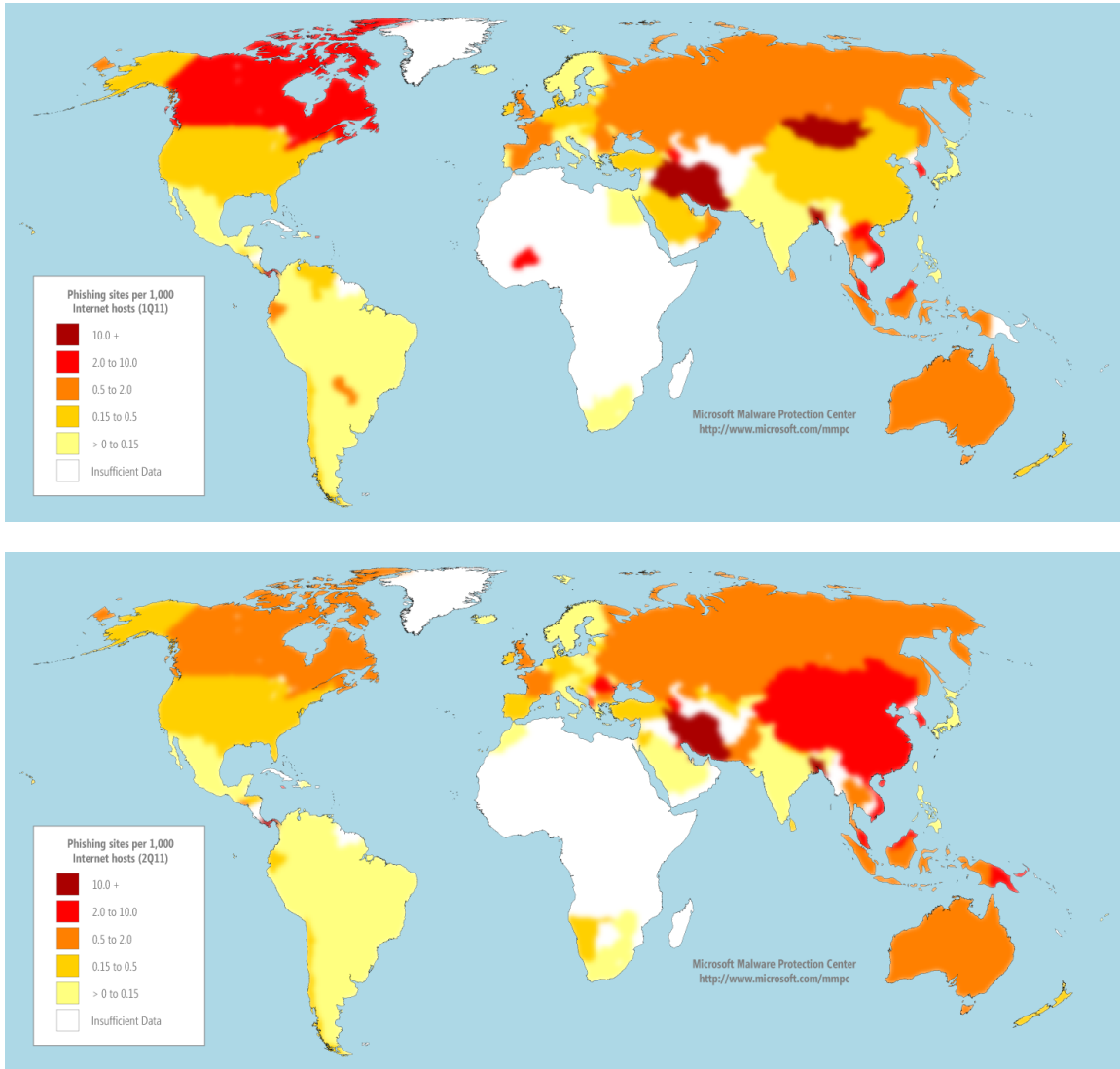
targeted social networks, reaching a high of 83.8 percent of impressions in April. Overall, impressions that targeted social networks accounted for 47.8 percent of all impressions in 1H11, followed by those that targeted financial institutions at 35.0 percent.

- By contrast, phishing sites that targeted financial institutions accounted for an average of 78.3 percent of active phishing sites tracked each month in 1H11, compared to just 5.4 percent for social networks. Financial institutions targeted by phishers can number in the hundreds, and customized phishing approaches are required for each one. The number of popular social networking sites is much smaller, so phishers who target social networks can effectively target many more people per site. Still, the potential for direct illicit access to victims' bank accounts means that financial institutions remain perennially popular phishing targets, and they continue to receive the largest or second-largest number of impressions each month.
- This phenomenon also occurs on a smaller scale with online services and gaming sites. A small number of online services account for the majority of traffic to such sites, so phishing sites that targeted online services garnered 11.0 percent of impressions with just 3.6 percent of sites. Online gaming traffic tends to be spread out among a larger number of sites, so phishing sites that targeted online gaming destinations accounted for 8.9 percent of active sites but gained just 4.3 percent of impressions.
- Phishing sites that targeted e-commerce were responsible for just 3.8 percent of active sites and 1.9 percent of impressions, suggesting that phishers have not found e-commerce sites to be especially profitable targets.

Global Distribution of Phishing Sites

Phishing sites are hosted all over the world on free hosting sites, on compromised web servers, and in numerous other contexts. Performing geographic lookups of IP addresses in the database of reported phishing sites makes it possible to create maps that show the geographic distribution of sites and to analyze patterns.

Figure 53. Phishing sites per 1,000 Internet hosts for locations around the world in 1Q11 (top) and 2Q11 (bottom)



- Locations with smaller populations and fewer Internet hosts tend to have higher concentrations of phishing sites, although in absolute terms most phishing sites are located in large, industrialized countries/regions with large numbers of Internet hosts.
- The worldwide distribution of phishing sites remained largely consistent between the first and second quarters. Exceptions include China, which increased from 0.35 phishing sites per 1000 hosts in 1Q11 to 2.54 in 2Q11; Canada, which decreased from 2.05 to 1.02; and France, which decreased from 1.34 to 0.81.

Malware Hosting Sites

SmartScreen Filter in Internet Explorer 8 and 9 helps provide protection against sites that are known to host malware, in addition to phishing sites. SmartScreen Filter uses URL reputation data and Microsoft antimalware technologies to determine whether those servers distribute unsafe content. As with phishing sites, Microsoft keeps track of how many people visit each malware hosting site and uses the information to improve SmartScreen Filter and to better combat malware distribution.

Figure 54. SmartScreen Filter in Internet Explorer 8 (top) and Internet Explorer 9 (bottom) displays a warning when a user attempts to download an unsafe file

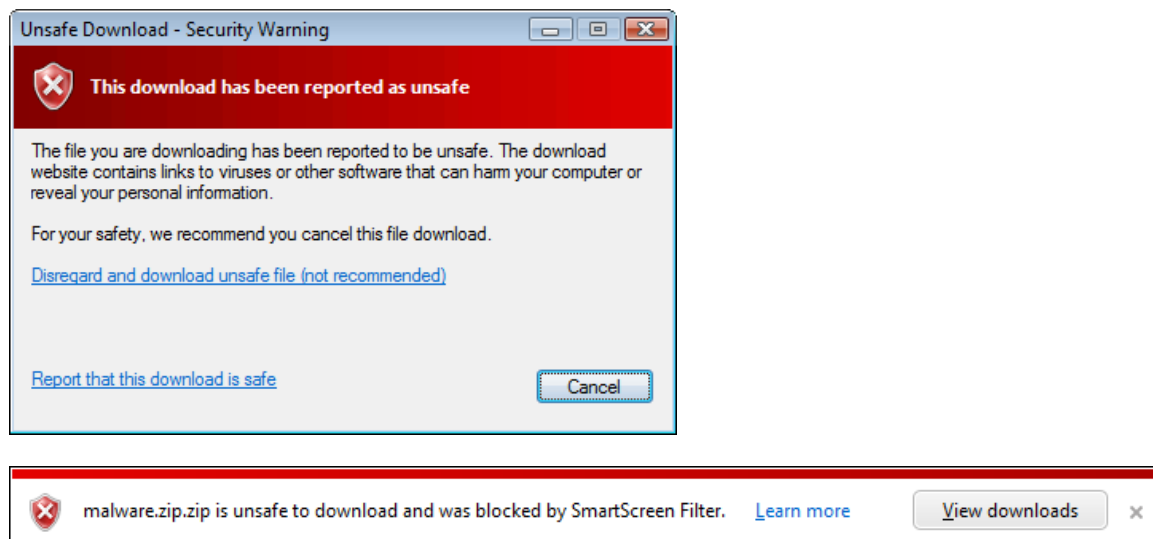
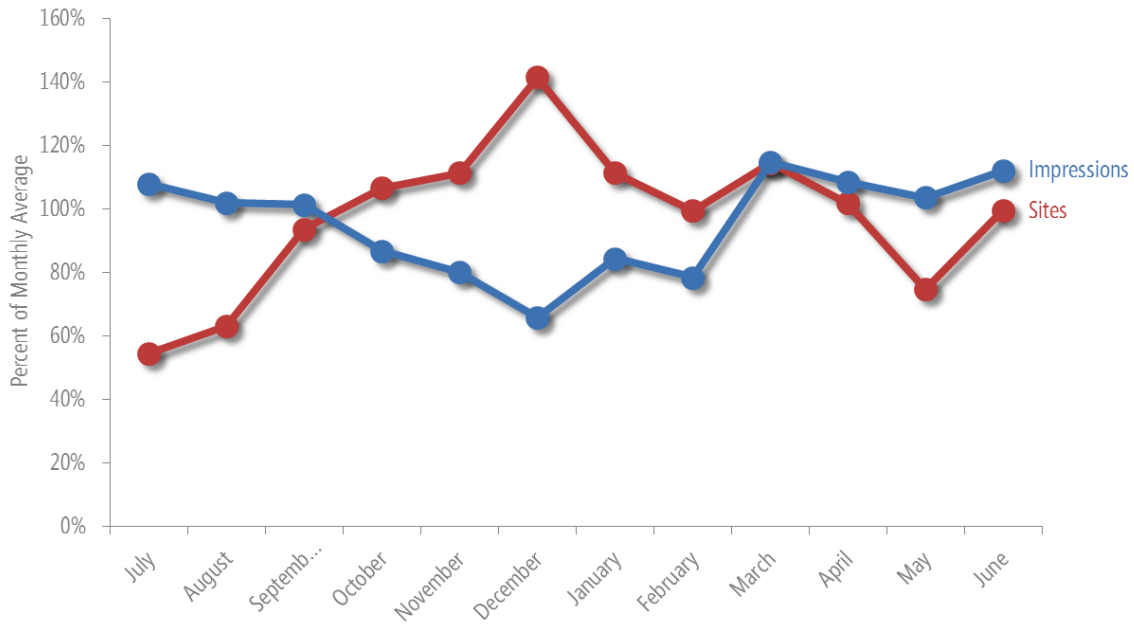


Figure 55 compares the volume of active malware hosting sites in the Microsoft URL Reputation Service database each month with the volume of malware impressions tracked by Internet Explorer.

Figure 55. Malware hosting sites and impressions tracked each month from July 2010 to June 2011, relative to the monthly average for each



- As with phishing, malware hosting impressions and active sites rarely correlate strongly with each other, and months with high numbers of sites and low numbers of impressions (or vice versa) are not uncommon.

Malware Categories

Figure 56 and Figure 57 show the types of threats hosted at URLs that were blocked by SmartScreen Filter in 1H11.

Figure 56. Threats hosted at URLs blocked by SmartScreen Filter in 1Q11 and 2Q11, by category

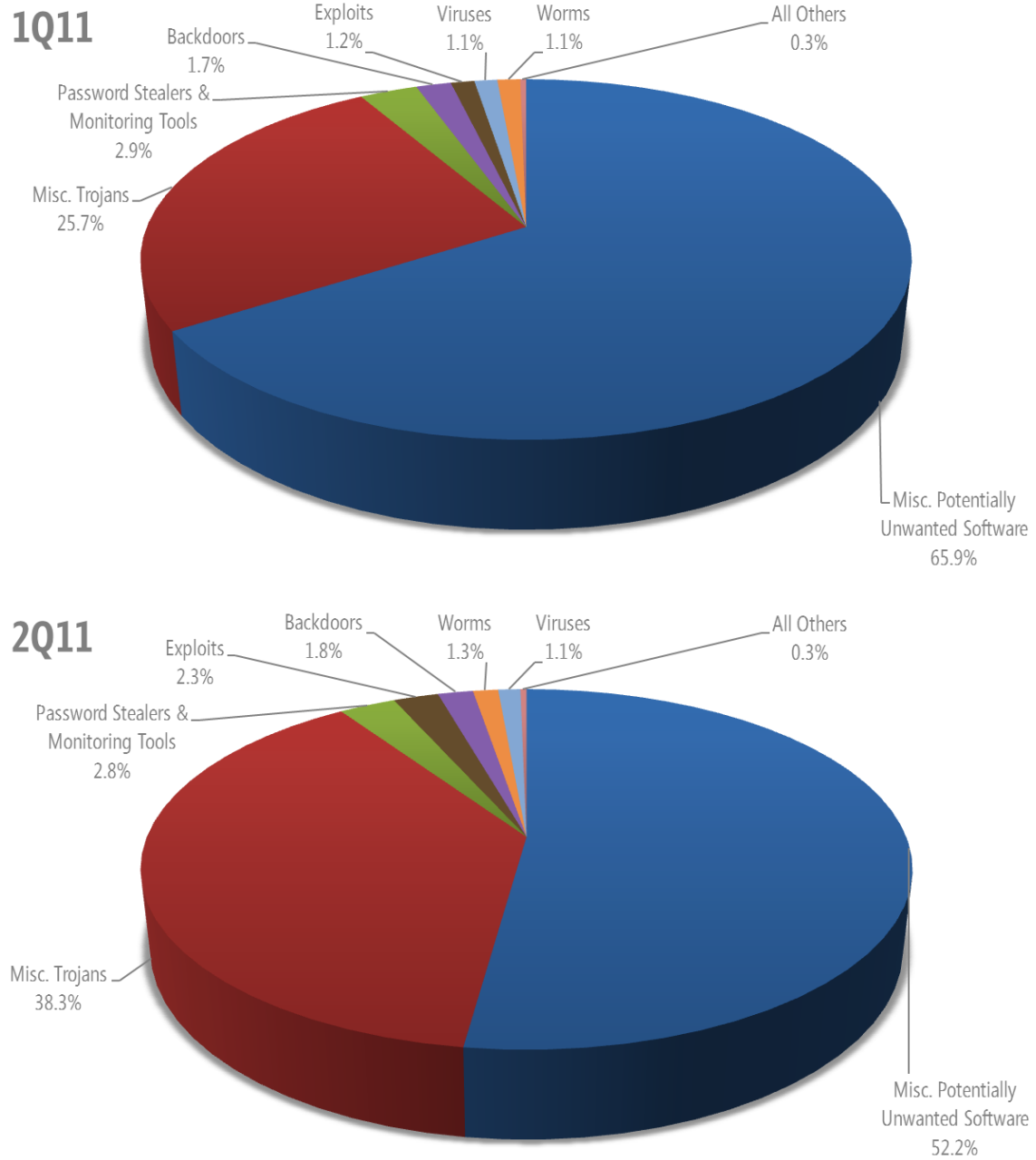


Figure 57. The top 10 malware families hosted on sites blocked by SmartScreen Filter in 1Q11 and 2Q11, by percent of all such sites

1Q11 Rank	Threat Name	Category	Percent	2Q11 Rank	Threat Name	Category	Percent
1	Win32/MoneyTree	Misc. Potentially Unwanted Software	45.8%	1	Win32/MoneyTree	Misc. Potentially Unwanted Software	38.8%
2	Win32/Obfuscator	Misc. Potentially Unwanted Software	6.3%	2	VBS/Startpage	Misc. Trojans	15.7%
3	Win32/Begseabug	Trojan Downloaders & Droppers	4.7%	3	Win32/Obfuscator	Misc. Potentially Unwanted Software	5.2%
4	VBS/Startpage	Misc. Trojans	4.7%	4	Win32/Bancos	Password Stealers & Monitoring Tools	2.3%
5	Win32/Delf	Trojan Downloaders & Droppers	2.6%	5	Win32/Small	Trojan Downloaders & Droppers	2.3%
6	Win32/Bancos	Password Stealers & Monitoring Tools	1.8%	6	Win32/Meredrop	Misc. Trojans	2.2%
7	Win32/VB	Worms	1.7%	7	Win32/VB	Worms	1.9%
8	Win32/Banload	Trojan Downloaders & Droppers	1.7%	8	Win32/Microjoin	Trojan Downloaders & Droppers	1.7%
9	Win32/Microjoin	Trojan Downloaders & Droppers	1.3%	9	Win32/Dynamer	Misc. Trojans	1.3%
10	Win32/GameHack	Misc. Trojans	1.0%	10	Win32/FakeRean	Misc. Trojans	1.0%

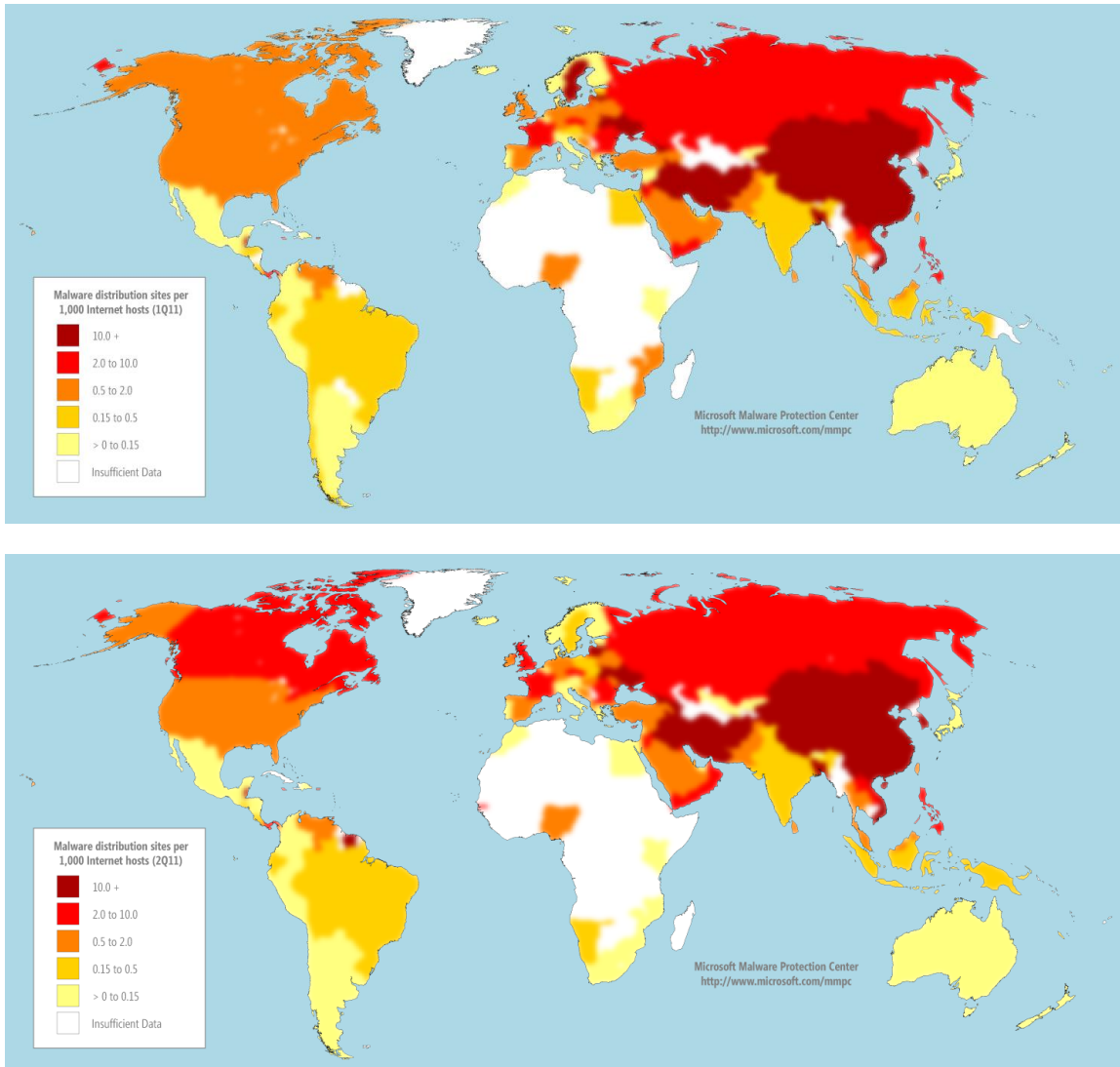
- Overall, sites that hosted the top 10 families constituted 71.6 percent of all impressions in the first quarter of 2011 and 72.3 percent in the second quarter.
- Miscellaneous Potentially Unwanted Software accounted for most impressions in both quarters, primarily because of [Win32/MoneyTree](#). MoneyTree has consistently been the family responsible for the greatest number of impressions since 2009.

- Miscellaneous Trojans increased from 25.7 percent of impressions in 1Q11 to 38.3 percent in 2Q11, primarily because of increased impressions for [VBS/Startpage](#), a generic detection for a range of threats that attempt to change the user's Internet Explorer home page.
- [Win32/Begseabug](#), the third most prevalent family in 1Q11, is a trojan that downloads and executes arbitrary files on an affected computer.
- [Win32/Bancos](#) and [Win32/Banload](#) are related families that target users' online banking credentials, usually involving Brazilian banks.
- [Win32/Obfuscator](#), [Win32/Delf](#), [Win32/Small](#), [Win32/VB](#), [Win32/Meredrop](#), [Win32/Microjoin](#), and [Win32/Dynamer](#) are all generic detections for collections of unrelated threats that share certain identifiable characteristics.

Global Distribution of Malware Hosting Sites

Figure 58 shows the geographic distribution of malware hosting sites reported to Microsoft in 1H11.

Figure 58. Malware distribution sites per 1,000 Internet hosts for locations around the world in 1Q11 (top) and 2Q11 (bottom)



- As with phishing sites, the worldwide distribution of malware hosting sites was largely stable between the first and second quarters. Exceptions include Sweden, which decreased from 22.48 malware hosting sites per 1000 hosts in 1Q11 to 0.15 in 2Q11; Israel, which decreased from 23.84 to 0.63; and China, which decreased from 34.64 to 23.70.

Drive-By Download Sites

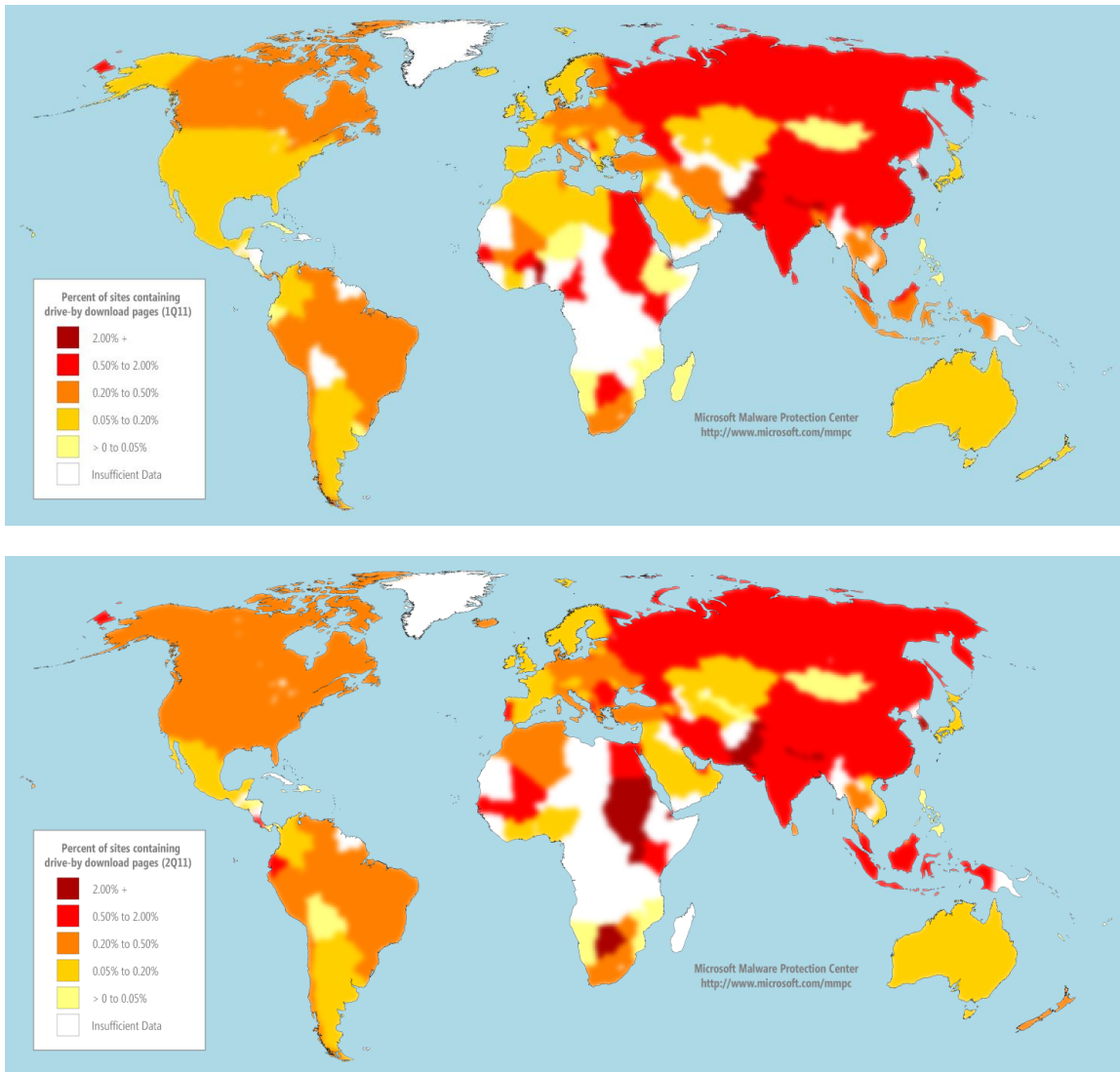
A *drive-by download* site is a website that hosts one or more exploits that target vulnerabilities in web browsers and browser add-ons. Users with vulnerable computers can be infected with malware simply by visiting such a website, even without attempting to download anything.

Search engines such as Microsoft Bing® have taken a number of measures to help protect users from drive-by downloads. Bing analyzes websites for exploits as they are indexed and displays warning messages when listings for drive-by download pages appear in the list of search results. (See [Drive-By Download Sites](#) at the *Microsoft Security Intelligence Report* website for more information about how drive-by downloads work and the steps Bing takes to protect users from them.)

The information in this section was generated from an analysis of the drive-by download URLs in the Bing index in 1H11.

In previous volumes of the *Microsoft Security Intelligence Report*, drive-by statistics were presented as the percentage of websites in each country-code top-level domain (ccTLD) that host drive-by download pages. To provide a more accurate perspective on the drive-by download landscape, the current volume presents these statistics as the number of individual drive-by pages in each country or region, determined by IP geolocation, as a percentage of the total number of URLs in each. This perspective incorporates two significant changes: individual URLs are used instead of domains, and IP address is used to determine country or region instead of ccTLD. For these reasons, the statistics presented here should not be directly compared to findings in previous volumes of the *Microsoft Security Intelligence Report*.

Figure 59. Drive-by download pages in 1Q11 (top) and 2Q11 (bottom), by percentage of all URLs in each country/region



- In 1H11, about 0.25 percent of the URLs in the Bing index were compromised by drive-by download exploit code.
- Among the locations with large numbers of URLs in the index, the locations with the most pages hosting drive-by download exploit code included Korea (2.77 percent of all pages in 2Q11), China (0.8 percent), and Romania (0.66 percent).
- The locations with the greatest increases from 1Q11 to 2Q11 included Romania, which increased from 0.18 percent of pages infected to 0.66

percent; Ireland, which increased from 0.08 percent to 0.19 percent; and the United States, which increased from 0.14 percent to 0.22 percent.

- The locations with the lowest percentage of malicious or compromised pages included Japan (0.06 percent of all pages in 2Q11), Austria (0.1 percent), and Australia (0.1 percent).
- The locations with the greatest decreases from 1Q11 to 2Q11 included Sweden, which decreased from 0.12 percent of pages infected to 0.07 percent; Denmark, which decreased from 0.35 percent to 0.24 percent; Vietnam, which decreased from 0.21 percent to 0.19 percent.

Guidance: Protecting Users from Unsafe Websites

Organizations can best protect their users from malicious and compromised websites by mandating the use of web browsers with appropriate protection features built in and by promoting safe browsing practices. For in-depth guidance, see the following resources in the “Managing Risk” section of the *Microsoft Security Intelligence Report* website:

- [Promoting Safe Browsing](#)
- [Protecting Your People](#)



Managing Risk



Protecting Organizations, Software, and People

Addressing threats and risks requires a concerted effort on the part of people, organizations, and governments around the world. The “[Managing Risk](#)” section of the *Microsoft Security Intelligence Report* website presents a number of suggestions for preventing harmful actions from malware, breaches, and other security threats and for detecting and mitigating problems when they occur:

- “[Protecting Your Organization](#)” offers guidance for IT administrators in small, medium-sized, and large companies seeking to improve their security practices and to stay up to date on the latest developments.
- For software developers, “[Protecting Your Software](#)” offers information about developing secure software, including in-house software, and securing Internet-facing systems from attack.
- “[Protecting Your People](#)” offers guidance for promoting awareness of security threats and safe Internet usage habits within an organization.

In addition, this volume of the report provides some additional guidance for IT and security professionals interested in increasing the level of protection they are able to provide in specific areas:

- “Advice to IT Professionals on Social Engineering,” beginning on page 25, explores some of the technical and policy measures IT departments can take to guard against social engineering attacks.
- “Advanced Malware Cleaning Techniques for the IT Professional,” beginning on page 96, gives some in-depth information about using Microsoft Sysinternals tools to investigate and remove malware.
- “Promoting Safe Browsing,” beginning on page 113, explores some of the security features built into Windows Internet Explorer and describes how users and administrators can take advantage of them to create a safer Internet browsing experience.

Advanced Malware Cleaning Techniques for the IT Professional

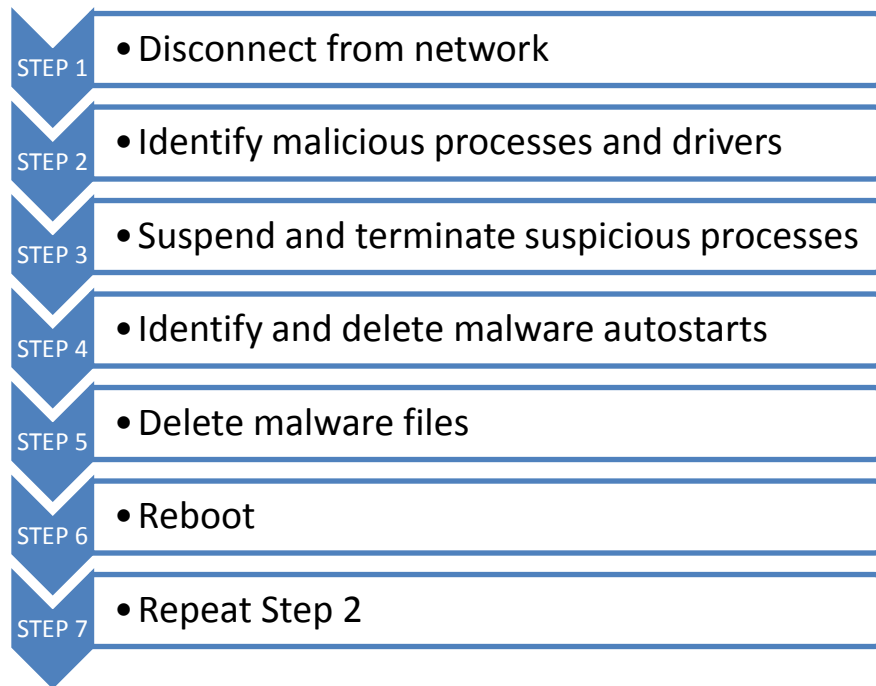
Mark Russinovich
Microsoft Technical Fellow

This section of the *Microsoft Security Intelligence Report* provides information and guidance for IT professionals about investigating, analyzing, and—when possible—removing malware from an infected computer.

Except in special situations, Microsoft recommends the use of antimalware software tools, such as Microsoft Forefront Endpoint Protection (for organizations) and Microsoft Security Essentials (for individuals), for keeping computers free from malware, rather than the manual techniques described in this section. This guidance is intended for advanced users who possess a good understanding of the inner workings of computers and Windows, and who wish to understand the disinfection process—how malware can be removed without the aid of antimalware software. It is designed to help IT professionals understand the impact of malware, understand how malware operates, learn how to use some specific software tools, and create a rudimentary roadmap for cleaning infected computers in special situations.

This guidance involves the use of several Windows Sysinternals tools. Sysinternals is a suite of advanced diagnostics and troubleshooting utilities for the Windows platform that is available for download at no charge from the Microsoft Download Center. See technet.microsoft.com/sysinternals for more information about the Sysinternals utilities.

Figure 60. A seven-step process for removing malware



Step 1: Disconnect from the Network

Disconnecting the infected computer or computers from the network is an essential part of the malware removal process, because it ensures that infected computers do not spread malware to other computers on the network. This step can be performed by physically disconnecting or disabling the network cable or card from each computer (including disabling wireless networking via hardware switch if possible), or by disabling all networking functions from the BIOS configuration screen (instructions for performing this task vary for different computers and motherboards).

Step 2: Identify Malicious Processes and Drivers

After an infected computer is disconnected from the network, the next step in the disinfection process is to identify any malicious processes. This step involves looking for telltale signs such as:

- Processes without custom icons.

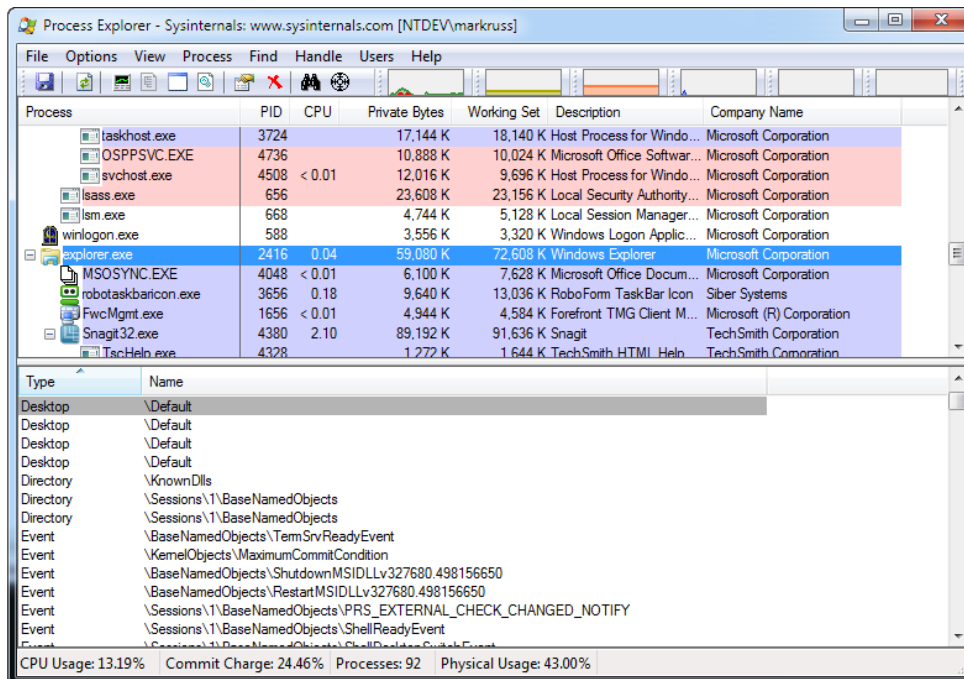
- Processes that have no description or company name associated with them.
- Files that represent themselves as being from Microsoft, but don't have digital signatures.
- Unfamiliar processes running from the Windows directory.
- Files that are *packed*, which means that they have been compressed or encrypted. Most malware files are packed by their distributors in an effort to make them more difficult for security software to identify.
- Strange URLs in strings embedded in files.
- Processes with open TCP/IP endpoints.
- Processes that host suspicious dynamic-link libraries (DLLs) or services.

By themselves, these signs do not conclusively indicate a malicious process. For example, many legitimate executables and other files are packed, and many legitimate processes run without custom icons. Also, not all malware files and processes exhibit all the signs listed here. However, these signs generally serve as useful clues for detecting malware on an infected computer. A Sysinternals tool called [Process Explorer](#) can help a troubleshooter spot malicious processes.

Using Process Explorer

Process Explorer is a kind of “super Task Manager” that provides a variety of general troubleshooting capabilities, including the discovery of DLL versioning problems, handle leaks, and locked file information; performance troubleshooting; and detailing hung processes.

Figure 61. The Process Explorer main window



The Process Explorer main window provides a simple paneled display of information about the processes that are running on the computer. Although there are superficial similarities between this view and the **Processes** tab in Windows Task Manager, Process Explorer provides a great deal more information about each process. Each row in the process list represents a process object running on the computer that has its own virtual address space and one or more threads that could conceivably execute code at some point.

The names of malicious processes often mimic the names of legitimate processes, which can make them difficult to identify in Task Manager. Using Process Explorer makes it easier to identify processes that run from suspicious locations, or that display suspicious characteristics. By default, processes are listed in a hierarchical view called the process tree, which shows parent/child relationships between processes. Columns display a range of properties for each process, including the name of the company that published the image, a brief description, version information, and more.

When investigating an infection, pay attention to the **Company Name**, **Description**, and **Version** columns. Legitimate software publishers usually provide values for some or all of these columns, but malware authors sometimes

neglect them. To display more columns or hide columns already in the display, click the **View** menu, and then click **Select Columns**.

Rows can be highlighted in different colors, which provides additional information:

- Blue indicates that the process is running in the same security context as Process Explorer. Generally, this means that it's running under the active user account, rather than a system or service account.
- Pink indicates that the process is hosting one or more Windows services. Services can run on their own, or as part of the services DLL inside a Svchost.exe process.
- Purple indicates that the image has been packed (compressed or encrypted).
- Green and red indicates that the process has just started or exited, respectively. By default, rows are only highlighted green or red for 1 second, which can make them difficult to track. You can change this default length by clicking **Difference Highlight Duration** in the **Options** menu.

Other colors indicate different process types, but the ones in the preceding list are the important ones that can help you locate and remove malware.

Moving the mouse pointer over a row displays a tooltip with information about the process, such as the full path to the process image, which can help you identify processes running from unusual or suspicious locations. Tooltips also provide additional information for system processes, such as DLLs hosted by Rundll32.exe, services hosted by Svchost.exe and other service processes, and COM server information for Dllhost.exe. Malware often attempts to disguise its presence by attaching itself to system processes such as these, so pay attention to tooltips when investigating the source of an infection.

Figure 62. Tooltips provide additional information about processes

svchost.exe	300	< 0.01	Host Process for Windows S...	Microsoft Corporation	n/a
WUDFHost.exe	1176				n/a
WUDFHost.exe	3520				n/a
dwm.exe	2460	0.37	Desktop Window Manager	Microsoft Corporation	DEP
svchost.exe	320	0.03	Host Process for Windows S...	Microsoft Corporation	n/a
svchost.exe	1088	< 0.01	Host Process for Windows S...	Microsoft Corporation	n/a
svchost.exe	1296	0.03	Host Process for Windows S...	Microsoft Corporation	n/a
spoolsv.exe				Microsoft Corporation	n/a
svchost.exe			Command Line: C:\Windows\system32\svchost.exe -k LocalService	Microsoft Corporation	n/a
svchost.exe			Path: C:\Windows\System32\svchost.exe (LocalService)	Microsoft Corporation	n/a
svchost.exe			Services: Amazon.com	Microsoft Corporation	n/a
svchost.exe			COM+ Event System [EventSystem]	Microsoft Corporation	n/a
svchost.exe			Diagnostic Service Host [WdiServiceHost]	Microsoft Corporation	n/a
mDNSRcvr.exe			Function Discovery Provider Host [fdPHost]	Microsoft Corporation	n/a
svchost.exe			Network List Service [netprofm]	Microsoft Corporation	n/a
svchost.exe			Network Store Interface Service [nsi]	Microsoft Corporation	n/a

To research a process you don't recognize, select **Search Online** from the **Process** menu or press Ctrl+M to search for the process name using the configured browser and search engine. Malware sometimes uses random or semi-random strings for process and file names, so even if you can't locate affirmative evidence that a process is a malicious one, a search that produces no results at all for a process name can sometimes indicate that the process is suspicious.

Figure 63 shows a malicious process created by a variant of the worm family Win32/Rimecud. This process has no icon, company name, or description, and a name that produces no results in an Internet search.

Figure 63. A malicious process in Process Explorer

cmd.exe	1556		Windows Command Processor	Microsoft Corporation
explorer.exe	3268		Windows Explorer	Microsoft Corporation
ctfmon.exe	3660		CTF Loader	Microsoft Corporation
rime0000.exe	3292	< 0.01		

DLL View

Malware can hide inside a legitimate process as a DLL, using a technique called DLL injection. Process Explorer's lower pane (which can be displayed by clicking the **Show Lower Pane** button on the toolbar or pressing Ctrl+L) lets you list the contents of the process selected in the upper pane. The lower pane can be configured to display in either DLL view or Handle view. DLL view lists all the DLLs and other files mapped into the process' address space, and Handle view lists all the kernel objects opened by the process. Pressing Ctrl+D opens DLL view.

Figure 64. DLL view lists the DLLs and other files used by a process

wininit.exe	600	1,440 K	3,716 K	
services.exe	648	14,520 K	15,332 K	
svchost.exe	840	4,768 K	8,856 K	Host Process for Windows Services
mobsync.exe	4636	< 0.01	2,720 K	10,832 K Microsoft Sync Center
wlcomm.exe	4952	0.05	17,160 K	24,704 K Windows Live Communications Platform
UcMapi.exe	6516	0.03	245,768 K	257,756 K Microsoft Lync 2010 MAPI COM Server
dllhost.exe	1696	2,436 K	7,268 K	

Name	Description	Company Name	Version
abssm.dll	Windows Live Contacts Synchroniz...	Microsoft Corporation	15.4.3538.513
advapi32.dll	Advanced Windows 32 Base API	Microsoft Corporation	6.1.7601.17514
apisetschema.dll	ApiSet Schema DLL	Microsoft Corporation	6.1.7600.16385
apphelp.dll	Application Compatibility Client Libr...	Microsoft Corporation	6.1.7601.17514
bcrypt.dll	Windows Cryptographic Primitives ...	Microsoft Corporation	6.1.7600.16385
bcryptprimitives.dll	Windows Cryptographic Primitives ...	Microsoft Corporation	6.1.7600.16385
cabinet.dll	Microsoft® Cabinet File API	Microsoft Corporation	6.1.7601.17514

In DLL view, each row in the lower pane lists information about a DLL, executable file, or other memory-mapped file that is being used by the process. For the System process, DLL view lists the image files mapped into kernel memory, including Ntoskrnl.exe and all the loaded device drivers. As with processes, any packed files are highlighted in purple.

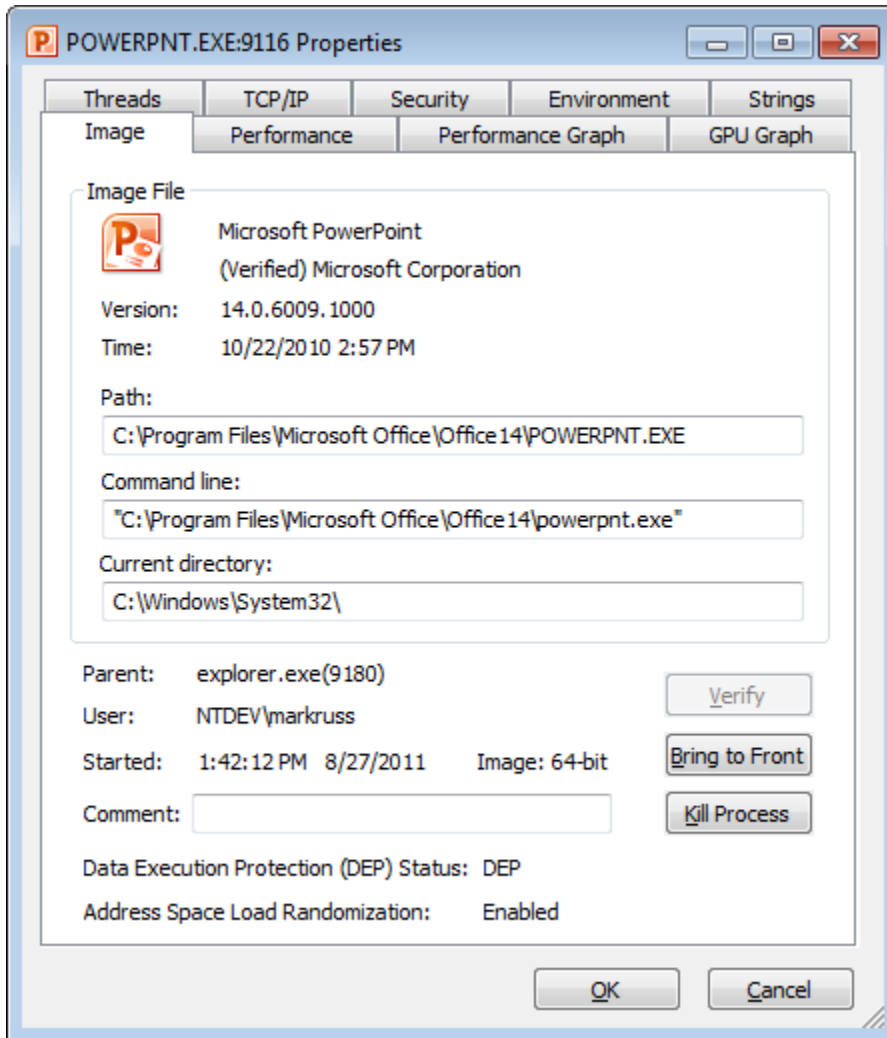
Double-clicking a row displays a **Properties** dialog with information about the file, including any strings found in the file on disk and in memory (see page 104). DLL view also supports the same Search Online functionality that the Process view does.

DLL view is empty for the System Idle Process and Interrupts pseudo-processes. You need to run Process Explorer with administrative rights to list DLLs loaded in processes running as a different user, but administrative rights are not required to list the images loaded in the System process.

Process Properties

Double-clicking a process launches the **Properties** dialog, which is shown in Figure 65.

Figure 65. The Properties dialog



This dialog provides detailed process information, much of which can be useful when investigating malware. Process information is arranged on a number of tabs, including:

- **Image.** This tab displays information about the executable file that launched the process, including the path to the file, the command-line argument used to launch it, the user account under which it is running, the creation time of the file, and the time the process was started.
- **Services.** This tab provides detailed information about the services registered in the process. This information includes the name used to

identify the service in the registry, the display name of the service, an optional description, and (for Svchost.exe DLLs) the DLL path.

- **Strings.** This tab lists any Unicode strings found in the executable file. Look for suspicious URLs, names, or debug strings—malware binaries are often “signed” by their creators, or include URLs for command-and-control (C&C) or download servers. Process Explorer allows you to view strings in the file’s address space in memory as well as on disk, which can be helpful in the investigation of packed files. (*Strings.exe*, another Sysinternals utility, provides a command-line interface for extracting strings from a file.) Clicking the **Memory** option button causes Process Explorer to list the strings visible in the file’s memory mapping, which can reveal strings that might be encrypted in the on-disk version of the file.

Image Verification

A malware author who takes the trouble to do so can easily add the name of a legitimate company, such as Microsoft, to the Company field of an executable file. Therefore, to provide assurance that their products are genuine, legitimate software vendors digitally sign most of the program files they publish. A digital signature can be used to verify that a file has been signed by the vendor using a private key and that the file has not been modified since being signed.

Process Explorer allows you to automatically verify the signature of a signed executable or DLL file. By default, verification is performed only on demand, and can be performed for individual files or for all running processes. In the **Properties** dialog for both processes and DLLs, the **Image** tab contains a **Verify** button that can be used to verify the digital signature for the associated file. Clicking the button causes Process Explorer to check the Certificate Revocation List (CRL) for the certificate to ensure that it is valid, and to check the cryptographic hash of the file to verify that it has not been tampered with since being signed. (Validating certificates requires reconnecting the computer to the Internet, which should only be considered if the risk of additional exfiltration or infection is low.)




To configure Process Explorer to automatically verify the signatures for all running processes and files, click the **Options** menu, and then click **Verify Image Signatures**.

The Verified Signer field, which displays next to the file icon in the **Properties** dialog and as a column that can be shown in the process list and DLL View, indicates the status of any signature check that has been performed. If Process

Explorer is able to verify the signature, the field displays “(Verified)”, followed by the subject name from the certificate. (Note that the name on the signing certificate might not be the same as the name in the Company Name field. For example, most executable files that ship as part of Windows display “Microsoft Corporation” as the company name but are signed with a “Microsoft Windows” certificate.)

If signature verification has not been attempted, or if the selected file is not an executable file type, the field is blank or displays “(Not verified)” followed by the company name from the file’s version resource. “(Unable to verify)” followed by the company name indicates that the file is not signed or that a signature check has failed. You can also use the command-line Sysinternals [Sigcheck](#) tool to verify signatures on specific files as well as view detailed version information and their MD5, SHA1, and SHA256 hashes.

Figure 66. Autorun.A, masquerading as a system process but failing signature verification

 pmon.exe	3736	Process Monitor	Sysinternals - www.sysinter... (Verified) Sysinternals
 pexp.exe	916	2.44 Sysinternals Process Explorer	Sysinternals - www.sysinter... (Verified) Sysinternals
 smss.exe	2840	1.08 ?????????? ????? ? ?????...	Microsoft Corporation (Unable to verify) Microsoft Corporation

Investigating Loaded Drivers

Some malicious files are designed to load as device drivers, so it’s important to investigate drivers as well. Click the **System** row in the process list to display all the currently loaded drivers in DLL View. From this display, you can inspect the same properties that are available for DLLs and other files, such as the path to the driver file, the verified signer, strings found in the file on disk or in memory, and so on.

When investigating a 64-bit installation of Windows, note that two drivers, Hal.dll and Ntoskrnl.exe, are highlighted in purple, the color used to indicate packed files. These two files are actually not packed, but they exhibit some of the characteristics Process Explorer uses to classify files as compressed or encrypted. By itself, the fact that these two drivers are highlighted should not be considered evidence of infection.

In addition to Process Explorer, a number of utilities ship with Windows that can be used to provide different views of running processes:

- The System Information tool provides information about system drivers, including name, description, path and file name, driver type, and more. To run System Information:

- In Windows XP, click **Start**, click **Run**, type **msinfo32.exe**, and then press Enter.
- In Windows Vista, click **Start**, click in the **Start Search** box, type **msinfo32.exe**, and then press Enter.
- In Windows 7, click **Start**, click in the **Search programs and files** box, type **msinfo32.exe**, and then press Enter.

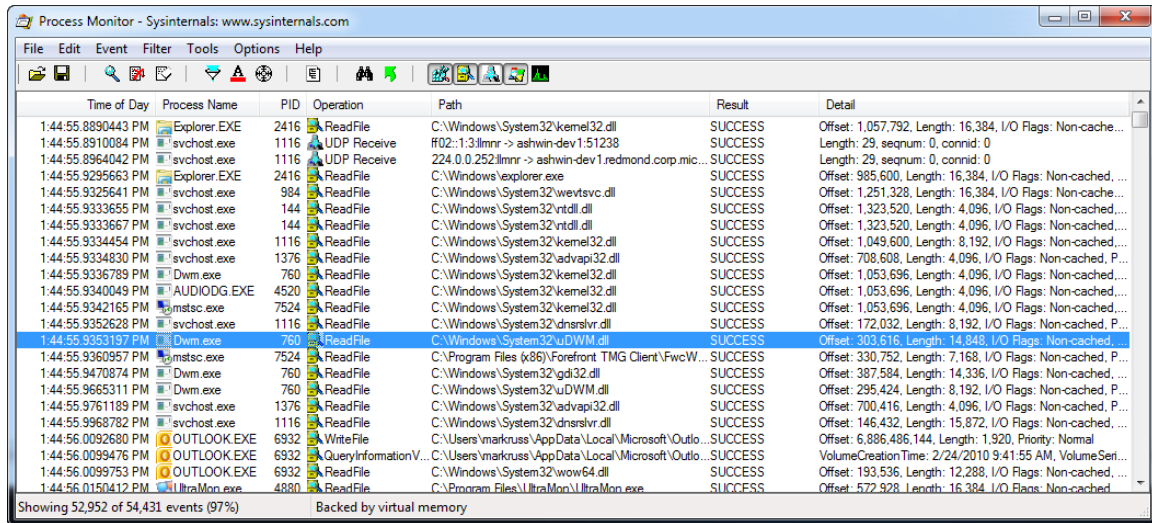
To display the list of system drivers, in the navigation pane, click **Software Environment**, and then click **System Drivers**.

- Sc.exe is a command line program used to communicate with the Service Control Manager and services. To display a list of drivers, at the command prompt type **sc query type= driver** and press Enter.
- In Device Manager, click the **View** menu, and then click **Show Hidden Devices** to display a list of devices that are normally hidden from view.

Tracing Malware

The list of active processes on a typical computer changes constantly, which can sometimes make it difficult to spot suspicious activity. In fact, if a malicious process starts and exits faster than Process Explorer's refresh rate, it may never show up in Process Explorer at all. You can use another Sysinternals tool, Process Monitor, to examine events in detail, including error messages and short-lived processes.

Figure 67. The Process Monitor main window



Process Monitor records many different kinds of activity as it runs; each row represents a specific event. Events tracked by Process Monitor include process starts and exits, thread starts and exits, network events, registry events, and many more. Each row gives a selection of information about the associated process, such as the operation performed, the path to the associated file or registry key, time information, and additional details.

To see short-lived processes in Process Monitor, open the Process Tree window by clicking the **Tools** menu and then clicking **Process Tree**, or by pressing Ctrl+I. The Process Tree window displays a list of all processes that have run since Process Monitor was launched, including processes that have exited.

Figure 68. The Process Tree View in Process Monitor shows details for current and exited processes

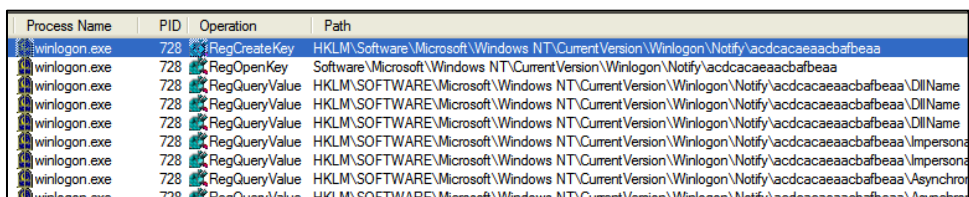
svchost.exe (3680)	Host Process for ...	C:\Windows\sys...	Microsoft Corporat...	NT AUTHORITY\...	C:\Windows\sys...	8/29/2011 8:33:54 AM	n/a
SearchIndexer.exe (4512)	Microsoft Window...	C:\Windows\sys...	Microsoft Corporat...	NT AUTHORITY\...	C:\Windows\sys...	8/29/2011 8:34:17 AM	n/a
SearchProtocolHost.exe	Microsoft Window...	C:\Windows\sys...	Microsoft Corporat...	NT AUTHORITY\...	C:\Windows\sys...	8/29/2011 8:39:07 AM	n/a
SearchFilterHost.exe (63)	Microsoft Window...	C:\Windows\sys...	Microsoft Corporat...	NT AUTHORITY\...	C:\Windows\sys...	8/29/2011 1:01:02 PM	8/29/2011 1:03:03 PM
SearchFilterHost.exe (52)	Microsoft Window...	C:\Windows\sys...	Microsoft Corporat...	NT AUTHORITY\...	C:\Windows\sys...	8/29/2011 1:03:16 PM	8/29/2011 1:05:16 PM
SearchFilterHost.exe (98)	Microsoft Window...	C:\Windows\sys...	Microsoft Corporat...	NT AUTHORITY\...	C:\Windows\sys...	8/29/2011 1:05:42 PM	8/29/2011 1:07:42 PM
svchost.exe (4900)	Host Process for ...	C:\Windows\Sys...	Microsoft Corporat...	NT AUTHORITY\...	C:\Windows\Sys...	8/29/2011 8:34:22 AM	n/a
wmpnetwk.exe (4936)	Windows Media P...	C:\Program Files\...	Microsoft Corporat...	NT AUTHORITY\...	C:\Program Files\...	8/29/2011 8:34:23 AM	n/a

Double-clicking a row displays a **Properties** dialog with all of the available information about the event, including the *call stack*—the hierarchical list of nested function calls that led to the event. By examining the call stack of a malicious event, you can determine which function directly invoked it, which may alert you to the presence of additional malware. You can integrate Process Monitor with [Debugging Tools for Windows](#), which are available for download at no

charge from the Microsoft Download Center, to make it easier to interpret the function calls in the stack.

Figure 69 shows events generated by a variant of the worm family [Win32/Swimmag](#), in the form of repeated queries of a registry key with a suspicious name. The DllName value of the suspicious key points to a malicious file in the system32 directory.

Figure 69. Malicious events in Process Monitor



Process Name	PID	Operation	Path
winlogon.exe	728	RegCreateKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\acdcacaeaacbfbeaa
winlogon.exe	728	RegOpenKey	Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\acdcacaeaacbfbeaa
winlogon.exe	728	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\acdcacaeaacbfbeaa\DllName
winlogon.exe	728	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\acdcacaeaacbfbeaa\DllName
winlogon.exe	728	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\acdcacaeaacbfbeaa\DllName
winlogon.exe	728	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\acdcacaeaacbfbeaa\Impersona
winlogon.exe	728	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\acdcacaeaacbfbeaa\Asynchro

For more information, visit the Process Monitor page at technet.microsoft.com/sysinternals/bb896645.

Step 3: Terminate Malicious Processes

After you locate the malicious processes, record the full path to each malicious file so you can remove them after terminating their processes.

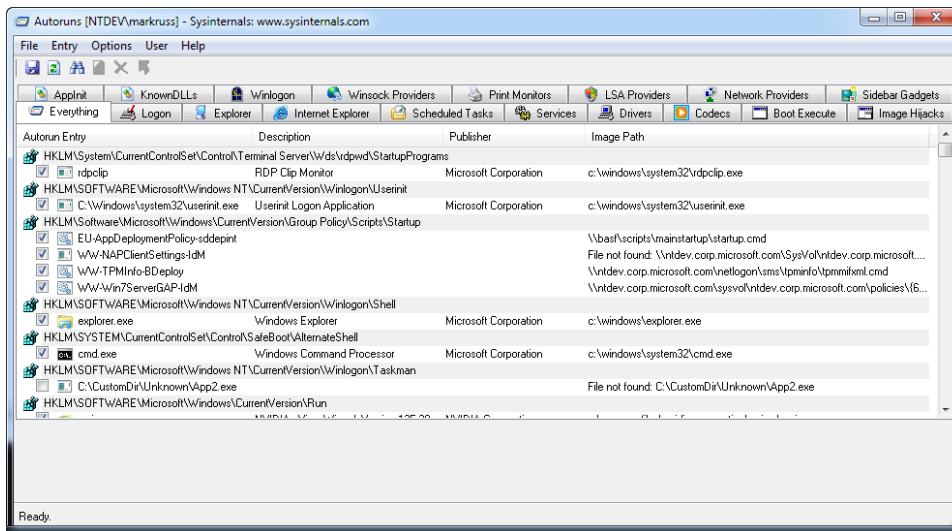
In an effort to resist removal, many malware infections include multiple processes, each of which monitors the others and restarts them when they are terminated. Instead of simply terminating malicious processes one by one, therefore, begin by suspending each process you've identified, and then terminate all of them. (Note that suspending Svchost.exe and other core system processes might cause parts of the system to become nonresponsive.) To suspend a process in Process Explorer, click the appropriate row in the process list, click the **Process** menu, and then click **Suspend**.

When terminating processes, watch for any newly started or restarted processes in the list (identified by green highlighting). If terminating malicious processes causes others to restart, it could be an indication that you're overlooking one or more sources of infection.

Step 4: Identify and Delete Malware Autostarts

Malware persists on an infected computer by configuring itself to run when Windows starts, or when a user logs in. The System Configuration utility (Msconfig.exe, sometimes called “Msconfig”) that ships with Windows displays a list of programs that load at startup, among other information. Although this utility can be useful for general troubleshooting purposes, Msconfig is often inadequate for dealing with a malware infection: it doesn’t check all of the *autostart extensibility points* (ASEPs), or the places that processes can automatically start from, and it doesn’t provide certain information that can be useful when investigating an infection. A better malware detection tool than Msconfig is another Sysinternals tool, [Autoruns](#).

Figure 70. Autoruns shows which programs run when Windows starts



Using Autoruns

When you launch Autoruns, it immediately begins filling its display with entries collected from known ASEPs. Each shaded row represents an ASEP location in either the file system or the registry. The rows beneath a shaded row indicate entries configured in that ASEP. Each row shows the item’s description, publisher, and path. Click a row to display more information about the item at the bottom of the Autoruns window, including file size, version number, and any command-line arguments used to launch the item. Double-clicking an item in the list displays the item in either Regedit or an Explorer window, depending on whether the item is a registry entry or a file on disk. For registry entries, you can also open the folder

that contains the file associated with the selected entry by clicking the **Entry** menu and then clicking **Jump to**.

On most computers, Autoruns is likely to display hundreds of entries for startup items. To reduce the number of items you have to investigate, enable the **Hide Microsoft and Windows Entries** and **Verify Code Signatures** items in the **Options** menu, and then click **Refresh** on the toolbar to filter out items with verified Microsoft signatures.

Autoruns can also be used to display autostart entries for other profiles, and for offline computers (for example, an offline virtual machine, or a physical computer booted into a preloader environment with Autoruns installed). To display entries for another profile, click the **User** menu, and then click the user account you want to check. To check an offline computer, click the **File** menu, and then click **Analyze Offline System**.

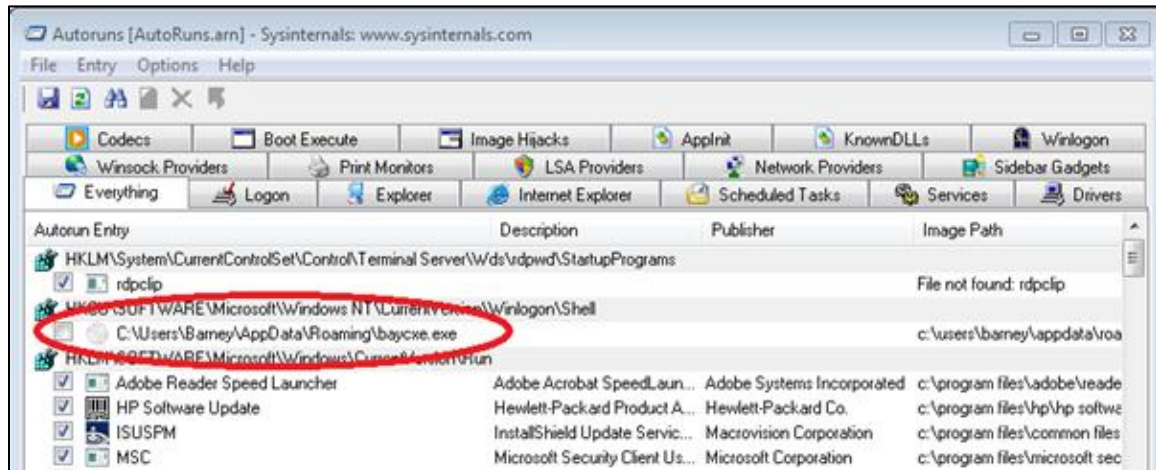
The Autoruns download package includes a command-line version of the tool, Autorunsc.exe. See technet.microsoft.com/sysinternals/bb963902 for usage instructions.

Identifying Malware Autostarts

Suspicious autostart items can often be identified by many of the same characteristics listed on page 97: look for files with no icon, entries with blank Description and Publisher fields, files with unusual or random-seeming names, files that can't be verified, and files in unexpected locations, among others. To quickly search for information about a filename online, click the **Entry** menu and then click **Search Online**, or press Ctrl+M.

Figure 71 shows a malicious autostart entry created by a variant of [Win32/FakePAV](#), a rogue security software program. This entry has blank Description and Publisher fields, has a random-seeming name with no obvious meaning, and comes from a location in the registry that usually points to Explorer.exe.

Figure 71. A malicious entry in Autoruns



Deleting Autostarts

To delete a selected autostart entry, click the **Entry** menu and then click **Delete**, or press Ctrl+D. To disable an entry without deleting it, clear the check box at the left end of the row. Before deleting any entries, record the full path to each malicious file, so you can remove them later.

After deleting or disabling suspicious autostarts, refresh the list by clicking the **Refresh** button on the toolbar or pressing F5. If you overlooked any malicious processes, they may monitor the autostart list and recreate any entries you delete. If this happens, return to Step 2 and use Process Explorer and Process Monitor to find and eliminate the responsible processes.

Step 5: Delete Malware Files

After terminating malicious processes and deleting autostart entries, the next step is to remove the malicious files themselves by visiting the file locations you recorded during the investigation, locating the malicious files, and deleting them.

Steps 6 and 7: Reboot and Repeat

To verify that you've eliminated the malware, reboot the computer and start the process over with step 1. Some malware families expend considerable effort to avoid detection, and repeating the investigation process a few times may help you uncover malicious processes and files that you missed earlier.

Conclusion

Unfortunately, the process of eliminating malware from a computer is likely to become much harder in the next few years. Malware has become a lucrative business for the criminals who create and distribute it, and they have a financial incentive to find new ways to evade detection and make malicious files and processes harder to remove.

Therefore, understanding how malware spreads, operates, and defends itself at a fundamental level should be considered a prerequisite for IT professionals charged with protecting their users from attack and containing outbreaks when they occur. However, the best guidance is that which helps prevent malware infection from ever occurring. For more information about how to prevent malware infection, see the Microsoft Malware Protection Center at www.microsoft.com/security/portal.

Promoting Safe Browsing

Windows Internet Explorer is a valuable source of data for the *Microsoft Security Intelligence Report*. Internet Explorer versions 7, 8, and 9 have built-in protection technologies that help protect users from attackers seeking to take advantage of them. Internet Explorer users who opt into providing telemetry data give Microsoft valuable insights into the ever-evolving tactics that attackers are using around the world. In response to customer requests from many parts of the world, this section of the *Microsoft Security Intelligence Report* provides details about the different security technologies that are included in Windows Internet Explorer.

All of the Internet Explorer versions examined here include security technologies that help establish and maintain a safe browsing experience for users. As attacks continue to evolve and new types of threats emerge, Microsoft has released new versions of Internet Explorer with new security technologies and strengthened implementations of older ones.

The following table shows a sample of security technologies across recent versions of Internet Explorer.

Figure 72. Security and privacy technologies in recent versions of Internet Explorer

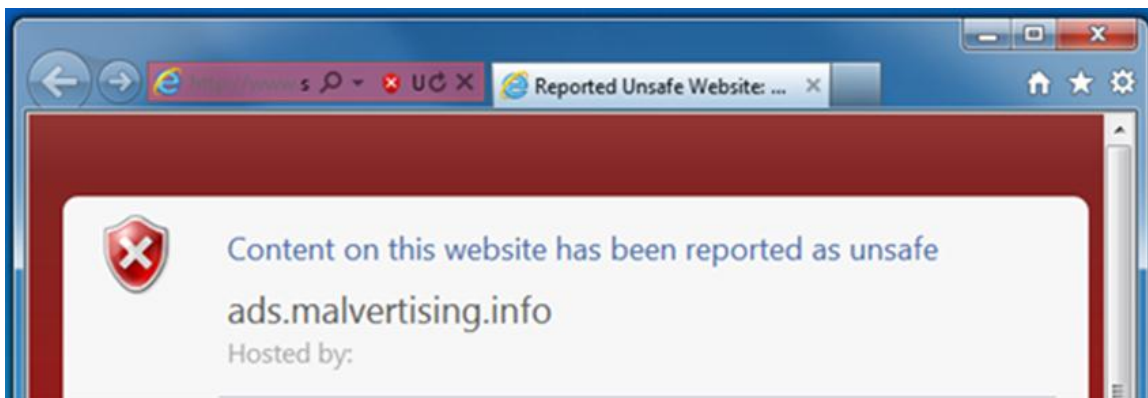
Security and privacy technologies	Internet Explorer 7	Internet Explorer 8	Internet Explorer 9
Security by default	•	•	•
SmartScreen – Phishing Filter	•	•	•
SmartScreen – Antimalware protection		•	•
InPrivate Browsing		•	•
Cross-site scripting filter		•	•
SmartScreen – Application Reputation			•
Tracking Protection			•
ActiveX® Filtering			•

SmartScreen Filter

SmartScreen Filter helps protect against phishing websites and sites known to distribute malware by blocking navigation to malicious sites or downloads. This feature helps reduce the likelihood of an attack and saves users time by stopping malware downloads before they infect a PC. SmartScreen Filter provides protection from different threats with a set of sophisticated tools and features:

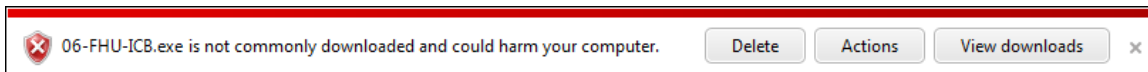
- Antiphishing protection screens threats from impostor websites that seek to acquire personal information such as user names, passwords, and billing data.
- Antimalware protection helps prevent the download of harmful software.

Figure 73. SmartScreen Filter in Internet Explorer 9



- Application Reputation removes unnecessary security warnings for well-known files, and shows severe warnings for unknown downloads that it considers high-risk.

Figure 74. Application Reputation in Internet Explorer 9



Microsoft strongly recommends that Internet Explorer users enable SmartScreen Filter to take advantage of the protections it provides.

ActiveX Filtering

ActiveX is a technology embedded in many popular websites to enrich the browsing experience. ActiveX plug-ins can be used for things such as playing videos, displaying animations, and viewing certain kinds of files. However, ActiveX can also pose security risks and slow down browser performance. Internet Explorer 8 added per-site ActiveX controls, which allowed users to restrict an ActiveX plug-in to one particular domain. Internet Explorer 9 introduces ActiveX Filtering, which provides users with more control over which sites can use ActiveX controls.

When ActiveX Filtering is enabled, only sites that are trusted by users can run ActiveX controls. This feature reduces the attack surface of a PC by restricting the ability to run ActiveX components to trusted sites. Users can allow specific sites to run ActiveX controls through an icon in the address bar. IT administrators can also enable ActiveX Filtering via Group Policy to prevent users from downloading ActiveX controls from the Internet Zone.

Figure 75. ActiveX technologies in recent versions of Internet Explorer

ActiveX technologies	Internet Explorer 7	Internet Explorer 8	Internet Explorer 9
Per-Site ActiveX		•	•
ActiveX Filtering			•

Cross-site scripting filter

Internet Explorer 8 and 9 include a cross-site scripting (XSS) filter that can help identify and block cross-site scripting attacks, which attempt to exploit vulnerabilities in legitimate websites. XSS-based attacks can steal login information and passwords, perform actions on behalf of users, or cause more damage. If an XSS attack is detected, Internet Explorer 9 can disable the harmful scripts. The cross-site scripting filter is turned on by default to help protect users.

Other browser defenses

Internet Explorer also contains technologies that make it harder to exploit memory vulnerabilities in the browser and its extensions. These technologies help stop an attacker's code from running, or else terminate the browser tab if an

exploit is detected. A listing of the browser defenses is provided in Figure 76, and brief descriptions of each are provided after the table.

Figure 76. Other browser defenses in recent versions of Internet Explorer

Browser defenses	Internet Explorer 7	Internet Explorer 8	Internet Explorer 9
Protected Mode	•	•	•
Data Execution Prevention	•	•	•
ASLR		•	•
Safe Structured Exception Handling			•
Enhanced Stack Buffer Overrun Detection			•

- **Protected Mode.** The Protected Mode feature takes advantage of Windows security enhancements to limit the damage an attacker can do. By limiting the privileges that the browser process has, many parts of the operating system, such as the file system, are off limits to the attacker.
- **Data Execution Prevention.** This feature prevents the execution of data placed into memory by an attacker. It is disabled by default in Internet Explorer 7 and enabled in later versions.
- **ASLR.** The Address Space Layout Randomization feature makes the memory layout of a PC unpredictable, which helps prevent attackers from being able to successfully exploit the PC. Before this technology, attackers were sometimes able to successfully exploit PCs by assuming that a specific program occupied a specific memory address, which they then targeted.
- **Safe Structured Exception Handling (SafeSEH).** This feature prevents attackers from injecting malicious code into the exception handling chain.
- **Enhanced Stack Buffer Overrun Detection.** This feature helps prevent stack buffer overruns by detecting stack corruption and preventing execution if such corruption is encountered.

Group Policy and the Security Compliance Manager

- Internet Explorer security features can be controlled by Group Policy. For example, IT administrators can mandate that the SmartScreen Filter is enabled and prevent users from circumventing Application Reputation warnings. Such controls allow an organization to enhance its security and save costs that result from fixing malware infections.
- Microsoft provides security baselines to help IT administrators configure Group Policy Objects (GPOs) that are specific to their needs. These baselines provide a set of standard recommended settings, which administrators can modify as needed. The Security Compliance Manager tool through which the baselines are accessed is available from the [Microsoft Security Compliance Manager](#) page on Microsoft TechNet.



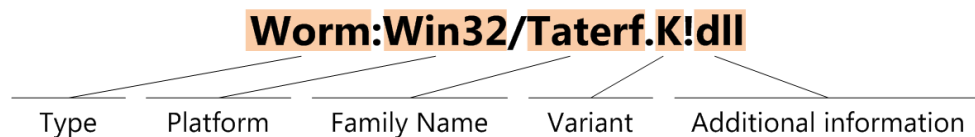
Appendixes

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 that are 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 77.

Figure 77. 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 *Microsoft 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 (www.microsoft.com/mmpc) 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 *Microsoft 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 *Microsoft Security Intelligence Report*, a threat name consisting of a platform and family name (for example, “Win32/Taterf”) is a reference to a family. When a longer threat name is given (for example, “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 would be referred to simply as “Taterf” on subsequent mention in some places, and Worm:Win32/Taterf.K simply as “Taterf.K.”

Appendix B: Data Sources

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 data allows the 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 webpage scans per year to seek out malicious content. After such content is detected, Bing displays warnings to users about it to help prevent infection.
- [Windows Live® Hotmail®](#) has hundreds of millions of active email users in more than 30 countries/regions around the world.
- [Forefront Online Protection for Exchange](#) (FOPE) protects the networks of thousands of enterprise customers worldwide by helping to prevent malware from spreading through email. FOPE scans billions of email messages every year to identify and block spam and malware.
- [Microsoft Forefront Endpoint Protection](#) is a unified product that provides protection from malware and potentially unwanted software for enterprise desktops, laptops, and server operating systems. It uses the Microsoft Malware Protection Engine and the Microsoft antivirus signature database to provide real-time, scheduled, and on-demand protection.
- [Windows Defender](#) is a program that is 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 that Microsoft 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 more than 600 million times each month on average in 1H11. 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 Security Essentials](#) is a real-time protection product that combines an antivirus and antispyware scanner with phishing and firewall protection.

- The [Microsoft Safety Scanner](#) is a free downloadable security tool that provides on-demand scanning and helps remove malware and other malicious software. The Microsoft 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 computer from becoming infected.
- [SmartScreen Filter](#), a feature in Internet Explorer 8 and 9, offers 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.

Figure 78. US privacy statements for the Microsoft products and services used in this report

Product or Service	Privacy Statement URL
Bing	privacy.microsoft.com/en-us/bing.aspx
Windows Live Hotmail	privacy.microsoft.com/en-us/fullnotice.aspx
Forefront Online Protection for Exchange	https://admin.messaging.microsoft.com/legal/privacy/en-us.htm
Windows Defender	www.microsoft.com/windows/products/winfamily/defender/privacypolicy.aspx
Malicious Software Removal Tool	www.microsoft.com/security/pc-security/msrt-privacy.aspx
Forefront Endpoint Protection	www.microsoft.com/download/en/details.aspx?id=23308
Microsoft Security Essentials	www.microsoft.com/en-us/security_essentials/privacy.aspx
Microsoft Safety Scanner	www.microsoft.com/security/scanner/en-us/Privacy.aspx
Windows Internet Explorer 9	windows.microsoft.com/en-US/internet-explorer/products/ie-9/windows-internet-explorer-9-privacy-statement

Appendix C: Worldwide Infection Rates

“Global Infection Rates,” on page 51, explains how threat patterns differ significantly in different parts of the world. Figure 79 shows the infection rates in locations with at least 100,000 quarterly MSRT executions in 1H11, as determined by geolocation of the IP address of the reporting computer. (CCM is the number of computers cleaned for every 1,000 executions of MSRT. See page 49 for more information about the CCM metric and how it is calculated.)

Figure 79. Infection rates (CCM) for locations around the world in 1H11, by quarter

Country/Region	1Q11	2Q11
Worldwide	11.0	9.8
Albania	23.7	25.0
Algeria	20.8	16.2
Angola	21.4	20.1
Argentina	11.4	11.1
Armenia	9.2	8.0
Australia	5.3	4.6
Austria	4.6	3.4
Azerbaijan	11.4	10.6
Bahamas, The	17.4	14.3
Bahrain	16.5	19.2
Bangladesh	13.0	13.7
Barbados	7.5	6.4
Belarus	6.0	6.0
Belgium	6.4	5.6
Bolivia	13.3	14.3
Bosnia and Herzegovina	18.4	16.4
Brazil	19.2	18.8
Brunei	14.4	12.9

Country/Region	1Q11	2Q11
Bulgaria	13.9	10.7
Cambodia	9.2	12.0
Cameroon	15.3	11.3
Canada	4.4	5.2
Chile	15.4	10.8
China	2.4	2.3
Colombia	11.8	11.5
Costa Rica	11.8	8.9
Côte d'Ivoire	15.3	12.7
Croatia	14.5	10.9
Cyprus	15.1	10.9
Czech Republic	5.2	2.9
Denmark	2.6	3.0
Dominican Republic	18.9	16.7
Ecuador	14.2	11.2
Egypt	20.9	19.5
El Salvador	13.6	10.7
Estonia	6.6	4.9
Ethiopia	10.2	10.9
Finland	1.4	1.3
France	6.0	5.0
Georgia	22.7	21.6
Germany	3.6	3.2
Ghana	13.7	11.5
Greece	13.0	10.1
Guadeloupe	14.8	13.0
Guatemala	12.4	10.7
Honduras	15.0	12.4
Hong Kong S.A.R.	8.9	7.9
Hungary	8.7	6.9
Iceland	6.8	4.7
India	15.2	15.9
Indonesia	16.2	18.4
Iran	9.1	10.0

Country/Region	1Q11	2Q11
Iraq	13.1	18.0
Ireland	5.9	4.7
Israel	15.1	12.1
Italy	7.8	6.4
Jamaica	16.2	12.5
Japan	2.7	2.1
Jordan	17.6	18.5
Kazakhstan	10.1	8.8
Kenya	13.0	11.4
Korea	30.1	19.8
Kuwait	17.0	15.5
Latvia	11.9	9.2
Lebanon	15.4	15.8
Lithuania	13.5	10.7
Luxembourg	4.2	3.2
Macao S.A.R.	6.9	5.8
Macedonia, F.Y.R.O.	20.2	14.4
Malaysia	13.4	12.0
Malta	8.7	6.0
Martinique	13.5	10.3
Mauritius	12.0	12.1
Mexico	16.7	13.5
Moldova	7.4	6.7
Mongolia	10.7	10.8
Morocco	14.4	13.1
Mozambique	18.1	14.3
Nepal	18.9	23.7
Netherlands	4.6	5.3
New Zealand	5.7	5.1
Nicaragua	11.6	9.2
Nigeria	13.1	10.6
Norway	2.9	2.5
Oman	19.3	18.1
Pakistan	27.7	31.1

Country/Region	1Q11	2Q11
Palestinian Authority	27.5	32.7
Panama	15.8	12.8
Paraguay	8.9	7.7
Peru	16.8	13.7
Philippines	11.7	11.0
Poland	14.1	11.4
Portugal	11.5	9.8
Puerto Rico	13.4	10.7
Qatar	61.5	34.4
Réunion	11.9	11.1
Romania	16.5	15.3
Russia	6.7	6.0
Saudi Arabia	16.4	16.2
Senegal	15.1	13.0
Serbia	16.0	15.6
Singapore	12.6	9.0
Slovakia	9.6	6.1
Slovenia	9.0	6.3
South Africa	13.4	10.6
Spain	13.2	11.4
Sri Lanka	11.3	12.0
Sudan	14.8	16.7
Sweden	2.8	2.4
Switzerland	3.5	2.8
Syria	11.2	14.0
Taiwan	17.7	16.1
Tanzania	17.6	13.6
Thailand	18.0	19.6
Trinidad and Tobago	17.5	11.9
Tunisia	16.0	13.6
Turkey	28.2	25.5
Uganda	16.9	15.0
Ukraine	7.4	6.6
United Arab Emirates	18.9	16.7

Country/Region	1Q11	2Q11
United Kingdom	5.1	5.1
United States	5.6	5.6
Uruguay	6.1	6.1
Venezuela	9.8	8.5
Vietnam	12.8	15.8
Yemen	20.4	21.7

Forefront Online Protection for Exchange (FOPE) tracks spambot activity around the world. Figure 80 lists the countries and regions that hosted at least 0.1 percent of the IP addresses used by spambots in 1H11.

Figure 80. Countries and regions hosting significant numbers of spambot IP addresses in 1H11

Country/Region	Percent of World Spambot IP Addresses	
	1Q11	2Q11
India	10.9%	11.0%
Korea	2.9%	8.4%
Russia	7.6%	7.7%
Vietnam	4.0%	7.3%
Indonesia	2.4%	5.6%
United States	6.0%	4.9%
Brazil	3.5%	4.4%
Ukraine	3.0%	3.3%
Romania	2.8%	2.3%
China	1.4%	2.0%
United Kingdom	3.4%	1.8%
Poland	2.3%	1.7%
Taiwan	2.1%	1.6%
Italy	3.6%	1.5%
Pakistan	0.63%	1.4%
Philippines	1.0%	1.4%
Colombia	1.6%	1.3%
Turkey	1.5%	1.3%
Kazakhstan	1.2%	1.2%
Israel	1.5%	1.0%
Australia	1.3%	1.0%
France	2.2%	0.98%
Spain	1.7%	0.96%

Country/Region	Percent of World Spambot IP Addresses	
	1Q11	2Q11
Argentina	1.0%	0.95%
Serbia	1.2%	0.84%
Saudi Arabia	0.99%	0.81%
Germany	1.7%	0.79%
Iran	0.81%	0.74%
Bulgaria	1.2%	0.68%
Morocco	0.63%	0.67%
Peru	0.66%	0.66%
Netherlands	0.62%	0.64%
Singapore	0.70%	0.59%
Belarus	0.36%	0.55%
Thailand	1.3%	0.53%
Chile	0.51%	0.52%
South Africa	0.55%	0.52%
Malaysia	0.45%	0.50%
Mexico	0.59%	0.46%
Czech Republic	0.83%	0.45%
Dominican Republic	0.30%	0.45%
Kenya	0.32%	0.39%
United Arab Emirates	0.34%	0.37%
Greece	0.78%	0.35%
Canada	0.84%	0.35%
Egypt	0.17%	0.34%
Macedonia, F.Y.R.O.	0.35%	0.30%
Austria	0.44%	0.28%
Kuwait	0.27%	0.28%
Bangladesh	0.22%	0.27%
Japan	0.43%	0.25%
Lithuania	0.38%	0.25%
Tunisia	0.26%	0.25%
Hong Kong S.A.R.	0.26%	0.24%
Venezuela	0.31%	0.24%
Portugal	0.40%	0.24%
Belgium	0.21%	0.23%
Sri Lanka	0.19%	0.22%
Sweden	0.20%	0.21%
New Zealand	0.21%	0.20%

Country/Region	Percent of World Spambot IP Addresses	
	1Q11	2Q11
Hungary	0.29%	0.19%
Azerbaijan	0.20%	0.19%
Algeria	0.11%	0.18%
Nigeria	0.11%	0.15%
Switzerland	0.25%	0.15%
Latvia	0.18%	0.14%
Guatemala	0.18%	0.14%
Costa Rica	0.12%	0.13%
Ireland	0.22%	0.13%
Slovakia	0.27%	0.13%
Mongolia	0.15%	0.13%
Croatia	0.31%	0.12%
Ghana	0.12%	0.11%
Slovenia	0.13%	0.11%
Lebanon	0.12%	0.11%
Bolivia	0.17%	0.11%
Denmark	0.12%	0.10%
Palestinian Authority	0.14%	0.10%
Armenia	0.07%	0.10%
Panama	0.12%	0.10%
Jordan	0.06%	0.10%
Cameroon	0.09%	0.10%

Appendix D: Microsoft Office Vulnerabilities Encountered in 1H11

To illustrate the importance of applying all service packs and other security updates, this table compares the relative levels of vulnerability of different versions of Microsoft Office as originally released, with the most recent service pack for each version installed, and with all security updates installed. See “Microsoft Office File Format Exploits” on page 43 for more information.

Figure 81. Versions of Microsoft Office and whether they are vulnerable to exploits observed in 1H11

Vulnerability	Office 2003 RTM	Office 2003 SP3	Office 2007 RTM	Office 2007 SP2	Office 2010 RTM	Office 2010 SP1	All Updates Installed*
CVE-2006-2492	Yes	No	No	No	No	No	No
CVE-2006-0022	Yes	No	No	No	No	No	No
CVE-2006-6456	Yes	No	No	No	No	No	No
CVE-2007-0671	Yes	No	No	No	No	No	No
CVE-2008-0081	Yes	No	No	No	No	No	No
CVE-2009-0238	Yes	Yes	Yes	No	No	No	No
CVE-2009-0557	Yes	Yes	Yes	Yes	No	No	No
CVE-2009-3129	Yes	Yes	Yes	Yes	No	No	No
CVE-2010-3333	Yes	Yes	Yes	Yes	Yes	No	No
CVE-2011-0979	Yes	Yes	Yes	Yes	Yes	No	No

* Users of all supported versions of Office who install all security updates as they are released would be protected from all of the exploits encountered in the sample set.

Glossary

For additional information about these and other terms, visit the MMPC glossary at www.microsoft.com/security/portal/Threat/Encyclopedia/Glossary.aspx.

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 unauthorized access to and control of infected computers. Bots are a subcategory of backdoor trojans. Also see *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, such as peer-to-peer (P2P) networking. Computers in a botnet are often called nodes or zombies.

buffer overflow

An error in an application in which the data written into a buffer exceeds the current capacity of that buffer, thus overwriting adjacent memory. Because memory is overwritten, unreliable program behavior may result and, in certain cases, allow arbitrary code to run.

C&C

Short for command and control. See *botnet*.

CCM

Short for *computers cleaned per mille* (thousand). The number of computers cleaned for every 1,000 executions of MSRT. For example, if 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 ($200 \div 50,000 \times 1,000$).

clean

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

cross-site scripting

Abbreviated XSS. An attack technique in which an attacker inserts malicious HTML and JavaScript into a vulnerable Web page, often in an effort to distribute malware or to steal sensitive information from the Web site or its visitors. Despite the name, cross-site scripting does not necessarily involve multiple websites. *Persistent cross-site scripting* involves inserting malicious code into a database used by a web application, potentially causing the code to be displayed for large numbers of visitors.

definition

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

disclosure

Revelation of the existence of a vulnerability to a third party.

disinfect

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

downloader/dropper

See *trojan downloader/dropper*.

exploit

Malicious code that takes advantage of software vulnerabilities to infect a computer or perform other harmful actions.

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 that is capable of detecting a 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 webpage, 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 that are hosted by trusted websites.

in the wild

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

keylogger

A program that sends keystrokes or screen shots to an attacker. Also see *password stealer (PWS)*.

malware

Any software that is designed specifically to cause damage to a user's computer, server, or network. Viruses, worms, and trojans are all types of malware.

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)*.

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*. Also see *monitoring tool*.

payload

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

phishing

A method of credential theft that tricks Internet users into revealing personal or financial information online. Phishers use phony websites or deceptive email 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, 8, or 9, and being blocked by the Phishing Filter or SmartScreen Filter. Also see *malware impression*.

pop-under

A webpage that opens in a separate window that appears beneath the active browser window. Pop-under windows are commonly used to display advertisements.

potentially unwanted software

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

remote control software

A program that provides access to a computer from a remote location. Such 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 that 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 hiding itself or other malware.

signature

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

social engineering

A technique that defeats security precautions by exploiting human vulnerabilities. Social engineering scams can be both online (such as receiving email messages that ask the recipient to click the attachment, which is actually malware) and offline (such as receiving a phone call from someone posing as a representative from one's 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 email. Malware authors may use spam to distribute malware, either by attaching the malware to email messages or by sending a message

containing a link to the malware. Malware may also harvest email addresses for spamming from compromised machines or may use compromised machines to send spam.

spambot

A bot that sends spam at the direction of a remote attacker, usually as part of a spam botnet.

spyware

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

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 a computer that it has infected, either by downloading them from a remote computer or by obtaining them directly from a copy contained in its own code.

virus

Malware that replicates, typically by infecting other files in the computer, to allow 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.

wild

See in the wild.

worm

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

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 (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/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 family of worms that spreads by copying itself to the mapped drives of an infected computer. The mapped drives may include network or removable drives.

Win32/Bagle. A worm that spreads by emailing 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/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/Begseabug. A trojan that downloads and executes arbitrary files on an affected computer.

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

Win32/Brontok. A mass-mailing email worm that spreads by sending copies of itself as email attachments to addresses gathered from files on the infected computer, and by copying itself to removable volumes. Brontok can disable security software, and may conduct DoS attacks against certain websites.

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/ClickPotato. A program that displays pop-up and notification-style advertisements based on the user's browsing habits.

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.

JS/CVE-2010-0806. A detection for malicious JavaScript that attempts to exploit the vulnerability addressed by Microsoft Security Bulletin [MS10-018](#).

Java/CVE-2010-0840. A detection for a malicious and obfuscated Java class that exploits a vulnerability described in CVE-2010-0840. Oracle Corporation addressed the vulnerability with a security update in March 2010.

Win32/Cycbot. A backdoor trojan that allows attackers unauthorized access and control of an affected computer. After a computer is infected, the trojan connects to a specific remote server to receive commands from attackers.

Win32/Delf. A detection for various threats written in the Delphi programming language. The behaviors displayed by this malware family are highly variable.

AndroidOS/DroidDream. A malicious program that affects mobile devices running the Android operating system. It may be bundled with clean applications, and is capable of allowing a remote attacker to gain access to the mobile device.

Win32/Dynamer. A generic detection for a variety of threats.

MacOS_X/FakeMacdef. A rogue security software family that affects Apple Mac OS X. It has been distributed under the names MacDefender, MacSecurity, MacProtector, and possibly others.

Win32/FakePAV. A rogue security software family that masquerades as Microsoft Security Essentials.

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

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

Win32/GameHack. Malware that is often bundled with game applications. It commonly displays unwanted pop-up advertisements and may be installed as a web browser helper object.

Win32/Hamweq. A worm that spreads through removable drives, such as USB memory sticks. It may contain an IRC-based backdoor that enables 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.

HTML/IframeRef. A generic detection for specially formed IFrame tags that point to remote websites that contain malicious content.

Win32/Jeefo. A parasitic file-infector virus that infects Windows portable executable (PE) files that are greater than or equal to 102,400 bytes long. When an infected PE file runs, the virus tries to run the original content of the file.

Win32/Keygen. A generic detection for tools that generate product keys for illegally obtained versions of various software products.

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

Java/Loic. An open-source network attack tool designed to perform denial-of-service (DoS) attacks.

Unix/Lotoor. A detection for specially crafted Android programs that attempt to exploit vulnerabilities in the Android operating system to gain root privilege.

Win32/Meredrop. A generic detection for trojans that drop and execute multiple forms of malware on a local computer. These trojans are usually packed, and may contain multiple trojans, backdoors, or worms. Dropped malware may connect to remote websites and download additional malicious programs.

Win32/Microjoin. A generic detection for tools that bundle malware files with clean files in an effort to deploy malware without being detected by security software.

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

Win32/Nuqel. A worm that spreads via mapped drives and certain instant messaging applications. It may modify system settings, connect to certain websites, download arbitrary files, or take other malicious actions.

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

Win32/OfferBox. A program that displays offers based on the user's web browsing habits. Some versions may display advertisements in a pop-under window. Win32/OfferBox may be installed without adequate user consent by malware.

Win32/OpenCandy. An adware program that may be bundled with certain third-party software installation programs. Some versions may send user-specific information, including a unique machine code, operating system information, locale, and certain other information to a remote server without obtaining adequate user consent.

Win32/Pameseg. A fake program installer that requires the user to send SMS messages to a premium number to successfully install certain 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. Such files contain malicious JavaScript that executes when the file is opened.

JS/Pornpop. A generic detection for specially-crafted JavaScript-enabled objects that attempt to display pop-under advertisements, usually with adult content.

Win32/Pramro. A trojan that creates a proxy on the infected computer for email and HTTP traffic, and is used to send spam email.

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 computer.

Win32/Ramnit. A family of multi-component malware that infects executable files, Microsoft Office files, and HTML files. Win32/Ramnit spreads to removable drives and steals sensitive information such as saved FTP credentials and browser cookies. It may also open a backdoor to await instructions from a remote attacker.

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/RealVNC. A management tool that allows a computer to be controlled remotely. It can be installed for legitimate purposes but can also be installed from a remote location by an attacker.

Win32/Renocide. A family of worms that spread via local, removable, and network drives and also using file sharing applications. They have IRC-based backdoor functionality, which may allow a remote attacker to execute commands on the affected computer.

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

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

Win32/Rlsloup. A family of trojans that are used to send spam email. Rlsloup consists of several components, including an installation trojan component and a spamming payload component.

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.

JS/ShellCode. A generic detection for JavaScript-enabled objects that contain exploit code and may exhibit suspicious behavior. Malicious websites and malformed PDF documents may contain JavaScript that attempts to execute code without the affected user's consent.

Win32/ShopperReports. Adware that displays targeted advertising to affected users while browsing the Internet, based on search terms entered into search engines.

Win32/Sinowal. A family of password-stealing and backdoor trojans. It may try to install a fraudulent SSL certificate on the computer. Sinowal may also capture user data such as banking credentials from various user accounts and send the data to Web sites specified by the attacker.

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

Win32/Sogou. A Chinese-language browser toolbar that may display pop-up advertisements and may download and install additional components without user consent.

VBS/Startpage. A detection for various threats that change the configured start page of the affected user's web browser, and may also perform other malicious actions.

Win32/Stuxnet. A multi-component family that spreads via removable volumes by exploiting the vulnerability addressed by Microsoft Security Bulletin [MS10-046](#).

Win32/Swinnag. A worm that spreads via removable drives and drops a randomly-named DLL in the Windows system folder.

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

Win32/Tedroo. A trojan that sends spam email messages. Some variants may disable certain Windows services or allow backdoor access by a remote attacker.

Win32/VB. A detection for various threats written in the Visual Basic programming language.

Win32/Vobfus. A family of worms that spreads via network drives and removable drives and download/executes arbitrary files. Downloaded files may include additional malware.

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

Win32/Yimfoca. A worm family that spreads via common instant messaging applications and social networking sites. It is capable of connecting to a remote HTTP or IRC server to receive updated configuration data. It also modifies certain system and security settings.

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

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



Microsoft[®]

One Microsoft Way
Redmond, WA 98052-6399
microsoft.com/security