Microsoft Security Intelligence Report

Volume 20 | July through December, 2015

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 website references, may change without notice. You bear the risk of using it.

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

Authors

Charlie Anthe
Cloud and Enterprise Security

Nir Ben Zvi
Enterprise and Cloud Group

Patti Chrzan
Microsoft Digital Crimes Unit

Bulent Egilmez
Office 365 - Information Protection

Elia Florio
Windows Defender Labs

Chad Foster
Bing

Roger Grimes
Microsoft IT

Paul Henry
Wadeware LLC

Beth Jester
Windows Defender

Jeff Jones
Corporate Communications

Dana Kaufman
Azure Active Directory Team

Nasos Kladakis
Azure Active Directory Team

Daniel Kondratyuk
Azure Active Directory Team

Andrea Lelli
Windows Defender Labs

Geoff McDonald
Windows Defender Labs

Michael McLaughlin
Identity Services

Nam Ng
Enterprise Cybersecurity Group

Niall O'Sullivan
Microsoft Digital Crimes Unit

Daryl Pecelj
Microsoft IT Information Security and Risk Management

Anthony Penta
Safety Platform

Ina Ragragio
Windows and Devices Group

Tim Rains
Commercial Communications

Paul Rebriy
Bing

Stefan Sellmer
Windows Defender Labs

Mark Simos
Enterprise Cybersecurity Group

Vikram Thakur
Windows Defender Labs

Alex Weinert
Azure Active Directory Team

Terry Zink
Office 365 - Information Protection

Contributors

Joey Caparas
Windows Defender Labs

Satomi Hayakawa
CSS Japan Security Response Team

Ben Hope
Corporate Communications

Yurika Kakiuchi
CSS Japan Security Response Team

Russ McRee
Windows and Devices Group

Dolcita Montemayor
Windows Defender Labs

Wendi Okun
Corporate, External, and Legal Affairs

Laura A.Robinson
Microsoft IT

Jasmine Sesso
Windows Defender Labs

Norie Tamura
CSS Japan Security Response Team

Henk van Roest
CSS Security Response Team

Pete Voss
Corporate, External, and Legal Affairs

Steve Wacker
Wadeware LLC

Iaan Whiltshire
Windows Defender Labs

Table of contents

Featured intelligence 1

Worldwide threat assessment 53

Mitigating risk 139

About this report

The *Microsoft Security Intelligence Report (SIR)* focuses on software vulnerabilities, software vulnerability exploits, malware, and unwanted software. 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 third and fourth quarters of 2015, with trend data for the last several quarters 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.

Throughout the report, half-yearly and quarterly time periods are referenced using the *n*H*yy* or *n*Q*yy* formats, in which *yy* indicates the calendar year and *n* indicates the half or quarter. For example, 1H15 represents the first half of 2015 (January 1 through June 30), and 4Q14 represents the fourth quarter of 2014 (October 1 through December 31). To avoid confusion, please note 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. For information about this standard, see "Appendix A: Threat naming conventions" on page 154. In this report, any threat or group of threats that share a common unique base name is considered a family for the sake of presentation. This consideration includes threats that may not otherwise be considered families according to common industry practices, such as generic detections. For the purposes of this report, a threat is defined as a malware or unwanted software family or variant that is detected by the Microsoft Malware Protection Engine.

Foreword

We've been publishing threat intelligence reports for our customers, partners and the industry for 10 years now. During that time, we've published over *12,500 pages of threat intelligence*, 100+ blog posts, many videos, and delivered thousands of customer briefings all over the world. Over the years, the feedback from customers on the value of the intelligence and guidance that we've published in the Microsoft Security Intelligence Report has been nothing short of overwhelming.

In the last few years, things have changed dramatically in the threat landscape, our visibility into it, and the speed at which we can make adjustments to help protect customers. The cloud has been a security game changer and it's becoming more powerful every day.

A few of the CISOs I have talked to still aren't leveraging cloud services to help them protect their organization. Their current on-premises security strategy has them investing in SIEMs to get improved visibility into their IT environment. This doesn't provide them with the intelligence they want on the threats that other organizations have had to face, so they augment their data by procuring multiple third party threat intelligence feeds. The hope is that combining all of this data will enable the organization to better protect, detect and respond to threats.

This approach has certainly benefited many organizations. But security teams know it has challenges. Not all threat intelligence feeds are equal; some data sets are stale. It can be hard to find meaningful threats in large data sets. More data can make this even harder. Attracting and retaining security talent to analyze this data is an industry-wide challenge. If organizations can't identify meaningful threats and take action in real time, the result can be more like a history lesson than it is helpful.

This is where the Microsoft cloud can help. Informed by trillions of signals from billions of sources, Microsoft creates an intelligent security graph that helps protect endpoints, better detect attacks and accelerate our response. The intelligent security graph is powered by inputs we receive across our endpoints, consumer services, commercial services and on-premises technologies.

Every day our machine learning systems process more than 10 terabytes of data, including information on over 13 billion logins from hundreds of millions of Microsoft Account users and Azure Active Directory accounts. We've included new data in this report that provides insight into how the Microsoft cloud uses this massive data and machine learning to literally detect and prevent over a million attacks every day.

The Microsoft cloud has the scale, the threat intelligence, and the security capabilities that CISOs are looking for. If you haven't evaluated or looked at our cloud services in a while, it's time to check out some of the new security capabilities. Start with Azure Security Center, Azure Active Directory Identity Protection, and Microsoft Cloud App Security. You won't be disappointed.

In addition, you'll see from some of the data in this report that Windows 10 has been providing superior protection compared to older operating systems.

I hope you find the 20th volume of the *Microsoft Security Intelligence Report* valuable.

Tim Rains
Director, Security
Microsoft

How to use this report

The *Microsoft Security Intelligence Report* has been released twice a year since 2006. Each volume is based upon data collected from millions of computers all over the world, which not only provides valuable insights on the worldwide threat landscape, both at home and at work, but also provides detailed information about threat profiles faced by computer users in more than a hundred individual countries and regions.

To get the most out of each volume, Microsoft recommends the following:

Read

Each volume of the report consists of several parts. The primary report typically consists of a worldwide threat assessment, one or more feature articles, guidance for mitigating risk, and some supplemental information. A summary of the key findings in the report can be downloaded and reviewed separately from the full report; it highlights a number of facts and subjects that are likely to be of particular interest to readers. The regional threat assessment, available for download and in interactive form at www.microsoft.com/security/sir/threat, provides individual summaries of threat statistics and security trends for more than 100 countries and regions worldwide.

Reading the volume in its entirety will provide readers with the most benefit and context, but the report is designed to provide value in small doses as well. Take a few minutes to review the summary information to find the information that will be of most interest to you and your organization. Consult the table of contents and the index to learn more about particular topics of interest.

Share

Microsoft also encourages readers to share each released volume, or its download link, with co-workers, peers, and friends with similar interests. The *Microsoft Security Intelligence Report* is written to be useful and accessible to a wide range of audiences. Each volume contains thousands of hours of research disseminated in easy to understand language, with advanced technical jargon kept to a minimum. Each section and article is written and reviewed to provide the most value for the time it takes to read.

Assess your own risk

Reading about the threats and risks that affect different types of environments presents a good opportunity to assess your own risks. Not every computer and entity faces the same risk from all threats. Assess your own risks and determine which topics and information can help you to best defend against the most significant risks.

The volume and scope of threats facing the typical organization make it important to prioritize. The greatest risk to any computer or organization is posed by currently and recently active threats. Pay attention to the threats that have most commonly affected your region or industry, focusing particularly on the most common successful attacks in the wild that cause the most problems. Give less consideration to very rare or theoretical-only attacks, unless your computers are at particular risk for such threats.

Educate

Microsoft strives to make this report one of the most valuable sources of threat and mitigation information that you can read and share. We encourage you to use the *Microsoft Security Intelligence Report* as a guide to educate your employees, friends, and families about security-related topics.

Anyone, including a business, may link, point to, or re-use articles in the *Microsoft Security Intelligence Report* for informational purposes, provided the material is not used for publication or sale outside of your company and you comply with the following terms: You must not alter the materials in any way. You must provide a reference to the URL at which the materials were originally found. You must include the Microsoft copyright notice followed by “Used with permission from Microsoft Corporation.” Please see Use of Microsoft Copyrighted Content for further information.

Ask questions

Contact your local Microsoft representative with any questions you have about the topics and facts presented in this report. We hope that each volume provides a good educational summary and helps promote dialog between people trying to best secure their computing devices. Thank you for trusting Microsoft to be your partner in the fight against malware, hackers, and other security threats.

Featured intelligence

PLATINUM: Targeted attacks in South and Southeast Asia

Microsoft proactively monitors the threat landscape for emerging threats. Part of this job involves keeping tabs on targeted activity groups, which are often the first ones to introduce new exploits and techniques that are later used widely by other attackers. The feature article "STRONTIUM: A profile of a persistent and motivated adversary," on page 3 of *Microsoft Security Intelligence Report, Volume 19 (January–June 2015)*, chronicled the activities of one such group that attracted interest because of its aggressive, persistent tactics and techniques as well as its repeated use of new zero-day exploits to attack its targets.

This section describes the history, behavior, and tactics of a newly discovered targeted activity group, which Microsoft has code-named PLATINUM. Microsoft is sharing some of the information it has gathered on this group in the hope that it will raise awareness of the group's activities and help organizations take immediate advantage of available mitigations that can significantly reduce the risks they face from this and similar groups.

Adversary profile

PLATINUM has been targeting its victims since at least as early as 2009, and may have been active for several years prior. Its activities are distinctly different not only from those typically seen in untargeted attacks, but from many targeted attacks as well. A large share of targeted attacks can be characterized as opportunistic: the activity group changes its target profiles and attack geographies based on geopolitical seasons, and may attack institutions all over the world. Like many such groups, PLATINUM seeks to steal sensitive intellectual property related to government interests, but its range of preferred targets is consistently limited to specific governmental organizations, defense institutes, intelligence agencies, diplomatic institutions, and telecommunication providers in South and Southeast Asia. The group's persistent use of *spear phishing* tactics (phishing attempts aimed at specific individuals) and access to previously undiscovered zero-day exploits have made it a highly resilient threat.

After researching PLATINUM, Microsoft has identified the following key characteristics of the group and its activities:

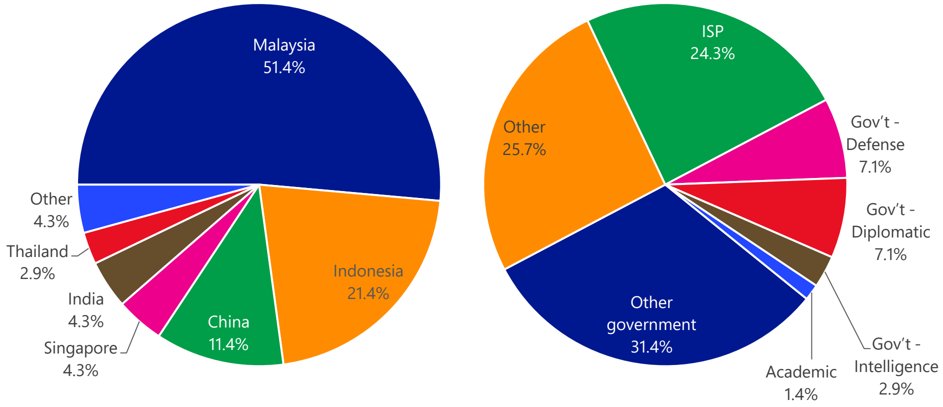
PLATINUM has been targeting its victims since at least as early as 2009.

- PLATINUM has conducted several cyber espionage campaigns since at least 2009.
- PLATINUM focuses on a small number of campaigns per year, which reduces the risk of detection and helps the group stay unnoticed and focused for a longer period of time.
- PLATINUM has focused on targets associated with governments and related organizations in South and Southeast Asia.
- PLATINUM has used multiple unpatched vulnerabilities in zero-day exploits against its victims.
- Spear phishing is the group's main method of infecting targeted users' computers.
- PLATINUM makes a concerted effort to hide their infection tracks, by self-deleting malicious components, or by using server side logic in "one shot mode" where remotely hosted malicious components are only allowed to load once
- PLATINUM often spear phishes its targets at their non-official or private email accounts, to gain access to the intended organization's network.
- PLATINUM uses custom-developed malicious tools and has the resources to update these tools often to avoid being detected.
- PLATINUM configures its backdoor malware to restrict its activities to victims' working hours, in an attempt to disguise post-infection network activity within normal user traffic.
- PLATINUM does not conduct its espionage activity to engage in direct financial gain, but instead uses stolen information for indirect economic advantages.
- In some cases, the combination of these mechanisms—use of undisclosed zero-day exploits, custom malware that is not used elsewhere, PLATINUM's skill in covering its tracks, and others—has enabled the group to compromise targets for several years without being detected.

Targeted activity groups are skilled at covering their tracks and evading detection, and it can be very difficult to definitively associate an activity group with a specific nation-state or group of individuals. Attackers could be patriotic groups, opportunistic cyber units, state-sponsored hackers, or intelligence agents. Although PLATINUM could belong to any one of the aforementioned categories, the group shows traits of being well funded, organized, and focused on information that would be of most use to government bodies.

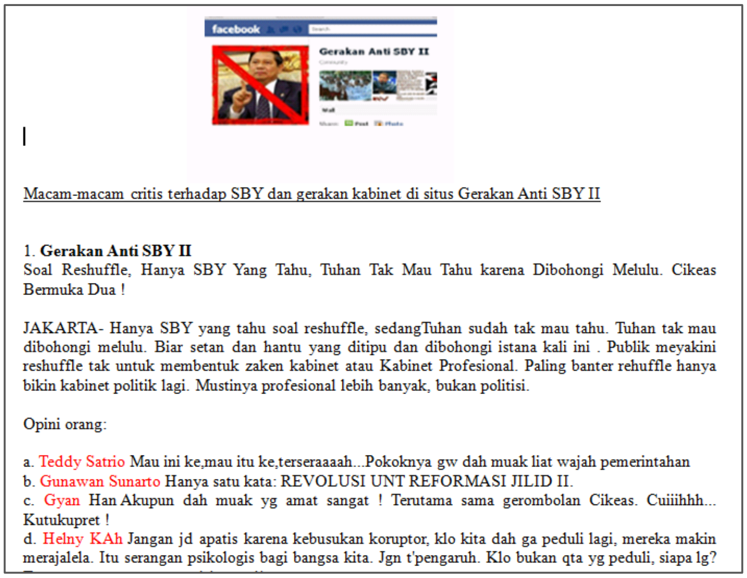
Methods of attack

Figure 1. Known victims attacked by PLATINUM since 2009, by country/region (left) and type of institution (right)



Although PLATINUM primarily targets its intended victims using spear phishing, some data indicates the group's usage of drive-by attacks against vulnerable browser-plugins. The group's methods for performing reconnaissance to determine whom to pursue remains unknown, and the number of victims targeted at each affected institution is consistently very small. In some cases, the victims were targeted at their non-official email addresses, demonstrating that the scope of PLATINUM's research capabilities is fairly extensive. For the initial infection, PLATINUM typically lures its victims by sending malicious documents that contain exploits for vulnerabilities in various software programs, with links or remotely loaded components (images or scripts or templates) that are delivered to targets only once. The group has made concerted efforts toward designing their initial spear-phishes in a manner that only delivers the final payload to the intended victim. The group is known to have used a number of zero-day exploits, for which no security update was available at the time of transmission, in these attempts. (All have subsequently been addressed by security updates from the affected vendors.)

Figure 2. A typical lure document sent by PLATINUM to a prospective victim



Macam-macam critis terhadap SBY dan gerakan kabinet di situs Gerakan Anti SBY II

1. **Gerakan Anti SBY II**
Soal Reshuffle, Hanya SBY Yang Tahu, Tuhan Tak Mau Tahu karena Dibohongi Melulu. Cikeas Bermuka Dua !

JAKARTA- Hanya SBY yang tahu soal reshuffle, sedangTuhan sudah tak mau tahu. Tuhan tak mau dibohongi melulu. Biar setan dan hantu yang ditipu dan dibohongi istana kali ini . Publik meyakini reshuffle tak untuk membentuk zaken kabinet atau Kabinet Profesional. Paling banter rehuffle hanya bikin kabinet politik lagi. Mustinya profesional lebih banyak, bukan politisi.

Opini orang:

a. Teddy Satrio Mau ini ke,mau itu ke,terseraaaah...Pokoknya gw dah muak liat wajah pemerintahan
b. Gunawan Sunarto Hanya satu kata: REVOLUSI UNT REFORMASI JILID II.
c. Gyan Han Akupun dah muak yg amat sangat ! Terutama sama gerombolan Cikeas. Cuiiihhh... Kutukupret !
d. Helny KAh Jangan jd apatis karena kebusukan koruptor, klo kita dah ga peduli lagi, mereka makin merajalela. Itu serangan psikologis bagi bangsa kita. Jgn t'pengaruh. Klo bukan qta yg peduli, siapa lg?

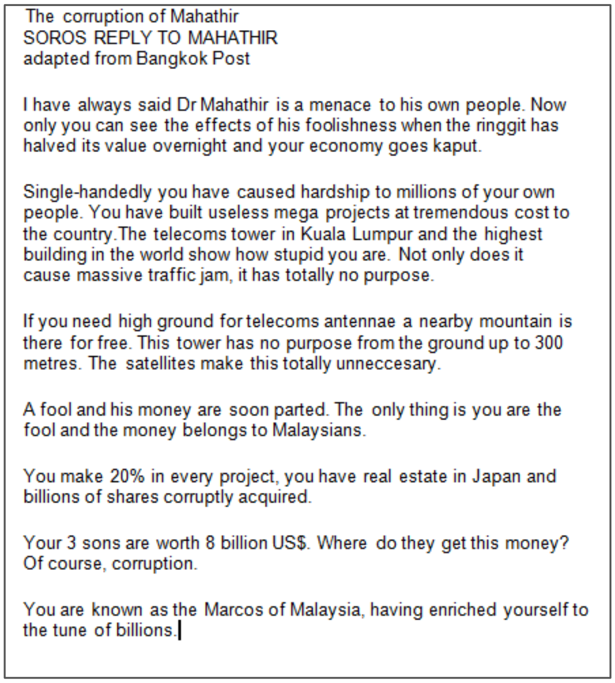
Lure documents are typically given topical names that may be of interest to the recipient. Such lures often address controversial subjects or offer provocative opinions, in an effort to incite the reader into opening them. Figure 3 shows a sample of such titles.

Figure 3. Example document titles used by PLATINUM to deliver exploits

SHA1	Filename
e9f900b5d01320ccd4990fd322a459d709d43e4b	Gambar gambar Rumah Gay Didiet Prabowo di Sentul Bogor.doc
9a4e82ba371cd2fedea0b889c879daee7a01e1b1	The real reason Prabowo wants to be President.doc
92a3ece981bb5e0a3ee4277f08236c1d38b54053	Malaysia a victim of American irregular warfare ops.doc
0bc08dca86bd95f43ccc78ef4b27d81f28b4b769	Tu Vi Nam Tan Mao 2011.doc
f4af574124e9020ef3d0a7be9f1e42c2261e97e6	Indians having fun.doc

These documents were sent to intended victims in Vietnam, Indonesia, India, and Malaysia, and the filenames contain references to cities, politicians, and current events in those locations. The oldest confirmed PLATINUM exploit was named "The corruption of Mahathir," a document that was transmitted in 2009 referencing the former prime minister of Malaysia, Mahathir Mohamad.

Figure 4. The oldest confirmed lure document sent by PLATINUM, in 2009



The corruption of Mahathir
SOROS REPLY TO MAHATHIR
adapted from Bangkok Post

I have always said Dr Mahathir is a menace to his own people. Now only you can see the effects of his foolishness when the ringgit has halved its value overnight and your economy goes kaput.

Single-handedly you have caused hardship to millions of your own people. You have built useless mega projects at tremendous cost to the country.The telecoms tower in Kuala Lumpur and the highest building in the world show how stupid you are. Not only does it cause massive traffic jam, it has totally no purpose.

If you need high ground for telecoms antennae a nearby mountain is there for free. This tower has no purpose from the ground up to 300 metres. The satellites make this totally unneccesary.

A fool and his money are soon parted. The only thing is you are the fool and the money belongs to Malaysians.

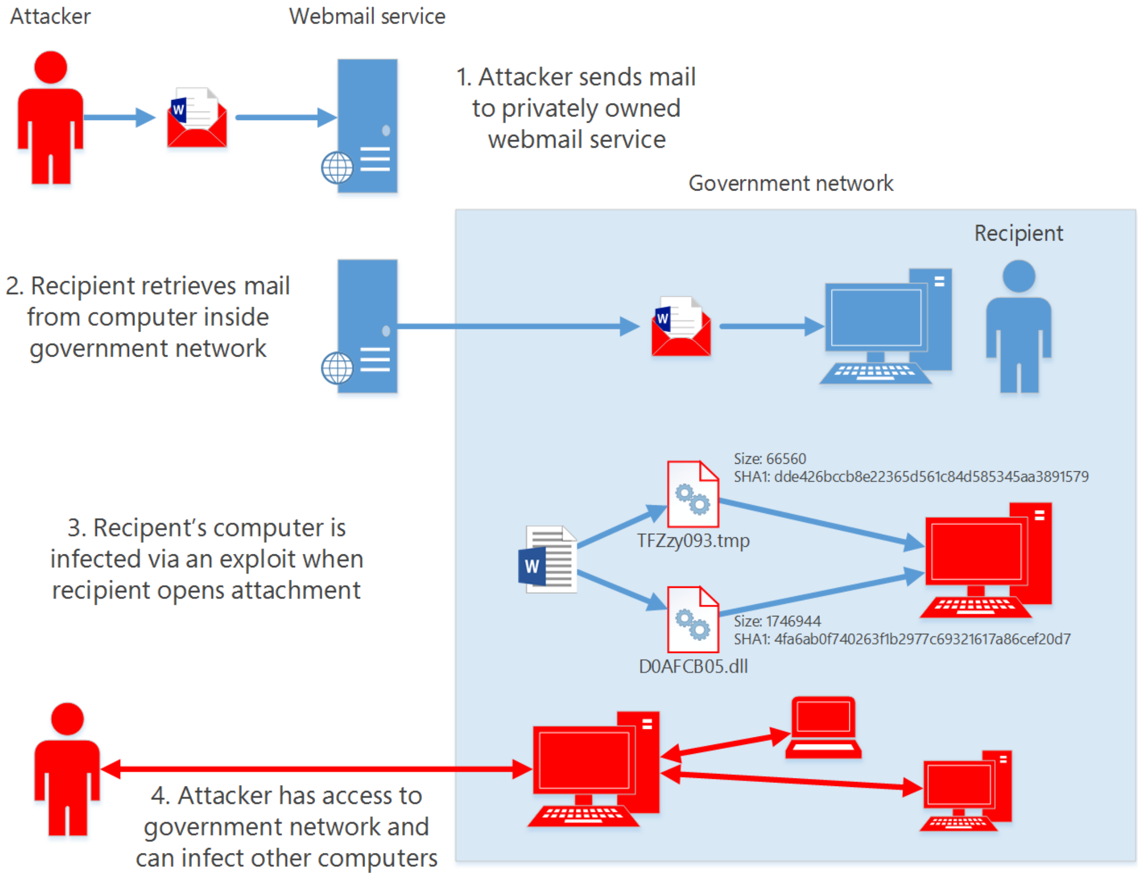
You make 20% in every project, you have real estate in Japan and billions of shares corruptly acquired.

Your 3 sons are worth 8 billion US$. Where do they get this money? Of course, corruption.

You are known as the Marcos of Malaysia, having enriched yourself to the tune of billions.

PLATINUM's recent activities remain focused on tactics such as these. In February 2016, PLATINUM was observed using a legitimate website dedicated to news about the Indian government as an infection vector. This site, which is not associated with the Indian government itself, also provides a free email service for its users, giving them email addresses with the site's own domain name. PLATINUM sent spear phishing messages to users of the service, which included some Indian government officials. After infecting an unsuspecting user this way, the attackers had complete control of the user's computer and used it as a stepping stone into the official network to which the user belonged.

Figure 5. PLATINUM used a private webmail service to infect a government network


Attacker
Webmail service
1. Attacker sends mail to privately owned webmail service
Government network
Recipient
2. Recipient retrieves mail from computer inside government network
Size: 66560
SHA1: dde426bccb8e22365d561c84d585345aa3891579
TFZzy093.tmp
3. Recipent's computer is infected via an exploit when recipient opens attachment
Size: 1746944
SHA1: 4fa6ab0f740263f1b2977c69321617a86cef20d7
D0AFCB05.dll
4. Attacker has access to government network and can infect other computers

PLATINUM's approach toward exploiting vulnerabilities varies between campaigns. In one case from 2013, the target was sent a malicious document through a spear phishing email message.[1] The document, when opened, used an embedded ActiveX control to download a JavaScript file from a remote site that used a previously unknown vulnerability in some versions of Windows (later designated CVE-2013-7331) to read information about the browser's installed components.[2]

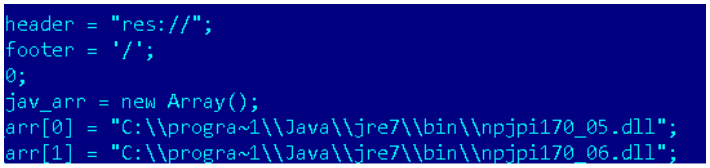
[1] Microsoft thanks Google for identifying and reporting this attack.

[2] Microsoft issued Security Bulletin MS14-052 in September 2014 to address the issue. CVE-2013-7331 has never affected Windows 10.

Figure 6. Malicious Word 2003 files used by PLATINUM to deliver CVE-2013-7331

Filename	SHA1	URL for PNG Exploit
Gerakan Anti SBY II.doc	1bdc1a0bc995c1beb363b11b71c14324be8577c9	mister.nofrillspace.com/users/web8_dice/4226/space.gif
Tu_Vi_Nam_Tan_Mao_2011.doc	2a33542038a85db4911d7b846573f6b251e16b2d	intent.nofrillspace.com/users/web11_focus/3807/space.gif
Wikileaks Indonesia.doc	d6a795e839f51c1a5aeabf5c10664936ebbef8ea	mister.nofrillspace.com/users/web8_dice/3791/space.gif
Top 11 Aerial Surveillance Devices.doc	f362feedc046899a78c4480c32dda4ea82a3e8c0	intent.nofrillspace.com/users/web11_focus/4307/space.gif
SEMBOYAN_1.doc	f751cdfaef99c6184f45a563f3d81ff1ada25565	www.police28122011.0fees.net/pages/013/space.gif

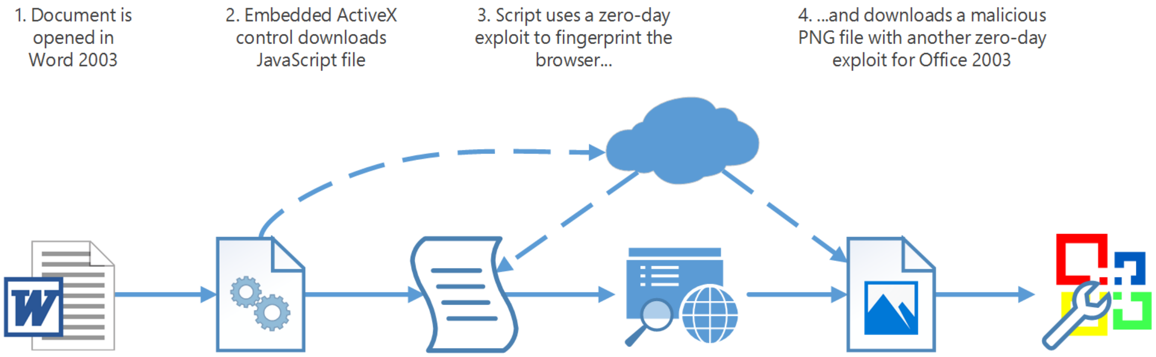
Figure 7. Malicious JavaScript used by PLATINUM to perform fingerprinting on a victim's browser



```
header = "res://";
footer = '/';
0;
jav_arr = new Array();
arr[0] = "C:\\progra~1\\Java\\jre7\\bin\\npjpi170_05.dll";
arr[1] = "C:\\progra~1\\Java\\jre7\\bin\\npjpi170_06.dll";
```

While fingerprinting the versions of the browser plug-ins, the script loads a remotely hosted malicious PNG file that exploited another previously unknown vulnerability (designated CVE-2013-1331), which affected Microsoft Office 2003 SP3.[3] Exploiting the vulnerability resulted in memory corruption, which allowed the attacker to execute remote code on the computer.

Figure 8. An exploit mechanism used by PLATINUM

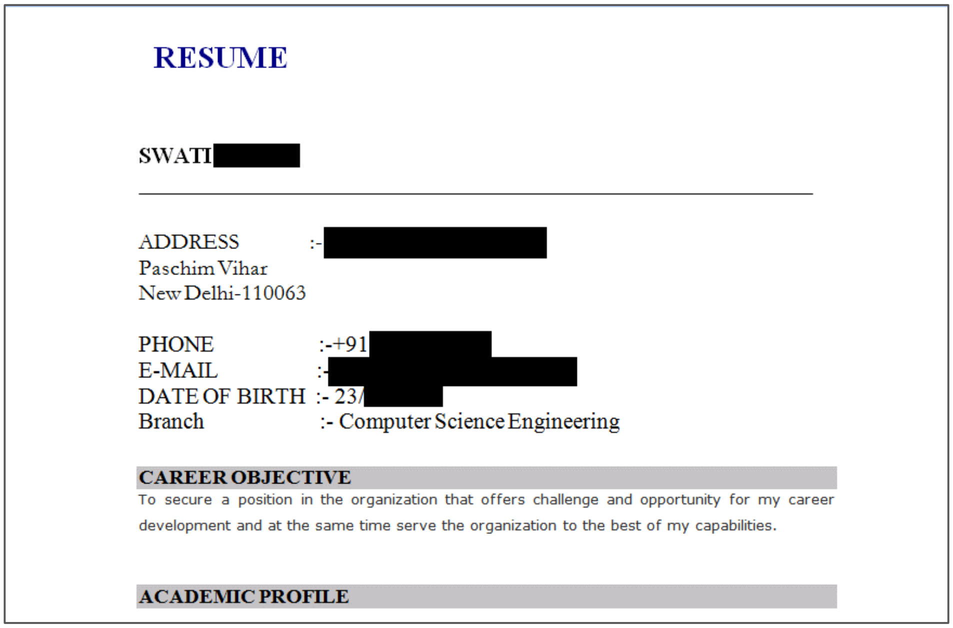

1. Document is opened in Word 2003
2. Embedded ActiveX control downloads JavaScript file
3. Script uses a zero-day exploit to fingerprint the browser…
4. …and downloads a malicious PNG file with another zero-day exploit for Office 2003
W

Another combination of lure documents with the aforementioned embedded ActiveX control was seen along with a Dipsind executable named as pp4x322.dll during a different attack. The unique name of this executable indicated a possible DLL side-loading vulnerability also being used by PLATINUM against PowerPoint 2007.

[3] Microsoft issued Security Bulletin MS13-051 in June 2013 to address the issue.

In another case from August 2015, Microsoft investigated a malicious document (named Resume.docx) that had been uploaded to the VirusTotal malware analysis service.[4] The person who submitted the file did so through an IP address based in India, suggesting that the person or their organization had been targeted by the spear phish document.

Figure 9. A malicious Word document used by PLATINUM to target a victim



RESUME

SWATI [redacted]

ADDRESS :- [redacted]
Paschim Vihar
New Delhi-110063

PHONE	:-+91 [redacted]
E-MAIL	:[redacted]
DATE OF BIRTH	:- 23/[redacted]
Branch	:- Computer Science Engineering

CAREER OBJECTIVE

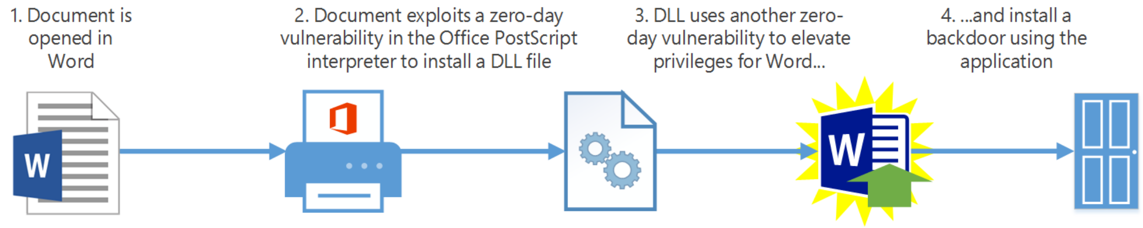
To secure a position in the organization that offers challenge and opportunity for my career development and at the same time serve the organization to the best of my capabilities.

ACADEMIC PROFILE

When the document was opened in Word, it exploited a previously unknown vulnerability in the Microsoft Office PostScript interpreter (designated CVE-2015-2545) that enabled it to execute the attacker's code and drop an attacker-generated malicious DLL onto the computer.[5] The DLL exploited another previously unknown vulnerability (designated CVE-2015-2546) in the Windows kernel, which enabled it to elevate privileges for the Word executable and subsequently install a backdoor through the application.[6] Researching this attack and the malware used therein led Microsoft to discover other instances of PLATINUM attacking users in India around August 2015.

[4] Microsoft thanks FireEye for identifying and reporting this attack.
[5] Microsoft issued Security Bulletin MS15-099 in September 2015 to address the issue. Windows 10 is not affected by the exploit used in this case due to built-in mitigations.
[6] Microsoft issued Security Bulletin MS15-097 in September 2015 to address the issue.

Figure 10. Another exploit mechanism used by PLATINUM


1. Document is opened in Word
2. Document exploits a zero-day vulnerability in the Office PostScript interpreter to install a DLL file
3. DLL uses another zero-day vulnerability to elevate privileges for Word...
4. ...and install a backdoor using the application
W
W

In total, PLATINUM made use of four zero-day exploits during these two attack campaigns (two remote code execution bugs, one privilege escalation, and one information disclosure), showing an ability to spend a non-trivial amount of resources to either acquire professionally written zero-day exploits from unknown markets or research and utilize the zero-day exploits themselves. In both these campaigns, the activity group included remote triggers to deactivate exploitation, with an attempt to conceal the vulnerability and prevent analysis of the attack. The resources required to research and deploy multiple zero-day exploits within the same attack campaign are considerable. Such activity requires a significant amount of investment in research and development, along with the discipline to ensure that the exploits are not used until the appropriate time, and that no one involved with the project leaks them to other parties.

> PLATINUM used four zero-day exploits during these two campaigns.

Technical details

After gaining access to a victim's computer, PLATINUM installs its own custom-built malware to communicate with the compromised computer, issue commands, and move laterally through the network. The broad collection of backdoors and tools, and the differences between them, suggest the involvement of multiple teams or vendors in the development process. This section describes some of the tools used by the group.

Dipsind

PLATINUM uses a number of different custom-developed backdoors to communicate with infected computers. The lack of any significant evidence of shared code between any of these backdoor families is another clue as to the scope of the resources on which the activity group is able to draw, and the precautions the group is willing and able to take to avoid losing its ability to conduct its espionage operations.

The group's most frequently used backdoors belong to a malware family that Microsoft has designated Dipsind, although some variants are detected under different names. Multiple Dipsind variants have been identified, all of which are believed to be used exclusively by PLATINUM.

The first variant, Win32/Dipsind.A!dha, is a lightweight application providing backdoor access to remote attackers. It can be customized for every victim to ensure that it remains undetected in targeted networks. It supports a small set of instructions that allow the attacker to perform basic functions, such as uploading or downloading files and spawning remote shells.

Figure 11. Sample configuration file for Win32/Dipsind.A

```
RunTimeFolderUser =   [……….]
RunTimeFolderAdmin = [%commonfiles%]\System\Network
Provision\
RunTimeFileNameDll = xmlprv.dll
ServiceKeyName = xmlprv
ServiceKeyNameDll = xmlprv

cmdpathAdmin = [%commonfiles%]\System\Network
Provision\Library\
cmdname = wscntfy.exe

slpSite1 = scienceweek.scieron.com
pollSlpTime = 10

pollcommandsite1 = [ AES encrypted ]
pollcommandTime = 10

officeStart = 00:00, officeEnd = 23:59
sat = 1, sun = 1

checkurl1 = http://www.google.com/
```

Each Dipsind file contains an embedded encrypted configuration file that acts as a control for the backdoor. This configuration file also includes the initial command and control (C&C) location the Dipsind backdoor uses in addition to the *pollcommandsite* variable, which references a URL where additional backup C&Cs can be polled. Configurable parameters include instructions on where Dipsind should install a copy of cmd.exe for spawning a remote shell, depending on the user's privileges, and the hours during which the backdoor should function and exfiltrate information. This capability allows the backdoor to confine its activities to normal working hours, making its communications harder to distinguish from normal network traffic.

Dipsind has been observed using a combination of IP addresses and domains for its C&C infrastructure. The domains are a mix of registered domains and free subdomains obtained through dynamic DNS providers. Collected data showed that most victim networks allowed unfiltered access to the dynamic DNS hosts. The hosts and domains are hosted on compromised infrastructure based in several different countries, some within academic institutions. In some cases, the backdoors are configured to connect to IP addresses instead of domain names. These factors make it challenging to locate the activity group's infrastructure.

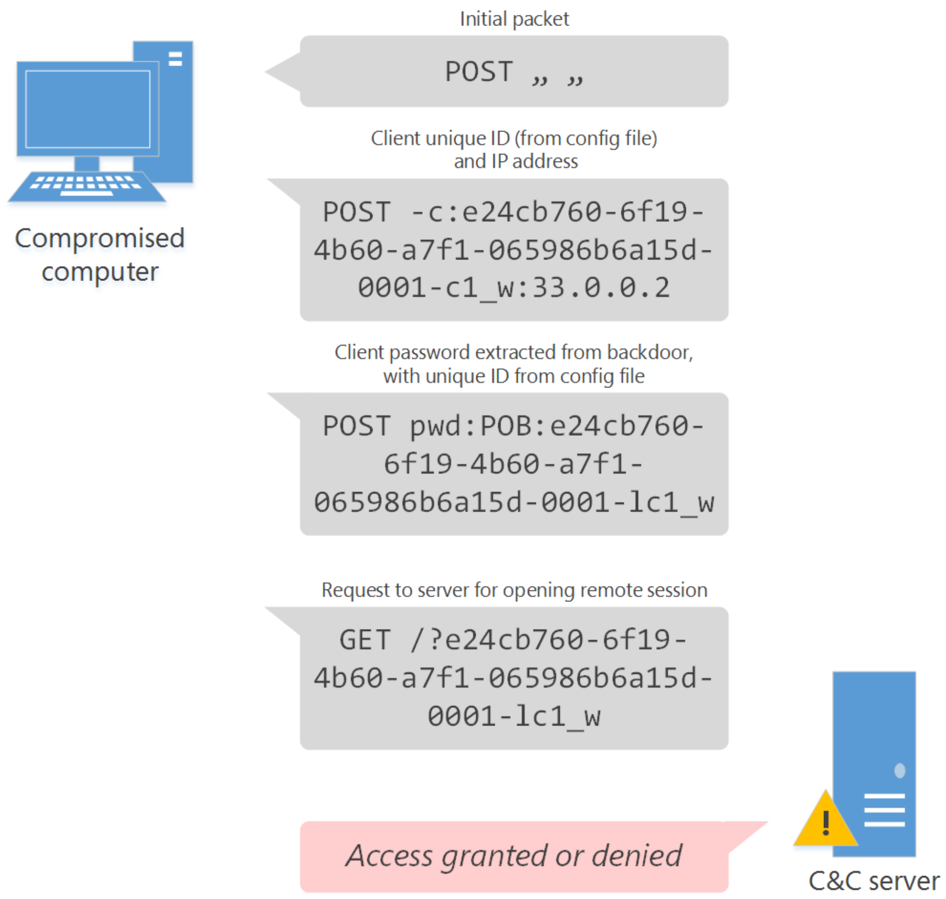
Figure 12 shows a sampling of C&C infrastructure used by PLATINUM between 2009 and 2015.

Figure 12. Some of the domains and addresses used by PLATINUM

Registered domains	Dynamic DNS	Hardcoded IPs
• box62.a-inet.net	• scienceweek.scieron.com	• 200.61.248.8
• eclipse.a-inet.net	• mobileworld.darktech.org	• 209.45.65.163
• joomlastats.a-inet.net	• geocities.efnet.at	• 190.96.47.9
• updates.joomlastats.co.cc	• bpl.blogsite.org	• 192.192.114.1
• server.joomlastats.co.cc	• wiki.servebbs.net	• 61.31.203.98

After Dipsind.A is installed on the victim's computer, it connects to its C&C server for authentication. All network traffic is over HTTP, base64 encoded, with the underlying data encrypted using AES256 in ECB mode. Authentication is a five-step process, as shown in the following figure:

Figure 13. Win32/Dipsind.A initial communication protocol (as decrypted)


Initial packet
Compromised computer
Client unique ID (from config file) and IP address
Client password extracted from backdoor, with unique ID from config file
Request to server for opening remote session
Access granted or denied
C&C server

```
POST „ „
```

```
POST -c:e24cb760-6f19-4b60-a7f1-065986b6a15d-0001-c1_w:33.0.0.2
```

```
POST pwd:POB:e24cb760-6f19-4b60-a7f1-065986b6a15d-0001-lc1_w
```

```
GET /?e24cb760-6f19-4b60-a7f1-065986b6a15d-0001-lc1_w
```

Analysis of several samples of this variant show exactly the same AES key (AOPSH03SK09POKSID7FF674PSLI91965) in use since 2009. The initial HTTP POST made by this backdoor appears as "*ud7LDjtsTHe2tWeC8DYo8A***", which translates to a simple whitespace. This sequence makes a simple network indicator usable by defenders.

A second Dipsind variant registers as a Winlogon Event Notify DLL. This backdoor contains a minimized feature list from the original Dipsind variant, and supports a more limited number of commands. It sets the following registry keys in the HKEY_LOCAL_MACHINE hive for persistence and functionality:

- SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\Cscdll32\Asynchronous
- SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\Cscdll32\DllName

- SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\Cscdll32\Impersonate
- SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\Cscdll32\Startup
- SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\Cscdll32\shutdown
- SOFTWARE\Microsoft\Windows\CurrentVersion\Run\cscdll32

At least two additional minor versions of this variant exist, each of which show improvements in command implementation.

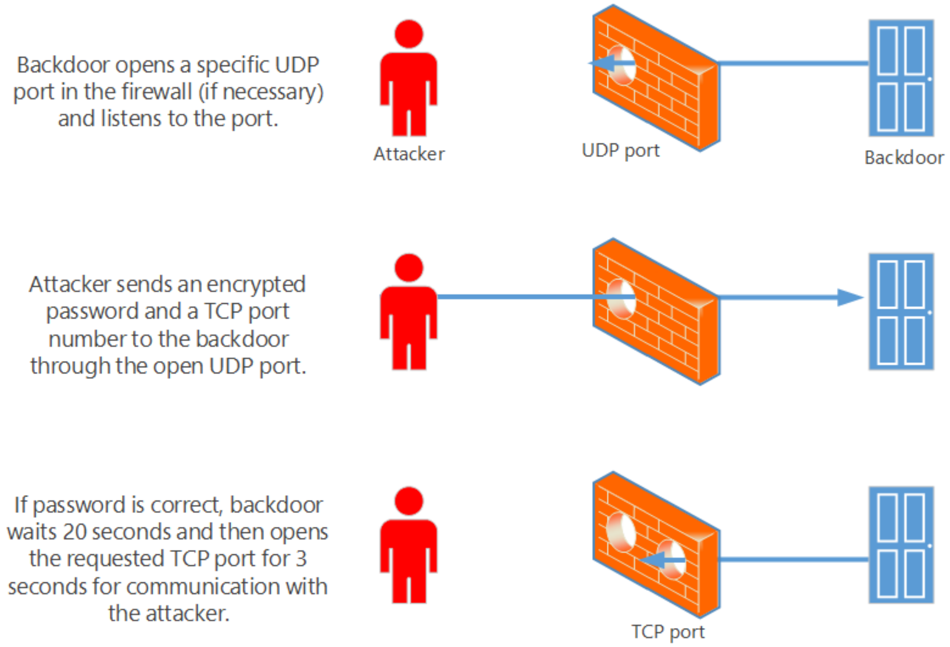
One interesting feature of this variant is the way it implements a mechanism similar to port knocking to allow remote attackers to connect to a compromised computer without leaving any connection open for too long. The sequence of events is as follows:

1. The backdoor is installed via an exploit.
2. The backdoor sets a registry key to open a specific UDP port through the local firewall, if any, and listens to the port for incoming traffic.
3. At a remote location, the attacker executes a tool (called PK2 here, although the actual name of the tool is unknown) using the following parameters:

   ```
   Pk2.exe <IP> <UDP Port> <TCP Port> <Password>
   ```

   where the IP address is that of the computer with the backdoor, the UDP port is the one specified by the backdoor, and the password is a string encrypted by the tool before being sent.
4. The backdoor receives the UDP packets, and then checks to see if the password is valid.
5. If the password is indeed valid, the backdoor will wait for exactly 20 seconds and only then open the PK2 specified TCP port for a window of 3 seconds.

Figure 14. How the Dipsind knocker component communicates with an attacker



Backdoor opens a specific UDP port in the firewall (if necessary) and listens to the port.

Attacker sends an encrypted password and a TCP port number to the backdoor through the open UDP port.

If password is correct, backdoor waits 20 seconds and then opens the requested TCP port for 3 seconds for communication with the attacker.

PK2 is also designed to connect to such open TCP ports and act as a console client for issuing commands to the backdoor. When running PK2 as a console client, the attacker needs to re-enter the password to authenticate a second time against the backdoor, and issue commands such as #sz to upload a file and #rz to download a file. During this research, one such collection of tools was obtained that had the password set to "t@ng0p@ss". All communication used by this backdoor and PK2 is encrypted. If a connection from PK2 is not received within the 3-second window, the TCP port is shut and PK2 would need to reinitialize the port-knocking process.

JPIN

In addition to Dipsind and its variants, PLATINUM uses a few other families of custom-built backdoors within its attack toolset. These families of backdoors are significantly different in their capabilities and have completely different code bases. While one family relies on a small number of supported commands and simple shells, the other delves into more convoluted methods of injections, checks, and supported feature sets.

Microsoft researchers refer to one such set of backdoor variants collectively as JPIN, which is the name of a service it uses when installed. JPIN is a comprehensive tool for executing and extracting information from the compromised computer. There is strong evidence to suggest that the developers of the JPIN and Dipsind code bases were in some way related.

JPIN has its own installer and uninstaller component, which deletes itself when it encounters a version of Windows earlier than Windows XP or finds any of these security-related processes running:

Figure 15. Security-related processes avoided by the JPIN installer

Process	Security product
360tray.exe	360 Safeguard
bdagent.exe	BitDefender
proguard.exe	Process Guard
blackd.exe	BlackICE
blackice.exe	BlackICE
savservice.exe	Sophos Anti-Virus
avp.exe	Kaspersky Anti-Virus
rstray.exe	Rising Anti-virus
cmccore.exe	CMC Antivirus
cmctrayicon.exe	CMC Antivirus
zhudongfangyu.exe	360 Safeguard

After installing the backdoor, the installer deletes itself from the compromised computer.

PLATINUM uses at least three distinct JPIN variants. One variant typically runs with a mutex named hMSVmm and installs itself in the folders %appdata%\Comm\Jpin and %userprofile%\AppData\Resource\Jpin. After it is installed and started, the JPIN service can perform the following tasks, among others:

- Obtain information about the computer, such as operating system version, user name, privileges, disk space, and so on.
- List running services, processes, job IDs, and task IDs.
- Enumerate drives and their types.
- Enumerate registry keys.
- Load a custom keylogger.

- Download files.
- Download and upgrade itself.
- Acquire network information such as DNS, IP, proxies, and so on.
- Exfiltrate information over HTTP GET and POST requests, with the data stored either within the HTTP body or within the URL parameters.
- Lower security settings by tampering with registry keys.
- Inject content into the lsass.exe process, in order to load the keylogger module into lsass and call its exported function.
- Communicate via FTP.
- Send email via SMTP.
- Change permissions on files using the cacls.exe command-line utility.

JPIN can also target mobile suite applications and extract data from them. The backdoor contains code that looks for installed instances of Symbian, BlackBerry, and Windows Phone management applications. If any are found, the backdoor logs sync dates, IMEI data, phone manufacturer and model information, software version date, memory, location, and capacity, among other information.

JPIN can target mobile suite applications and extract data from them.

The second JPIN variant is very similar to the first one. It downloads the backdoor payload from remote locations via the BITS service, using the COM object for BITS. This variant also has its own installer and uninstaller component, which deletes itself when it encounters a version of Windows earlier than Windows XP or finds any of the processes listed in Figure 15 running.

The third known variant does not check for the processes listed in Figure 15. It uses an installer component that includes the backdoor as payload disguised as a bitmap within its resource section. The payload is in an encrypted and compressed form, disguised to avoid any suspicion from security solutions. This variant has been seen installing itself into the following file system paths:

- %appdata%\Java\support
- %appdata%\support
- %userprofile%\AppData\Local\Java\Support

- %userprofile%\AppData\Local\Support

adbupd

Another backdoor used by PLATINUM is very similar to the Dipsind family. It is informally referred to internally at Microsoft as adbupd, which is the name of the service under which it is installed. Salient features of this backdoor include the following:

- It tries to install itself under several different names within the Program Files directory.
- It has the ability to support plug-ins to modularize functionality.
- It contains a copy of the OpenSSL library to support encryption when sending or receiving data.
- It contains functionality to run a copy of cmd.exe.
- The configuration file is very similar to the original Dipsind family.
- This backdoor class uses multiple methods of achieving persistence, one of which is using WMI /MOF compiled scripts, such as the one shown in Figure 16.

Figure 16. WMI script used by the adpupd backdoor to achieve persistence

```
#pragma namespace("\\\\.\\ROOT\\cimv2")
instance of __Win32Provider as $P
{
    Name = "adbupdConsumer";
    ClsId = "{74ba9ce4-fbf1-4097-32b8-34f446f037d8}";
    HostingModel = "LocalSystemHost";
};
instance of __EventConsumerProviderRegistration
{
    Provider = $P;
    ConsumerClassNames = {"adbupdConsumer"};
};
class adbupdConsumer : __EventConsumer
{
    [key] string Mode;
};
instance of adbupdConsumer as $CONSMR
{
    Mode = "persistent";
};
```

```
instance of __EventFilter as $FLT
{
    Name = "adbupdFilter";
    Query = "SELECT * FROM __InstanceCreationEvent WHERE
TargetInstance ISA \"Win32_NTLogEvent\"";
    QueryLanguage = "WQL";
};
instance of __FilterToConsumerBinding as $B
{
    Consumer = $CONSMR;
    Filter = $FLT;
};
```

Keyloggers

The PLATINUM group has written a few different versions of keyloggers that perform their functions in different ways, most likely to take advantage of different weaknesses in victims' computing environments. The keyloggers can be broadly classified into two groups: those that log keystrokes through raw device input, and user mode keyloggers that use Windows hook interfaces to gather information. In particular, this second group also has the capability of dumping users' credentials using the same technique employed by Mimikatz. Both groups can set permissions on specific files to Everyone, and work in tandem with the PLATINUM backdoors.

Hot patcher

One of PLATINUM's most recent and interesting tools is meant to inject code into processes using a variety of injection techniques. In addition to using several publicly known injection methods to perform this task, it also takes advantage of an obscure operating system feature known as *hot patching*.

Hot patching is an operating system-supported feature for installing updates without having to reboot or restart a process. At a high level, hot patching can transparently apply patches to executables and DLLs in actively running processes, which does not happen with traditional methods of code injection such as CreateRemoteThread or WriteProcessMemory. Instead, the kernel is instructed to perform the injection by invoking NtSetSystemInformation (with an appropriate SystemInformationClass) to apply the patch. The information about the patch is delivered via a specially crafted DLL that is loaded into the target process.

The hot patching feature originally shipped with Windows Server 2003 and was used to ship 10 patches to Windows Server 2003. It was removed in Windows 8 and has not been included in subsequent releases of Windows. PLATINUM appears to believe that enough of their targeted users continue to run the earlier versions of Windows to make the technique a useful tool, at least until early 2017 (see page 22).

The technique PLATINUM uses to inject code via hot patching was first documented by security researchers in 2013.[7] Administrator permissions are required for hot patching, and the technique used by PLATINUM does not attempt to evade this requirement through exploitation. Rather, the component's use of the hot patching feature appears to be a way to avoid being detected, because many antivirus solutions monitor non-system processes for the regular injection methods such as CreateRemoteThread. If the tool fails to inject code using hot patching, it reverts to attempting the other more common code injection techniques into common Windows processes, primarily targeting winlogon.exe, lsass.exe, and svchost.exe:

- CreateRemoteThread
- NtQueueApcThread
- RtlCreateUserThread
- NtCreateThreadEx

The hot patching component performs the following steps:

1. It patches the loader with a proper hot patch to treat injected DLLs with execute page permissions. This step is required for DLLs loaded from memory (in an attempt to further conceal the malicious code).

2. The backdoor is injected into svchost using the hot patch API. Patching the loader is done by creating a section named \knowndlls\mstbl.dll. This DLL does not reside on disk, but is rather treated as a cached DLL by the session manager. It then proceeds to write a PE file within that section.

3. The PE file will have one section (.hotp1) with the hot patch header structure. This structure contains all the information necessary to perform the patching

[7] Alex Ionescu, "Hotpatching the Hotpatcher: Stealth File-less DLL Injection," SyScan 2013, https://www.yumpu.com/en/document/view/14255220/alexsyscan13/23.

of function ntdll!LdrpMapViewOfSection, which will cause the loader to treat created sections as PAGE_EXECUTE_READWRITE instead of PAGE_READWRITE. The patch is successfully applied by invoking NtSetSystemInformation.

4. After the memory permission issue is solved, the injector proceeds to inject the malicious DLL into svchost. Again, it creates a (now executable) section named knowndlls\fgrps.dll and invokes NtSetSystemInformation, which causes the final payload to be loaded and executed within the target process (svchost).

5. The malicious hot patching component appears to have an expiration date of January 15, 2017. After that date, the DLL will no longer perform the injection, but rather execute another PLATINUM implant (C:\Program Files\Windows Journal\Templates\Cpl\jnwmon.exe –ua), which may be related to an uninstall routine. (The component has not been observed in use since March 9, 2016, which may indicate that PLATINUM has chosen to stop using it earlier than the configured expiration date.)

Miscellaneous

Finally, the PLATINUM group also uses small single-purpose applications that duplicate some of the functionality of the backdoors. A couple of examples are:

- A stand-alone persistence tool that takes other files as input and ensures persistence across reboots.
- A stand-alone loader that runs another executable. It has some exported functions whose names can be used in DLL files installed as LSA password filters, but such functions are basically empty and there is no known evidence that this tool was ever used in this way. On the whole, this DLL looks like a test, suggesting that the attackers may have researched and possibly implemented variants of their malware that can be installed as LSA password filters.

Exploit (CVE-2015-2545)

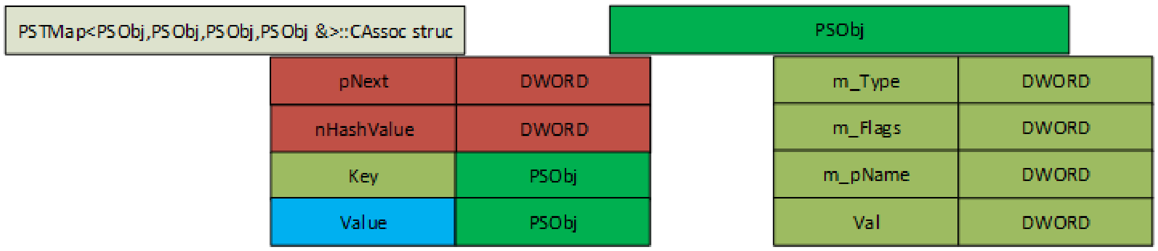
CVE-2015-2545 is a use-after-free vulnerability in the embedded PostScript filter of Microsoft Office.[8] The exploit was crafted in PostScript and is able to bypass

[8] Microsoft issued Security Bulletin MS15-099 in September 2015 to address the issue.

Address Space Layout Randomization (ASLR) and Data Execution Prevention (DEP).

This vulnerability allows the attacker to forge a CAssoc structure, shown in Figure 17, and so also indirectly the PSObjs in the structure. The PostScript interpreter deciphers the value field (Val) based on the type field (m_type), which are under complete control of the attacker. Having developed this technique, the attacker will craft and use a combination of file, string, and integer objects to gain a reliable arbitrary code execution.

Figure 17. Memory layout of CSssoc structure and its embedded PSObjs



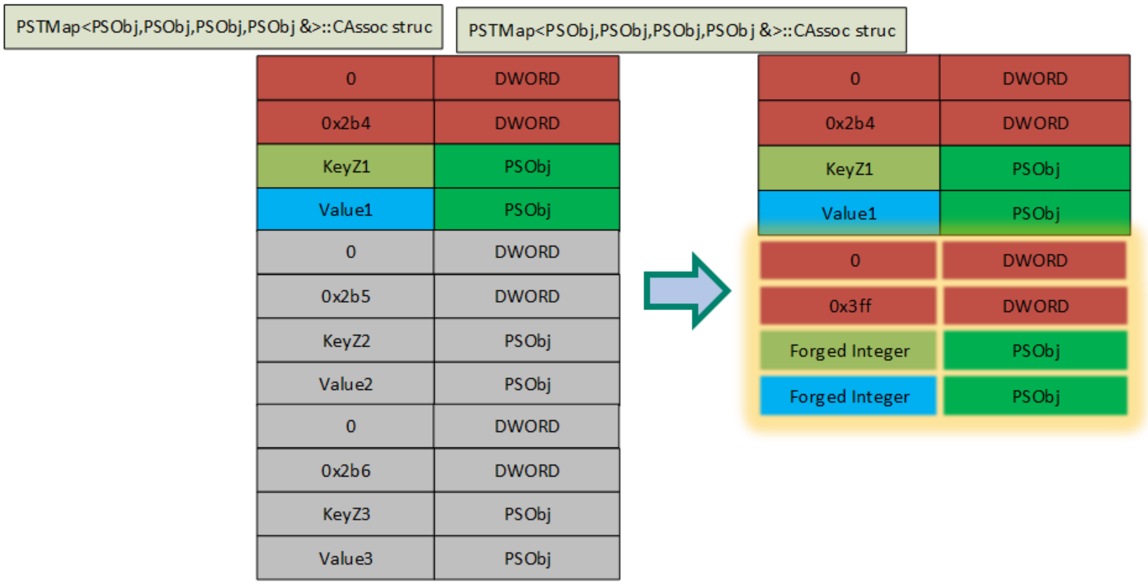
PSTMap<PSObj,PSObj,PSObj,PSObj &>::CAssoc struc	
pNext	DWORD
nHashValue	DWORD
Key	PSObj
Value	PSObj

PSObj	
m_Type	DWORD
m_Flags	DWORD
m_pName	DWORD
Val	DWORD

Root cause: The attacker defined in PostScript a dictionary with three elements, which leads to an allocation of three CAssoc structures in PSTMap.

Within a Forall loop, the last two elements are undefined and a string is initialized. The PostScript statement results in a deallocation of the last two CAssoc structures and the string gets allocated in the previously freed memory address. The PostScript-put operand is used to fill the string with data to mimic a CAssoc structure. By setting the hash table index to 0x3ff, the loop will exit because the hash table at that time has a max-size of 0x400. Upon exiting the loop, a reference will be returned to the secondary element, which is the forged structure.

Figure 18. Reusage of deallocated memory by a forged CAssoc structure



PSTMap<PSObj,PSObj,PSObj,PSObj &>::CAssoc struc

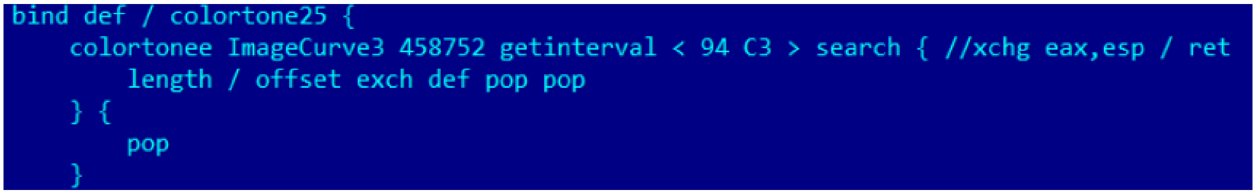
0	DWORD
0x2b4	DWORD
KeyZ1	PSObj
Value1	PSObj
0	DWORD
0x2b5	DWORD
KeyZ2	PSObj
Value2	PSObj
0	DWORD
0x2b6	DWORD
KeyZ3	PSObj
Value3	PSObj

PSTMap<PSObj,PSObj,PSObj,PSObj &>::CAssoc struc

0	DWORD
0x2b4	DWORD
KeyZ1	PSObj
Value1	PSObj
0	DWORD
0x3ff	DWORD
Forged Integer	PSObj
Forged Integer	PSObj

Acquire full memory RW access: The described method is used to craft a PSString object in which the length of the string is set to a maximum value. As a result, the exploit can use PostScript methods to search for ROP gadgets to dynamically assemble a ROP shellcode.

Figure 19. Getinterval method of PSString is used to find ROP gadgets



```
bind def / colortone25 {
    colortonee ImageCurve3 458752 getinterval < 94 C3 > search { //xchg eax,esp / ret
        length / offset exch def pop pop
    } {
        pop
    }
```

The purpose of this approach is to call VirtualProtect to set the pages of the second-stage shellcode as executable. As a result, DEP and ASLR are bypassed.

Arbitrary code execution: To redirect code execution to the ROP chain, the exploit crafts a PSFile Object in which the vtable is controlled by the attacker. By calling the bytesavailable method within the PostScript code, arbitrary code execution is achieved.

Identity

Although the exact identity of PLATINUM remains unknown, the technical indicators observed so far can help create a profile of the attacker.

- Usage of multiple backdoors. The different backdoors written by or for the group indicate a considerable investment over time. Research indicates that PLATINUM has used multiple backdoors concurrently at times, which could represent either multiple teams within the activity group performing different campaigns or different versions of the tools being used against varying victim networks.
- Zero-day exploits. PLATINUM has used several zero-day exploits against its victims. Regardless of whether PLATINUM researched these exploits themselves or purchased them from independent researchers, the monetary investment required to collect and deploy zero-day exploits at this level is considerable.
- Victim geography. More often than not, research into targeted attacks shows activity groups becoming opportunistic and attacking topical targets; that is, targets considered valuable based on the geopolitical events of the year. PLATINUM has consistently targeted victims within a small set of countries in South and Southeast Asia. In addition, the victims are consistently associated with a small set of entities that are directly or indirectly connected to governments.

> The monetary investment required to collect and deploy zero-day exploits at this level is considerable.

- Tools. Some of the tools used by PLATINUM, such as the port-knocking backdoor, show signs of organized thinking. PLATINUM has developed or commissioned a number of custom tools to provide the group with access to victim resources. This behavior exhibits PLATINUM's ability to adapt to victim networks, which is further evidence of the group's considerable resources for development and maintenance.

  Any of these traits by themselves could be the work of a single resourceful attacker or a small group of like-minded individuals, but the presence of all of them is a clear indication of a well-resourced, focused, and disciplined group of attackers vying for information from government-related entities.

Guidance

PLATINUM is an extremely difficult adversary for targeted organizations to defend against. It possesses a wide range of technical exploitation capabilities, significant resources for researching or purchasing complicated zero-day exploits, the ability to sustain persistence across victim networks for years, and the manpower to develop and maintain a large number of tools to use within unique victim networks. Their ability to research their victims prior to targeting them, along with the capability to architect exploits that only work once or for a short period of time, makes it very difficult to investigate or track their activities. That said, there are steps that organizations can take to reduce the likelihood of PLATINUM conducting successful attacks against their employees and networks.

- Take advantage of native mitigations built into Windows 10. Newer versions of Windows include critical mitigations that render some of PLATINUM's exploits ineffective when deployed. For example, the summer 2015 attack that used the unusual "resumé" would not have been successful on Windows 10 as-is because of the presence of the Supervisor Mode Execution Prevention (SMEP) mitigation, even without the latest security updates installed. Even if CVE-2015-2546 affected Windows 10, the exploitation would have required much more technical prowess to succeed; ultimately, SMEP makes it more difficult for attackers. The hooking and in-memory patching techniques used by the malicious hot patcher component are also not effective against newer versions of Windows.
- Apply all security updates as soon as they become available. Microsoft deeply researches each security issue, proactively addresses the flaw, and mitigates the attack surface around the affected component(s). For example, one zero-day exploit (CVE-2015-2545) used by PLATINUM was addressed immediately in September 2015. Subsequently, in November, Microsoft also released a proactive security update for the same component that ended up mitigating other exploits surfacing in-the-wild after the first attack. Customers who applied the security updates in November without delay would have been protected against the second wave of exploits. Such measures of hardening the underlying application happen often. MS09-017 is yet another example, in which installation of newly available security updates significantly reduced the attack surface.
- Consider disabling features, such as EPS or macros, in powerful products like Microsoft Office by using Group Policy. Not all organizations find the need to enable all features. For example, in the PLATINUM attack campaign that used CVE-2015-2545, a network in which Office EPS was disabled would not have been affected.
- Enterprise networks should segregate high business impact (HBI) data-holding segments from Internet-connected networks. Sharing of removable

media between these air-gapped networks should be strictly enforced. In the case of PLATINUM, such a network architecture would prevent targeted users from accessing third-party email services and thereby granting attackers access to sensitive segments of the organizational network.

- Conduct enterprise software security awareness training, and build awareness of malware prevention. PLATINUM may have used zero-day flaws to compromise victim computers, but doing so required action by the user, who either clicked a link in an email or opened an attachment to allow the attacker to take control of their computer. Security training can raise awareness and reduce the risk associated with this attack vector.

> Apply all security updates as soon as they become available.

- Institute a strong network firewall and proxy. Many tools used by attackers are not compatible with network proxies. In the case of PLATINUM's version of port knocking, the opening of a UDP port would have been rendered moot if a network firewall was blocking access for inbound packets to the host's open port.
- Enterprise networks should consider blocking certain types of websites that don't serve the interest of the business. PLATINUM makes extensive use of C&Cs that use dynamic DNS hosts. Although such free services can be very useful at a personal level, blocking access to such hosts at a local DNS server can minimize post-compromise activity.
- Prepare your network to be forensically ready, so that you can achieve containment and recovery if a compromise occurs. A forensically ready network that records authentications, password changes, and other significant network events can help identify affected systems quickly.
- Make sure that your organization's Internet-facing assets are always running up-to-date applications and security updates, and that they are regularly audited for suspicious files and activity. A number of researched PLATINUM victims had their public-facing infrastructure compromised through previously unknown flaws.

Detection indicators

Figure 20 consists of detection rules for a number of PLATINUM malware samples to be used with YARA (https://plusvic.github.io/yara/), an open source pattern matching tool for malware detection.

Figure 20. Detection indicators for PLATINUM malware

```
rule Trojan_Win32_PlaSrv : Platinum
{
  meta:
    author = "Microsoft"
    description = "Hotpatching Injector"
    original_sample_sha1 =
"ff7f949da665ba8ce9fb01da357b51415634eaad"
    unpacked_sample_sha1 =
"dff2fee984ba9f5a8f5d97582c83fca4fa1fe131"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $Section_name = ".hotp1"
    $offset_x59 = { C7 80 64 01 00 00 00 00 01 00 }

  condition:
    $Section_name and $offset_x59
}

rule Trojan_Win32_Platual : Platinum
{
  meta:
    author = "Microsoft"
    description = "Installer component"
    original_sample_sha1 =
"e0ac2ae221328313a7eee33e9be0924c46e2beb9"
    unpacked_sample_sha1 =
"ccaf36c2d02c3c5ca24eeeb7b1eae7742a23a86a"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $class_name = "AVCObfuscation"
    $scrambled_dir = { A8 8B B8 E3 B1 D7 FE 85 51 32 3E C0 F1 B7
73 99 }

  condition:
    $class_name and $scrambled_dir
}

rule Trojan_Win32_Plaplex : Platinum
{
  meta:
    author = "Microsoft"
    description = "Variant of the JPin backdoor"
    original_sample_sha1 =
"ca3bda30a3cdc15afb78e54fa1bbb9300d268d66"
    unpacked_sample_sha1 =
"2fe3c80e98bbb0cf5a0c4da286cd48ec78130a24"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $class_name1 = "AVCObfuscation"
    $class_name2 = "AVCSetiriControl"

  condition:
    $class_name1 and $class_name2
}

rule Trojan_Win32_Dipsind_B : Platinum
{
  meta:
    author = "Microsoft"
    description = "Dipsind Family"
    sample_sha1 = "09e0dfbb5543c708c0dd6a89fd22bbb96dc4ca1c"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $frg1 = {8D 90 04 01 00 00 33 C0 F2 AE F7 D1 2B F9 8B C1 8B F7
8B FA C1 E9 02 F3 A5 8B C8 83 E1 03 F3 A4 8B 4D EC 8B 15 ?? ?? ??
?? 89 91 ?? 07 00 00 }
    $frg2 = {68 A1 86 01 00 C1 E9 02 F3 AB 8B CA 83 E1 03 F3 AA}
    $frg3 = {C0 E8 07 D0 E1 0A C1 8A C8 32 D0 C0 E9 07 D0 E0 0A C8
32 CA 80 F1 63}

  condition:
    $frg1 and $frg2 and $frg3
}
rule Trojan_Win32_PlaKeylog_B : Platinum
{
  meta:
    author = "Microsoft"
    description = "Keylogger component"
    original_sample_sha1 =
"0096a3e0c97b85ca75164f48230ae530c94a2b77"
```

```
    unpacked_sample_sha1 =
"6a1412daaa9bdc553689537df0a004d44f8a45fd"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $hook = {C6 06 FF 46 C6 06 25}
    $dasm_engine = {80 C9 10 88 0E 8A CA 80 E1 07 43 88 56 03 80
F9 05}

  condition:
    $hook and $dasm_engine
}
rule Trojan_Win32_Adupib : Platinum
{
  meta:
    author = "Microsoft"
    description = "Adupib SSL Backdoor"
    original_sample_sha1 =
"d3ad0933e1b114b14c2b3a2c59d7f8a95ea0bcbd"
    unpacked_sample_sha1 =
"a80051d5ae124fd9e5cc03e699dd91c2b373978b"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = "POLL_RATE"
    $str2 = "OP_TIME(end hour)"
    $str3 = "%d:TCP:*:Enabled"
    $str4 = "%s[PwFF_cfg%d]"
    $str5 = "Fake_GetDlgItemTextW: ***value***="

  condition:
    $str1 and $str2 and $str3 and $str4 and $str5
}
rule Trojan_Win32_PlaLsaLog : Platinum
{
  meta:
    author = "Microsoft"
    description = "Loader / possible incomplete LSA Password
Filter"
    original_sample_sha1 =
"fa087986697e4117c394c9a58cb9f316b2d9f7d8"
    unpacked_sample_sha1 =
"29cb81dbe491143b2f8b67beaeae6557d8944ab4"
```

```
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = {8A 1C 01 32 DA 88 1C 01 8B 74 24 0C 41 3B CE 7C EF 5B
5F C6 04 01 00 5E 81 C4 04 01 00 00 C3}
    $str2 = "PasswordChangeNotify"

  condition:
    $str1 and $str2
}
rule Trojan_Win32_Plagon : Platinum
{
  meta:
    author = "Microsoft"
    description = "Dipsind variant"
    original_sample_sha1 =
"48b89f61d58b57dba6a0ca857bce97bab636af65"
    unpacked_sample_sha1 =
"6dccf88d89ad7b8611b1bc2e9fb8baea41bdb65a"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"

  strings:
    $str1 = "VPLRXZHTU"
    $str2 = {64 6F 67 32 6A 7E 6C}
    $str3 = "Dqpqftk(Wou\"Isztk)"
    $str4 = "StartThreadAtWinLogon"


  condition:
    $str1 and $str2 and $str3 and $str4
}
rule Trojan_Win32_Plakelog : Platinum
{
  meta:
    author = "Microsoft"
    description = "Raw-input based keylogger"
    original_sample_sha1 =
"3907a9e41df805f912f821a47031164b6636bd04"
    unpacked_sample_sha1 =
"960feeb15a0939ec0b53dcb6815adbf7ac1e7bb2"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"

  strings:
    $str1 = "<0x02>" wide
    $str2 = "[CTR-BRK]" wide
    $str3 = "[/WIN]" wide
    $str4 = {8A 16 8A 18 32 DA 46 88 18 8B 15 08 E6 42 00 40 41 3B
CA 72 EB 5E 5B}

  condition:
    $str1 and $str2 and $str3 and $str4
}
rule Trojan_Win32_Plainst : Platinum
{
  meta:
    author = "Microsoft"
    description = "Installer component"
    original_sample_sha1 =
"99c08d31af211a0e17f92dd312ec7ca2b9469ecb"
    unpacked_sample_sha1 =
"dcb6cf7cf7c8fdfc89656a042f81136bda354ba6"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = {66 8B 14 4D 18 50 01 10 8B 45 08 66 33 14 70 46 66 89
54 77 FE 66 83 7C 77 FE 00 75 B7 8B 4D FC 89 41 08 8D 04 36 89 41
0C 89 79 04}
    $str2 = {4b D3 91 49 A1 80 91 42 83 B6 33 28 36 6B 90 97}

  condition:
     $str1 and $str2
}
rule Trojan_Win32_Plagicom : Platinum
{
  meta:
    author = "Microsoft"
    description = "Installer component"
    original_sample_sha1 =
"99dcb148b053f4cef6df5fa1ec5d33971a58bd1e"
    unpacked_sample_sha1 =
"c1c950bc6a2ad67488e675da4dfc8916831239a7"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = {C6 44 24 ?? 68 C6 44 24 ?? 4D C6 44 24 ?? 53 C6 44 24
?? 56 C6 44 24 ?? 00}
    $str2 = "OUEMM/EMM"
    $str3 = {85 C9 7E 08 FE 0C 10 40 3B C1 7C F8 C3}

  condition:
    $str1 and $str2 and $str3
}
rule Trojan_Win32_Plaklog : Platinum
{
  meta:
    author = "Microsoft"
    description = "Hook-based keylogger"
    original_sample_sha1 =
"831a5a29d47ab85ee3216d4e75f18d93641a9819"
    unpacked_sample_sha1 =
"e18750207ddbd939975466a0e01bd84e75327dda"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"

  strings:
    $str1 = "++[%s^^unknown^^%s]++"
    $str2 = "vtfs43/emm"
    $str3 = {33 C9 39 4C 24 08 7E 10 8B 44 24 04 03 C1 80 00 08 41
3B 4C 24 08 7C F0 C3}

  condition:
    $str1 and $str2 and $str3
}
rule Trojan_Win32_Plapiio : Platinum
{
  meta:
    author = "Microsoft"
    description = "JPin backdoor"
    original_sample_sha1 =
"3119de80088c52bd8097394092847cd984606c88"
    unpacked_sample_sha1 =
"3acb8fe2a5eb3478b4553907a571b6614eb5455c"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = "ServiceMain"
    $str2 = "Startup"
    $str3 = {C6 45 ?? 68 C6 45 ?? 4D C6 45 ?? 53 C6 45 ?? 56 C6 45
?? 6D C6 45 ?? 6D}

  condition:
    $str1 and $str2 and $str3
}
rule Trojan_Win32_Plabit : Platinum
{
  meta:
    author = "Microsoft"
    description = "Installer component"
    sample_sha1 = "6d1169775a552230302131f9385135d385efd166"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = {4b D3 91 49 A1 80 91 42 83 B6 33 28 36 6B 90 97}
    $str2 = "GetInstanceW"
    $str3 = {8B D0 83 E2 1F 8A 14 0A 30 14 30 40 3B 44 24 04 72
EE}

  condition:
    $str1 and $str2 and $str3
}
rule Trojan_Win32_Placisc2 : Platinum
{
  meta:
    author = "Microsoft"
    description = "Dipsind variant"
    original_sample_sha1 =
"bf944eb70a382bd77ee5b47548ea9a4969de0527"
    unpacked_sample_sha1 =
"d807648ddecc4572c7b04405f496d25700e0be6e"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = {76 16 8B D0 83 E2 07 8A 4C 14 24 8A 14 18 32 D1 88 14
18 40 3B C7 72 EA }
    $str2 = "VPLRXZHTU"
    $str3 = "%d) Command:%s"
    $str4 = {0D 0A 2D 2D 2D 2D 2D 09 2D 2D 2D 2D 2D 2D 0D 0A}

  condition:
    $str1 and $str2 and $str3 and $str4
}
rule Trojan_Win32_Placisc3 : Platinum
{
  meta:
    author = "Microsoft"
    description = "Dipsind variant"
    original_sample_sha1 =
"1b542dd0dacfcd4200879221709f5fa9683cdcda"
    unpacked_sample_sha1 =
"bbd4992ee3f3a3267732151636359cf94fb4575d"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = {BA 6E 00 00 00 66 89 95 ?? ?? FF FF B8 73 00 00 00 66
89 85 ?? ?? FF FF B9 64 00 00 00 66 89 8D ?? ?? FF FF BA 65 00 00
00 66 89 95 ?? ?? FF FF B8 6C 00 00 00}
    $str2 = "VPLRXZHTU"
    $str3 = {8B 44 24 ?? 8A 04 01 41 32 C2 3B CF 7C F2 88 03}

  condition:
    $str1 and $str2 and $str3
}
rule Trojan_Win32_Placisc4 : Platinum
{
  meta:
    author = "Microsoft"
    description = "Installer for Dipsind variant"
    original_sample_sha1 =
"3d17828632e8ff1560f6094703ece5433bc69586"
    unpacked_sample_sha1 =
"2abb8e1e9cac24be474e4955c63108ff86d1a034"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = {8D 71 01 8B C6 99 BB 0A 00 00 00 F7 FB 0F BE D2 0F BE
04 39 2B C2 88 04 39 84 C0 74 0A}
    $str2 = {6A 04 68 00 20 00 00 68 00 00 40 00 6A 00 FF D5}
    $str3 = {C6 44 24 ?? 64 C6 44 24 ?? 6F C6 44 24 ?? 67 C6 44 24
?? 32 C6 44 24 ?? 6A}

  condition:
    $str1 and $str2 and $str3
```

```
}
rule Trojan_Win32_Plakpers : Platinum
{
  meta:
    author = "Microsoft"
    description = "Injector / loader component"
    original_sample_sha1 =
"fa083d744d278c6f4865f095cfd2feabee558056"
    unpacked_sample_sha1 =
"3a678b5c9c46b5b87bfcb18306ed50fadfc6372e"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = "MyFileMappingObject"
    $str2 = "[%.3u]  %s  %s  %s [%s:" wide
    $str3 = "%s\\{%s}\\%s" wide

  condition:
    $str1 and $str2 and $str3
}
rule Trojan_Win32_Plainst2 : Platinum
{
  meta:
    author = "Microsoft"
    description = "Zc tool"
    original_sample_sha1 =
"3f2ce812c38ff5ac3d813394291a5867e2cddcf2"
    unpacked_sample_sha1 =
"88ff852b1b8077ad5a19cc438afb2402462fbd1a"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = "Connected [%s:%d]..."
    $str2 = "reuse possible: %c"
    $str3 = "] => %d%%\x0a"


  condition:
    $str1 and $str2 and $str3
}
rule Trojan_Win32_Plakpeer : Platinum
{
  meta:
    author = "Microsoft"
    description = "Zc tool v2"
    original_sample_sha1 =
"2155c20483528377b5e3fde004bb604198463d29"
    unpacked_sample_sha1 =
"dc991ef598825daabd9e70bac92c79154363bab2"
    activity_group = "Platinum"
    version = "1.0"
    last_modified = "2016-04-12"
  strings:
    $str1 = "@@E0020(%d)" wide
    $str2 =
/exit.{0,3}@exit.{0,3}new.{0,3}query.{0,3}rcz.{0,3}scz/ wide
    $str3 = "---###---" wide
    $str4 = "---@@@---" wide


  condition:
    $str1 and $str2 and $str3 and $str4
}
```

Protecting identities in the cloud: Mitigating password attacks

Azure Active Directory Team

Protecting identities is foundational to how Microsoft protects its customers' user accounts, devices, apps, and data. In a mobile-first, cloud-first world, identity and access management is a critical capability that enables secure communication, collaboration, and information and resource sharing. Identity is the key to controlling access to services, devices, and information, as well as to providing visibility and insight into where and how data is being used.

Account compromise is among the most serious cyber risks that consumers and organizations face. For consumers, a compromised account could provide an attacker with access to their personal information, pictures, videos, financial information, and access to their social networks, which could lead to identity theft. For organizations, a single compromised identity provides attackers an entry point that can be used to further compromise the organization's environment.

Microsoft is an identity and access provider for both consumers and enterprise users, spanning both on-premises infrastructures and cloud services. The scale of Microsoft cloud services is such that tremendous insights are gained when attackers seek to compromise user accounts of consumers and enterprises. Microsoft uses these insights to provide world-class protection.

This section of the *Microsoft Security Intelligence Report* focuses on some of the things that Microsoft does to prevent account compromise, even in cases where attackers have possession of valid account credentials. Two sources provided the data for this section: Microsoft Accounts, which are primarily used by consumers, and Azure Active Directory, which is primarily used by organizations such as enterprise customers and schools.

Microsoft Account

A Microsoft Account, formerly called Windows Live ID, is the combination of a user name and a password that customers use to sign into services such as Bing, Outlook.com, OneDrive, Windows Phone, Skype, Xbox LIVE, Windows 8.1, Windows 10, and others. When a Microsoft Account is set up across a user's devices and services, access to contacts, documents, photos, and settings follow them on whatever devices they use, including Windows PCs, tablets, phones, Xbox consoles, Macs, iPhones, or Android devices.

Azure Active Directory

Azure Active Directory provides single sign-on to thousands of cloud (SaaS) apps such as Office 365, Workday, Box, Google Apps and more. It also provides access to on-premises web apps. Azure Active Directory features multi-factor authentication (MFA), access control based on device health, user location, identity, and risk, in addition to holistic security reports, audits, and alerts.

The following statistics describe how different services are being used by organizations, which helps put the scale of Azure Active Directory into context. These statistics were obtained at the end of the reporting period for this volume of the *Security Intelligence Report*, December 31, 2015:

- 95 percent of all organizations and 90 percent of the world's 2,000 largest organizations use Active Directory on-premises.
- There were *8.24 million tenants* in Azure Active Directory comprising *more than 550 million users*.
- Most of these tenants were small businesses with fewer than 500 user accounts and were not synchronizing from an on-premises instantiation of Active Directory.
- A minority of these 8.24 million tenants had more than 500 user accounts, but because they are comparatively large, they accounted for 91 percent of all the identities in Azure Active Directory.
- At the time these statistics were collected, Azure Active Directory was averaging *more than 1.3 billion authentications per day*.

Scale + intelligence = Better protection

Across the aforementioned services and devices, Microsoft processes more than *13 billion* requests from hundreds of millions of users *every day*.

This massive scale enables Microsoft to gather an enormous amount of intelligence on malicious behavior, which helps prevent the compromise of Microsoft Accounts and block the use of leaked or stolen credentials. These efforts help protect consumers who use Microsoft Accounts as well as organizations and enterprise customers.

Mitigating password attacks

Ever since passwords were first implemented in computer technology, attackers have used password-based attacks in their attempts to compromise user accounts. Their efforts have been directed at networks, websites, devices, and, more recently, cloud services. Over time, attackers have developed extremely sophisticated means of compromising accounts; phishing, brute force, social engineering, and other types of attacks are used to obtain user passwords. When breaches occur on websites and databases across the industry, the credentials that are harvested from such attacks are used in future attacks. They are sometimes compiled into massive lists of leaked and stolen passwords (some of these lists have been found with more than a billion passwords) that are sold, traded, and shared on the Internet. Because password reuse across accounts is common, even a single leaked password can provide an attacker with access to every one of a user's accounts.

> To prevent and mitigate password attacks, Microsoft uses a multi-layered system of protection mechanisms.

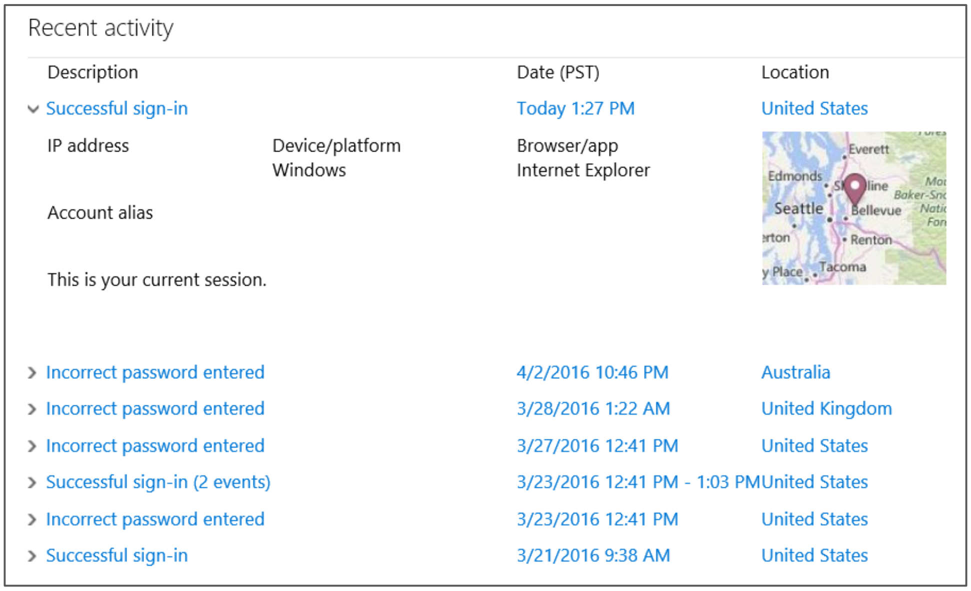
To prevent and mitigate such attacks, Microsoft uses a multi-layered system of protection mechanisms. The keystone of these protection systems is machine learning. Every day, Microsoft machine learning systems process more than 10 terabytes of data, including information on more than 13 billion requests from hundreds of millions of Microsoft Account users. These systems are powerful tools that enable Microsoft protection systems to aggregate and analyze huge data sets to take timely action. Microsoft also uses tools such as incorrect password lockout and location-based blocking. Multiple algorithms analyze a wide range of data produced by Microsoft systems, working in real-time to stop attacks before they are successful, and retroactively to swiftly remediate compromised accounts and revoke any access that an attacker might have obtained.

The capabilities described in the preceding paragraph are combined with other protection algorithms, data feeds from the Microsoft Digital Crimes Unit and the Microsoft Security Response Center, phishing attack data from Outlook.com and Exchange Online, and information acquired by partnering with academia, law enforcement, security researchers, and industry partners around the world to create a comprehensive protection system that helps keep customers' accounts safe.

From all this data gathering and analysis, each day Microsoft's account protection systems *automatically* **detect and prevent more than 10 million attacks, from tens of thousands of locations, including millions of attacks where the attacker has valid credentials**. That's **over 4 billion** attacks prevented last year alone.

Microsoft Accounts that are determined to be compromised are automatically entered into an account that are determined to be compromised are automatically entered into an account recovery process that allows only the rightful owner to regain sole access to the account. Microsoft Account users can also check the recent sign-in activity for their Microsoft account and report suspicious activity, as seen in Figure 21.

Figure 21. Screen shot of "Check the recent sign-in activity for your Microsoft account"

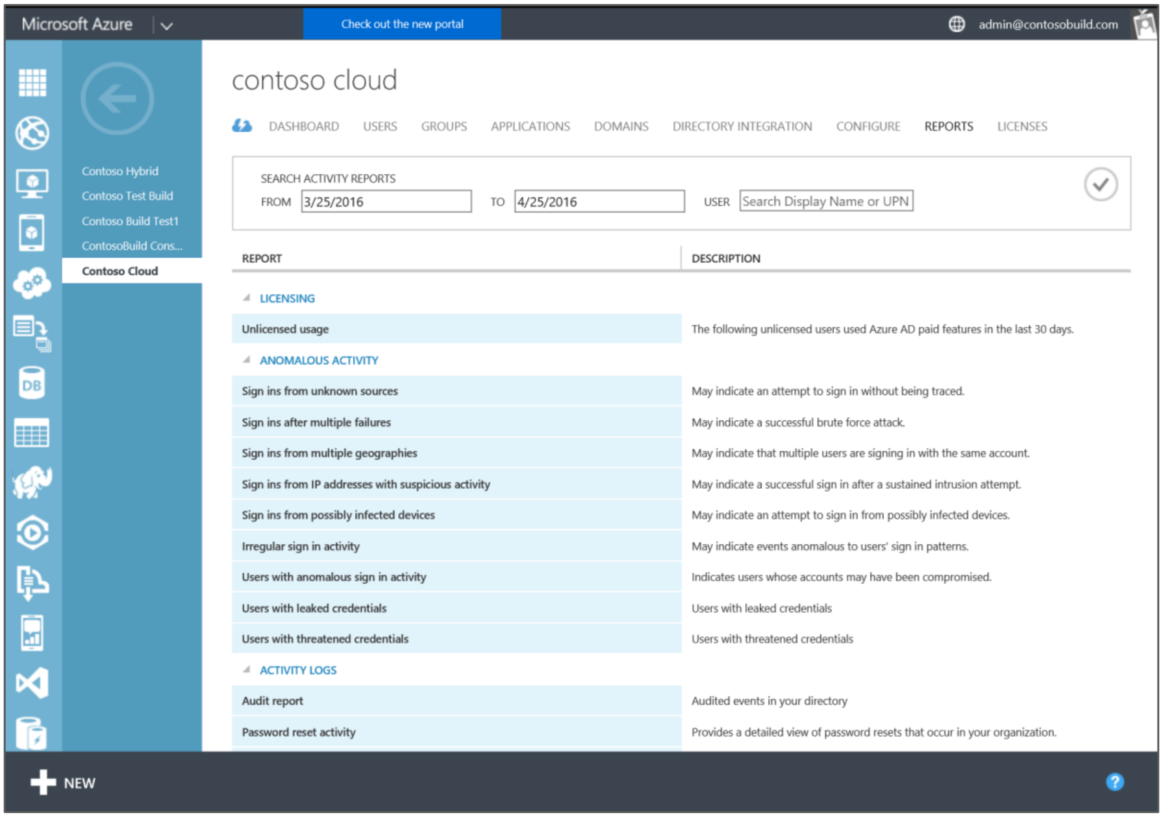


Recent activity

Description		Date (PST)	Location
Successful sign-in		Today 1:27 PM	United States
IP address	Device/platform Windows	Browser/app Internet Explorer	
Account alias			
This is your current session.			
Incorrect password entered		4/2/2016 10:46 PM	Australia
Incorrect password entered		3/28/2016 1:22 AM	United Kingdom
Incorrect password entered		3/27/2016 12:41 PM	United States
Successful sign-in (2 events)		3/23/2016 12:41 PM - 1:03 PM	United States
Incorrect password entered		3/23/2016 12:41 PM	United States
Successful sign-in		3/21/2016 9:38 AM	United States

Similarly, for Azure Active Directory accounts, Microsoft protection systems work to help mitigate problems for any accounts that are determined to be compromised. Potentially fraudulent login attempts and compromised accounts are reported to organizations via Access and Usage reports provided by Microsoft Azure Active Directory Premium, as seen in Figure 22.

Figure 22. Azure Active Directory access and usage reports



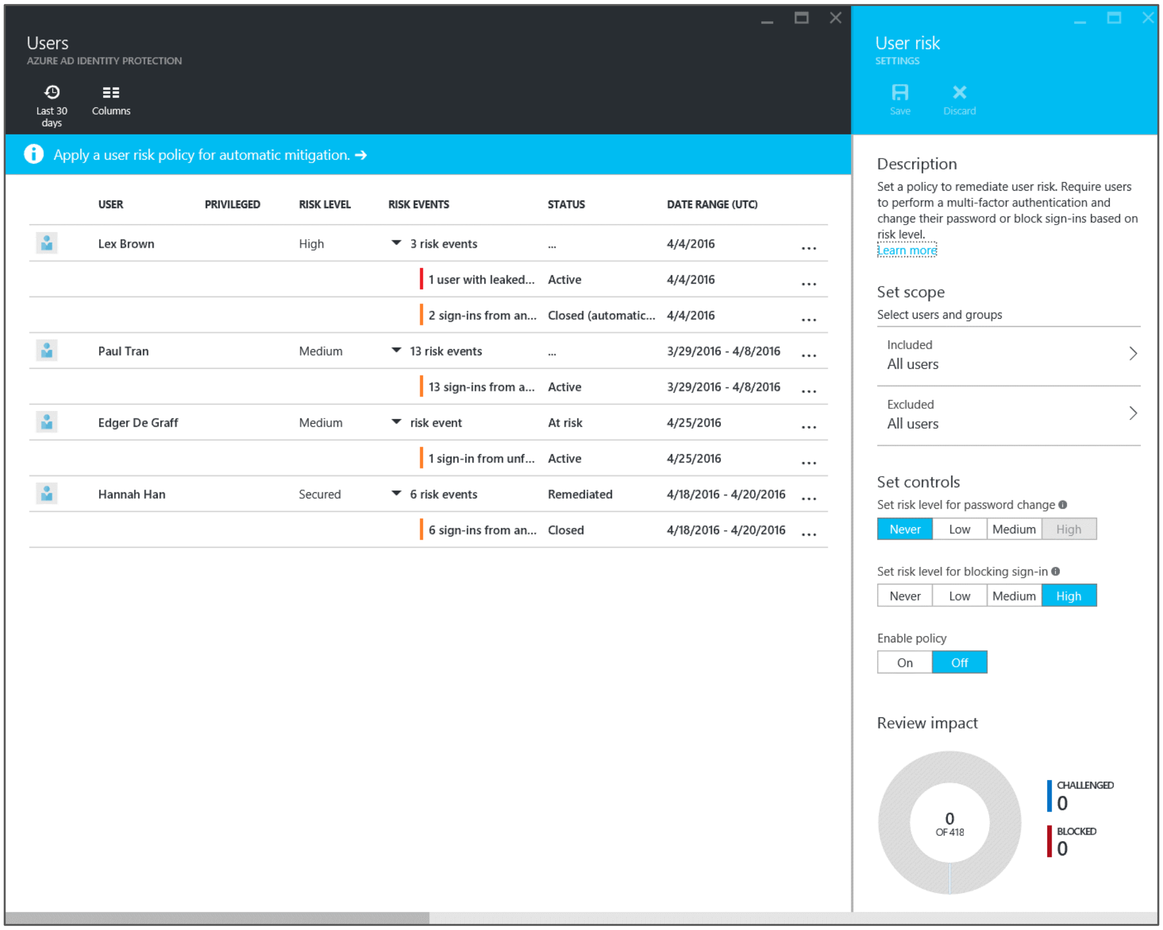
SEARCH ACTIVITY REPORTS

FROM 3/25/2016 TO 4/25/2016 USER Search Display Name or UPN

REPORT	DESCRIPTION
LICENSING	
Unlicensed usage	The following unlicensed users used Azure AD paid features in the last 30 days.
ANOMALOUS ACTIVITY	
Sign ins from unknown sources	May indicate an attempt to sign in without being traced.
Sign ins after multiple failures	May indicate a successful brute force attack.
Sign ins from multiple geographies	May indicate that multiple users are signing in with the same account.
Sign ins from IP addresses with suspicious activity	May indicate a successful sign in after a sustained intrusion attempt.
Sign ins from possibly infected devices	May indicate an attempt to sign in from possibly infected devices.
Irregular sign in activity	May indicate events anomalous to users' sign in patterns.
Users with anomalous sign in activity	Indicates users whose accounts may have been compromised.
Users with leaked credentials	Users with leaked credentials
Users with threatened credentials	Users with threatened credentials
ACTIVITY LOGS	
Audit report	Audited events in your directory
Password reset activity	Provides a detailed view of password resets that occur in your organization.

In a world in which massive lists of leaked and stolen passwords exist and passwords are commonly reused across websites, services, and devices, account compromises by attackers who use valid credentials are equally common. Microsoft machine learning systems consider the full scope of data described earlier to determine when an account login attempt, even with a valid password, is likely fraudulent. For Microsoft Accounts, these login attempts are blocked until a second authentication factor is provided. For Azure Active Directory, Identity Protection allows administrators to create policies that require additional authentication or block the attempt outright, based on the risk score of the login. An example can be seen in Figure 23.

Figure 23. Screen shot of Identity Protection, displaying some risky events and possible configuration vulnerabilities and the risk-based policy console



Users
AZURE AD IDENTITY PROTECTION

Apply a user risk policy for automatic mitigation. →

	USER	PRIVILEGED	RISK LEVEL	RISK EVENTS	STATUS	DATE RANGE (UTC)	
	Lex Brown		High	▼ 3 risk events	...	4/4/2016	...
				1 user with leaked...	Active	4/4/2016	...
				2 sign-ins from an...	Closed (automatic...	4/4/2016	...
	Paul Tran		Medium	▼ 13 risk events	...	3/29/2016 - 4/8/2016	...
				13 sign-ins from a...	Active	3/29/2016 - 4/8/2016	...
	Edger De Graff		Medium	▼ risk event	At risk	4/25/2016	...
				1 sign-in from unf...	Active	4/25/2016	...
	Hannah Han		Secured	▼ 6 risk events	Remediated	4/18/2016 - 4/20/2016	...
				6 sign-ins from an...	Closed	4/18/2016 - 4/20/2016	...

User risk
SETTINGS

Description

Set a policy to remediate user risk. Require users to perform a multi-factor authentication and change their password or block sign-ins based on risk level.
Learn more

Set scope

Select users and groups

Included
All users

Excluded
All users

Set controls

Set risk level for password change
[✓ Never] Low Medium High

Set risk level for blocking sign-in
Never Low Medium [✓ High]

Enable policy
On [✓ Off]

Review impact

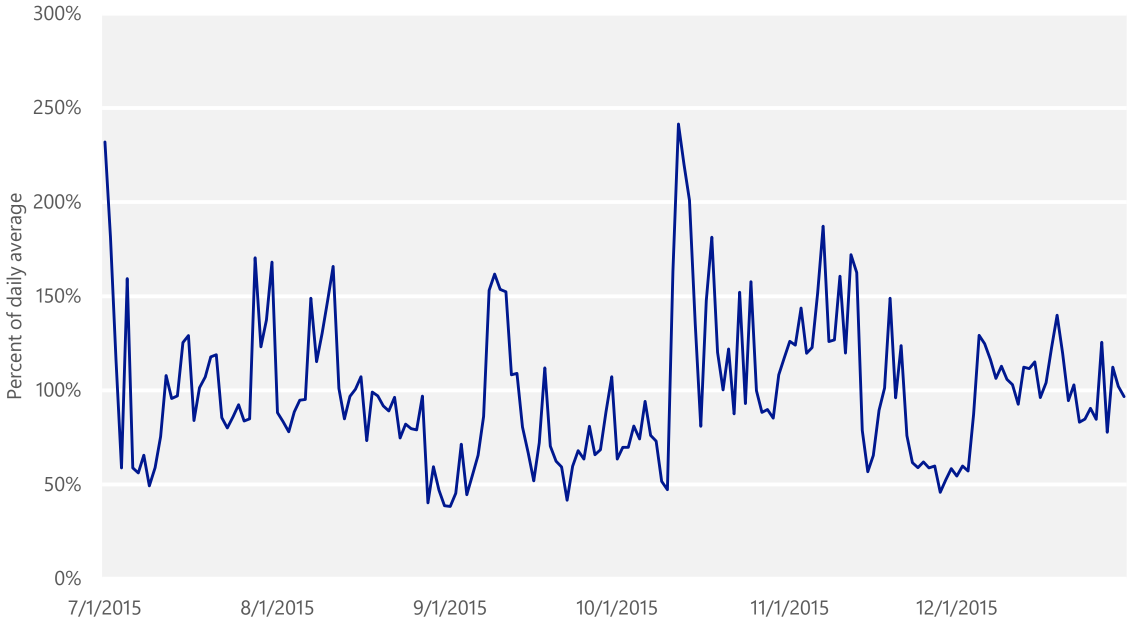
0
OF 418

CHALLENGED
0

BLOCKED
0

Figure 24 illustrates how the volume of blocked IP addresses changed from week to week in the second half of 2015 when the account password was valid but the protection system determined that the login attempt was likely fraudulent.

Figure 24. Number of IP addresses blocked from logging into Microsoft consumer cloud services in 2H15 when indicators suggested a fraudulent login attempt



The geographic locations of IP addresses that attempt fraudulent login requests are unevenly distributed around the world. Figure 25 provides a view into the distribution of IP addresses that attempted to log in to Microsoft consumer services during the second half of 2015 but were automatically blocked from doing so.

Figure 25. Number of IP addresses blocked from logging into Microsoft consumer cloud services in 2H15, by region

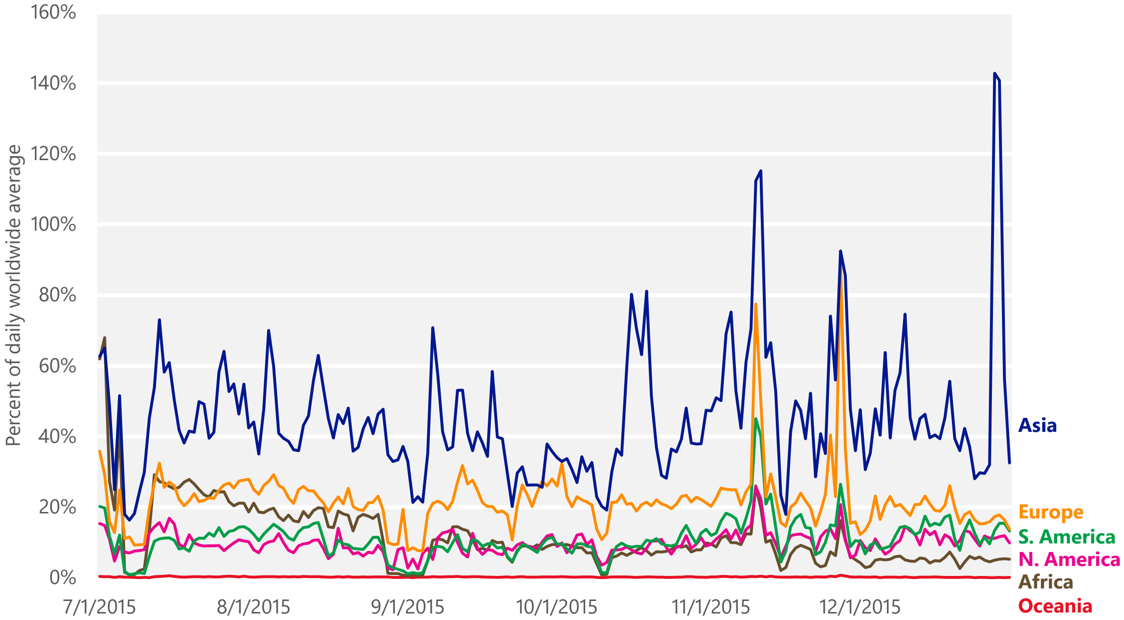
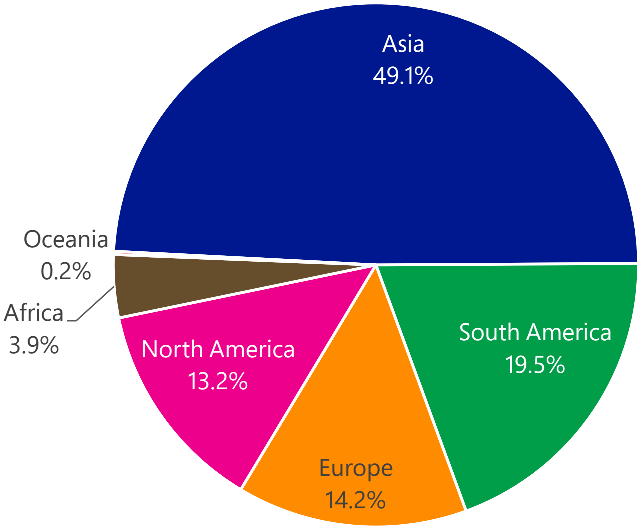


Figure 26 shows the geographic distribution of blocked IP addresses in the second half of 2015 by region. Almost half (49 percent) of these IP addresses were located in Asia, while 20 percent were in South America, 14 percent were in Europe, 13 percent in North America, and 4 percent in Africa.

Figure 26. Geographic distribution of IP addresses blocked from logging into Microsoft consumer cloud services during 2H15, by region



One of the factors that the machine learning system uses to block login attempts is whether the location of the login attempt is a familiar location to the legitimate user. Compromised login attempts that were blocked during the second half of 2015 were attempted from unfamiliar locations almost three quarters of the time.

Consumers and organizations can do a number of things to help mitigate the threat of account compromise as a result of password-based attacks.

- The security of your account is particularly important if your username is an email address, because other services may rely on your email address to verify your identity. If an attacker takes over your account, they may be able to take over your other accounts too (like banking and online shopping) by resetting your passwords by email.
- Tips for creating a strong and unique password:
    - Don't use a password that is the same or similar to one you use on any other website. A cybercriminal who can break into that website can steal your password from it and use it to steal your account.
    - Don't use a single word (e.g. "princess") or a commonly-used phrase (e.g. "Iloveyou").
    - Do make your password hard to guess even by those who know a lot about you (such as the names and birthdays of your friends and family, your favorite bands, and phrases you like to use).
- Two-step verification boosts account security by making it more difficult for hackers to sign in—even if they know or guess your password.
- If you turn on two-step verification and then try to sign in on a device we don't recognize, we'll ask you for two things:
    - Your password.
    - An extra security code.
    - We can send a new security code to your phone or your alternate email address, or you can get one through an authenticator app on your smartphone.
- Organizations should take advantage of Azure Active Directory Identity Protection. More information is available at: Azure AD and Identity Show:

Identity Protection Preview. Go to www.aka.ms/identityprotection to get started with Identity Protection.

Additional information about holistic protection strategy is available in this eBook: Protect Identities, Devices and Your Company Information in Today's Device-Centric World.

Fighting email spoofing with Exchange Online Protection

Business email compromise, in which an attacker spoofs the email address of a high-ranking official at an organization to steal money from the organization, has become a significant and growing problem for enterprise email users in recent years. To help Office 365 customers protect themselves against this fraud, Exchange Online Protection (EOP) has introduced a new antispoofing feature and made it available to all of its customers.

The business email compromise scam is a form of *spear phishing*, which targets specific individuals, organizations, or groups using information the attacker knows about the targets in order to deflect suspicion. (See "Phishing sites" beginning on page 129 for information about more conventional phishing methods and targets.) In a typical business email compromise attack, the attacker masquerades as a high-ranking official at an organization, such as the CEO, and sends an email to another official with access to money, such as the CFO. Unlike most phishing lures, the email message usually contains no links or attachments, just a customized request to transfer money to an account that is secretly controlled by the attacker. For example:

From: Rudy Bosive (the CEO) <rudy@woodgrovebank.com>
To: Tom Amtir (the CFO) <tom@woodgrovebank.com>
Subject: Can you make this wire transfer for me?

Tom, we just closed on an acquisition of a new service but we're trying to keep it quiet. Could you wire over $50,000 to them? The account number is below and we need to get this taken care of today.

Thanks.

Rudy

Sent from Outlook for iPhone

In this example, the sender's address (rudy@woodgrovebank.com) is spoofed by the attacker. The message is well-composed, and the identity and email address of the CEO are accurate. The message appears to have been sent internally, but actually came from outside the organization. If the recipient's suspicions are not aroused, he may follow the instructions without giving the matter any additional thought.

EOP's new antispoofing feature gives all Office 365 Business subscribers a new way to protect themselves against such attacks. EOP already provides its customers with industry-standard antispoofing and authentication mechanisms, including Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM), and Domain-based Message Authentication, Reporting, and Conformance (DMARC).[9] However, many organizations don't have either the expertise or resources needed to configure or maintain these mechanisms. The new feature provides spoofing protection for customer domains even in the absence of the authentication records required by the other mechanisms.

How EOP's antispoofing feature works

When an incoming message arrives from outside the customer's organization, EOP checks to see whether the sender's address matches any of the customer's provisioned domains, or any subdomain of any of the customer's provisioned domains. If so, EOP determines whether the sender's IP address is authorized to send mail on the customer's behalf, or if the message originates from a known good bulk sender. All such messages are considered legitimate, and are not marked as spam. For other messages, EOP uses a combination of sending domain reputation (or lack thereof), recipient reputation (how many messages the customer receives from this sender, and how the customer's email is routed through the EOP service), and machine learning to mark malicious messages as spam but leave legitimate messages alone. If EOP believes the message is a spoof, it marks the message as spam, and adds the following field to the X-Microsoft-Antispam header EOP adds to incoming messages:

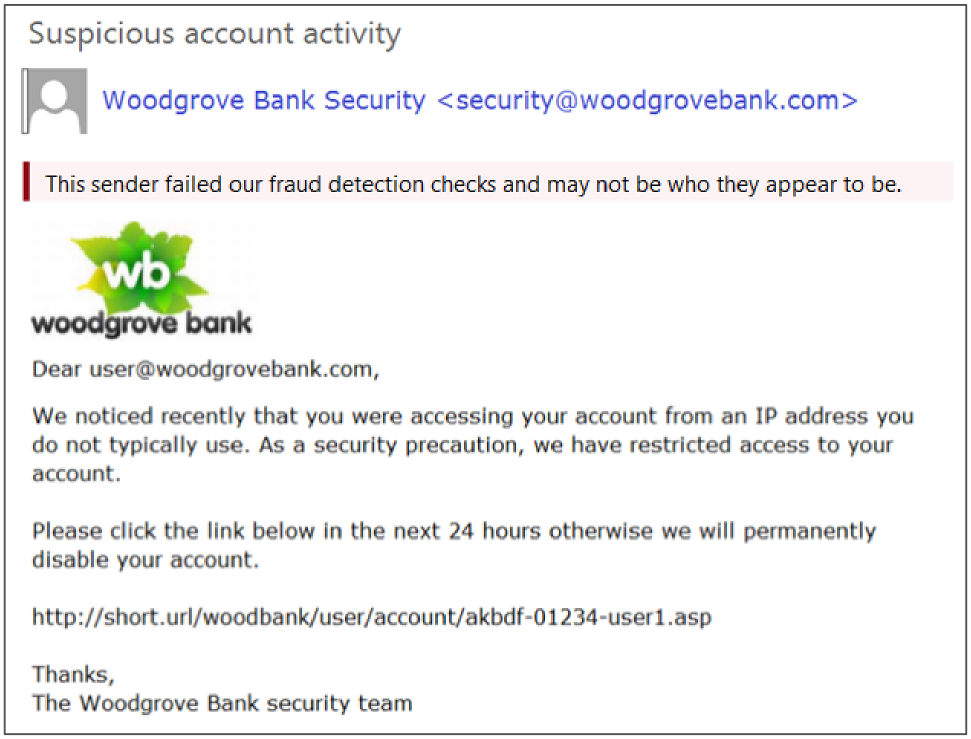
> By the end of the second quarter of 2016, EOP will start adding Safety Tips to the message

[9] EOP supports DMARC for inbound email, which is a technology to stop spoofing of the From: domain. The main difference between DMARC and EOP's antispoofing feature is that DMARC requires certain DNS records to be published, whereas EOP's antispoofing feature does not.

```
X-Microsoft-Antispam: […];SFTY:9.5
```

By the end of the second quarter of 2016, EOP will start adding Safety Tips to the message. Safety Tips are visual indicators that the message is fraudulent or may be a phishing scam. These Safety Tips are viewable when using Outlook to view email.

Figure 27. A Safety Tip indicating a possibly spoofed message



Suspicious account activity

Woodgrove Bank Security <security@woodgrovebank.com>

This sender failed our fraud detection checks and may not be who they appear to be.

Dear user@woodgrovebank.com,

We noticed recently that you were accessing your account from an IP address you do not typically use. As a security precaution, we have restricted access to your account.

Please click the link below in the next 24 hours otherwise we will permanently disable your account.

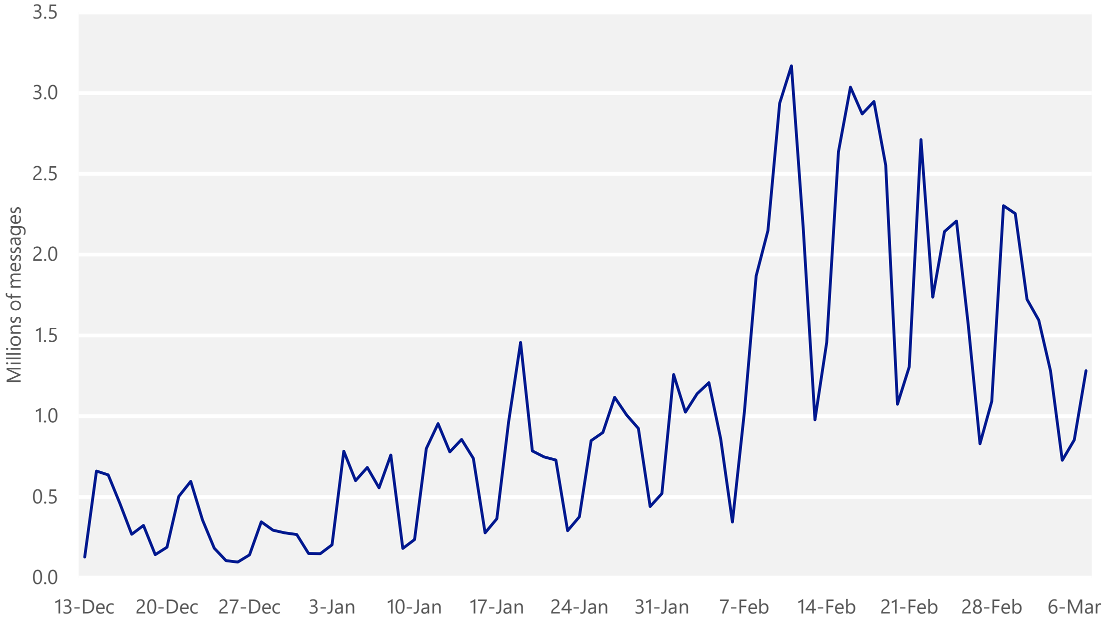
http://short.url/woodbank/user/account/akbdf-01234-user1.asp

Thanks,
The Woodgrove Bank security team

Antispoofing statistics

Figure 28 shows the number of messages identified as spoofs between December 13, 2015 and March 7, 2016, as the antispoofing feature was being made available to Office 365 Business customers.

Figure 28. Spoofed messages identified by EOP per day, December 2015–March 2016



As Figure 28 illustrates, spoofed messages are a fairly pervasive issue for users of business email, with more than half a million spoofs identified on some days even at the very beginning of the rollout. As more and more customer domains received the new feature, spoofed message volumes climbed into the millions per weekday, culminating in a total of more than 3 million messages on February 11. In subsequent weeks, spoof volumes began to decline slightly, suggesting that even a few weeks of antispoofing protection may have begun to convince some attackers to concentrate their efforts elsewhere.

Making the most of EOP's antispoofing services

See "How antispoofing protection works in Office 365" at https://blogs.msdn.microsoft.com/tzink/2016/02/23/how-antispoofing-protection-works-in-office-365/ for a more in-depth explanation of the new antispoofing feature and how to use it. This blog entry explains the new feature in more detail, including how to use Windows PowerShell to generate reports, how to designate IP addresses as safe, and how to configure domains to receive the best protection against spoofed messages.

Worldwide threat assessment

Vulnerabilities

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

Industry-wide vulnerability disclosures

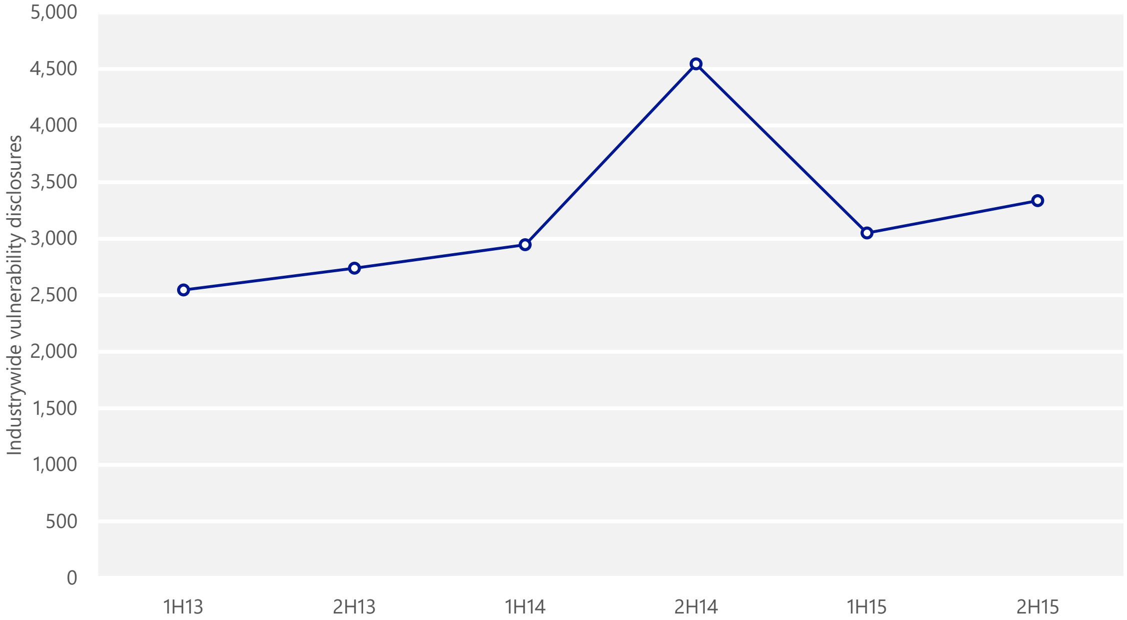
A *disclosure*, as the term is used in the *Microsoft Security Intelligence Report*, is the revelation of a software vulnerability to the public at large. Disclosures can come from a variety of sources, including publishers of the affected software, security software vendors, independent security researchers, and even malware creators.

The information in this section is compiled from vulnerability disclosure data that is published in the National Vulnerability Database (NVD), the US government's repository of standards-based vulnerability management data at nvd.nist.gov. The NVD represents all disclosures that have a published CVE (Common Vulnerabilities and Exposures) identifier.[10]

Figure 29 illustrates the number of vulnerability disclosures across the software industry for each half-year period since 1H13. (See "About this report" on page vi for an explanation of the reporting period nomenclature used in this report.)

[10] CVE entries are subject to ongoing revision as software vendors and security researchers publish more information about vulnerabilities. For this reason, the statistics presented here may differ slightly from comparable statistics published in previous volumes of the *Microsoft Security Intelligence Report*.

Figure 29. Industrywide vulnerability disclosures, 1H13–2H15



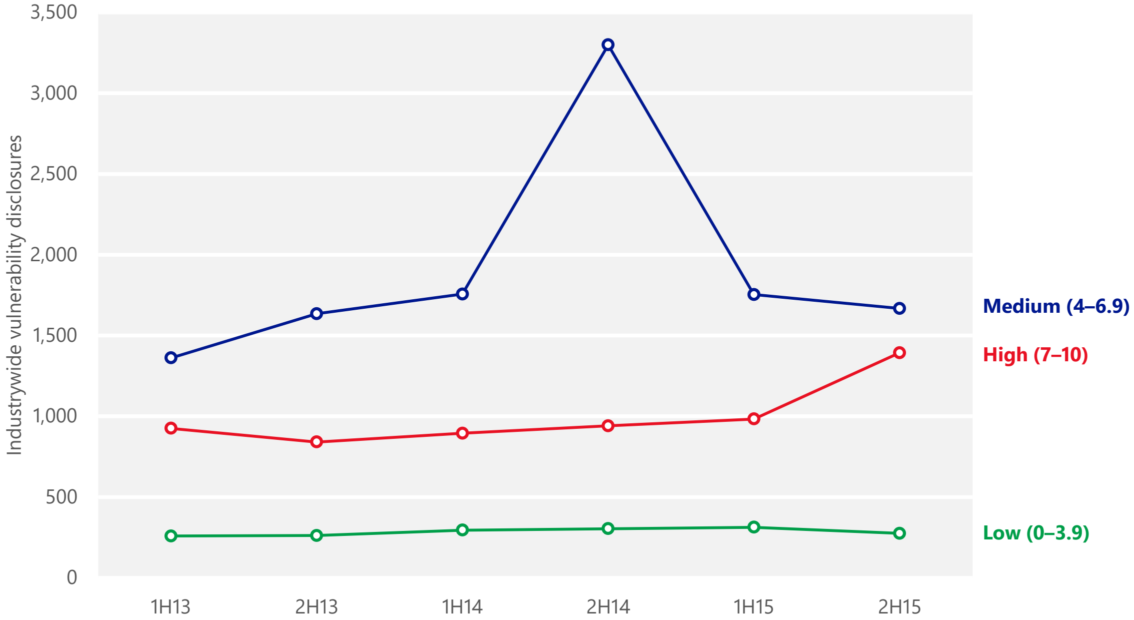
Vulnerability disclosures have trended generally upward over the past three years.

- Vulnerability disclosures across the industry increased 9.4 percent between 1H15 and 2H15, to just above 3,300.
- Vulnerability disclosures have trended generally upward over the past three years, with the exception of a spike in 2H14 caused by a research project at the Computer Emergency Response Team (CERT) Coordination Center (CERT/CC) that uncovered SSL-related man-in-the-middle vulnerabilities in a large number of Android apps in the Google Play Store.

Vulnerability severity

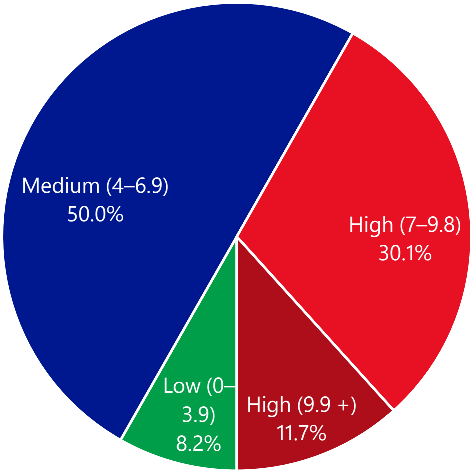
The Common Vulnerability Scoring System (CVSS) is a standardized, platform-independent scoring system for rating IT vulnerabilities. The CVSS base metric assigns a numeric value between 0 and 10 to vulnerabilities according to severity, with higher scores representing greater severity. (See A Complete Guide to the Common Vulnerability Scoring System Version 2.0 at first.org for more information.)

Figure 30. Industrywide vulnerability disclosures by severity, 1H13–2H15



- Disclosures of high-severity vulnerabilities—those with CVSS scores of 7 and above—increased 41.7 percent across the industry in 2H15, to account for 41.8 percent of all vulnerabilities, the largest share for such vulnerabilities for at least three years.
- This increase included a disproportionate rise in disclosures of vulnerabilities rated 9.9 or higher, which increased 73.7 percent in 2H15. These highest severity vulnerabilities accounted for 11.7 percent of all disclosures, as shown in Figure 31.

Figure 31. Industrywide vulnerability disclosures in 2H15, by severity



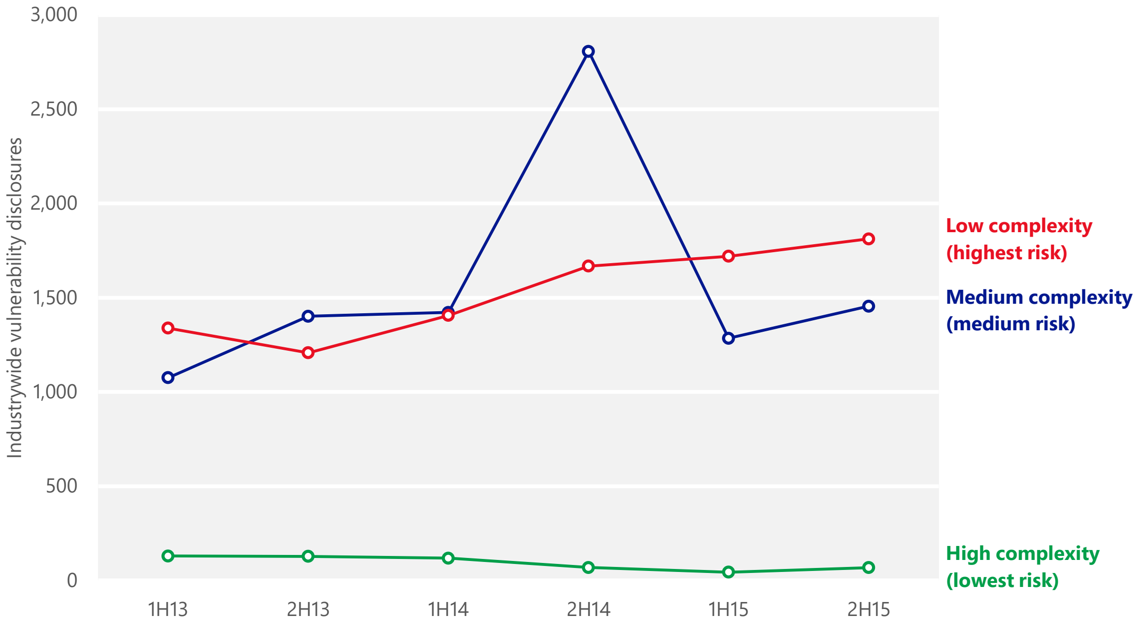
- Disclosures of medium- and low-severity vulnerabilities, by contrast, both decreased slightly between 1H15 and 2H15.

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 assigns each vulnerability a complexity ranking of Low, Medium, or High. (See A Complete Guide to the Common Vulnerability Scoring System Version 2.0 at first.org for more information about the CVSS complexity ranking system.) Figure 32 shows complexity trends for vulnerabilities disclosed since 1H13. Note that Low complexity in Figure 32 indicates greater risk, just as High severity indicates greater risk in Figure 30.

Figure 32. Industrywide vulnerability disclosures by access complexity, 1H13–2H15



- Overall, disclosures increased slightly at all levels of complexity between 1H15 and 2H15.

- Disclosures of low-complexity vulnerabilities—those that are the easiest to exploit—accounted for the largest category of disclosures, at 54.3 percent of all disclosures for the period.
- Medium-complexity vulnerabilities accounted for the second largest share, at 43.6 percent of all vulnerabilities.
- Disclosures of high-complexity vulnerabilities accounted for just 2.0 percent of all disclosures.

Operating system, browser, and application vulnerabilities

Comparing vulnerabilities that affect a computer's operating system to vulnerabilities that affect other components, such as applications and utilities, requires a determination of whether the affected component is considered part of the operating system. This determination is not always simple and straightforward, given the componentized nature of modern operating systems. Some programs (media players, for example) ship by default with some operating system software but can also be downloaded from the software vendor's website and installed individually. Linux distributions, in particular, are often assembled from components developed by different teams, many of which provide crucial operating functions such as a graphical user interface (GUI) or Internet browsing.

> Disclosures of low-complexity vulnerabilities—those that are the easiest to exploit—accounted for the largest category of disclosures.

To facilitate analysis of operating system and browser vulnerabilities, the *Microsoft Security Intelligence Report* distinguishes among four different kinds of vulnerabilities:

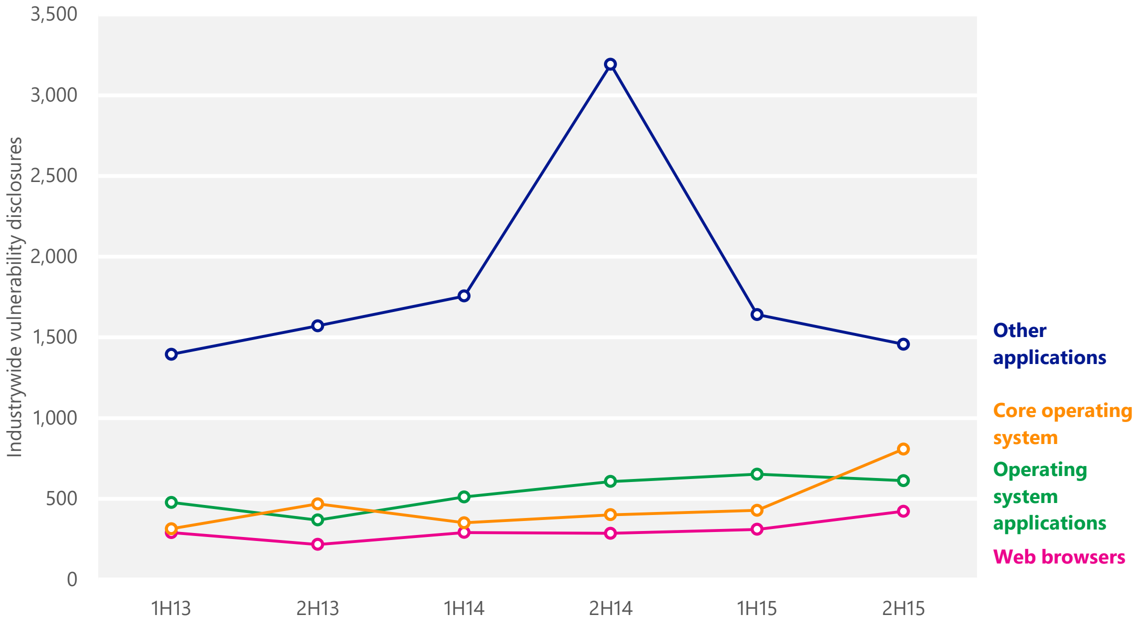
- *Core operating system vulnerabilities* are those with at least one operating system platform enumeration (/o) in the NVD that do not also have any application platform enumerations (/a).[11]
- *Operating system application vulnerabilities* are those with at least one /o platform enumeration and at least one /a platform enumeration listed in the NVD, except as described in the next bullet point.

[11] See nvd.nist.gov/cpe.cfm for information about the Common Platform Enumeration (CPE) standard for naming information technology systems, software, and packages.

- *Browser vulnerabilities* are those that affect components defined as part of a web browser, including web browsers such as Internet Explorer and Apple's Safari that ship with operating systems, along with third-party browsers such as Mozilla Firefox and Google Chrome.
- *Other application vulnerabilities* are those with at least one /a platform enumeration in the NVD that do not have any /o platform enumerations, except as described in the previous bullet point.

Figure 33 shows industrywide vulnerabilities for operating systems, browsers, and applications since 1H13.

Figure 33. Industrywide operating system, browser, and application vulnerabilities, 1H13–2H15

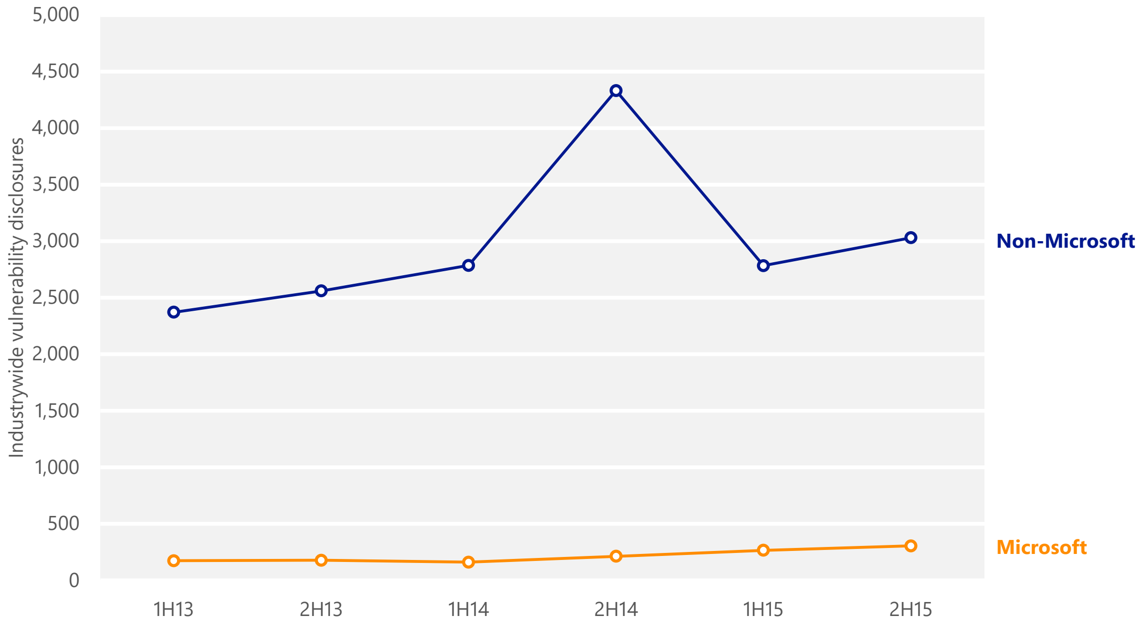


- Disclosures of vulnerabilities in applications other than web browsers and operating system applications decreased in 2H15, but remained the most common type of vulnerability during the period, accounting for 44.2 percent of all disclosures for the period.
- Core operating system vulnerability disclosures increased 88.8 percent from 1H15, moving it into second place, at 24.5 percent of all disclosures in 2H15.
- Operating system application vulnerability disclosures decreased slightly to account for 18.6 percent of all disclosures in 2H15.
- Browser vulnerability disclosures increased 36.1 percent from 1H15, and accounted for 12.8 percent of all disclosures in 2H15.

Microsoft vulnerability disclosures

Figure 34 shows trends for vulnerability disclosures that affect Microsoft products compared to the rest of the industry.

Figure 34. Vulnerability disclosures for Microsoft and non-Microsoft products, 1H13–2H15



- Microsoft vulnerability disclosures increased from 266 disclosures in 1H15 to 305 in 2H15, an increase of 14.7 percent.

Guidance: Developing secure software

The Security Development Lifecycle (SDL) (www.microsoft.com/sdl) is a free software development methodology that incorporates security and privacy best practices throughout all phases of the development process, with the goal of protecting software users. Using such a methodology can help reduce the number and severity of vulnerabilities in software and help manage vulnerabilities that might be discovered after deployment.

"Life in the Digital Crosshairs," at sdlstory.com, is a multimedia presentation that explores the genesis and development of the SDL from its origins in the Windows team's well-documented all-hands security push in the early 2000s. It includes interviews with several of the pivotal figures in the history of the SDL and Microsoft's focus on secure software. Security professionals and anyone else with an interest in secure development are likely to find the site invaluable for putting the SDL into historical context and understanding what the future holds.

To learn more about how the SDL is applied in the present day, see "State of Application Security: Immature Practices Fuel Inefficiencies, but Positive ROI Is Attainable - A Forrester Consulting Thought Leadership Paper Commissioned by Microsoft" to learn how organizations are putting SDL techniques to work for them, and "Secure Software Development Trends in the Oil & Gas Sectors" for an example of how the SDL has helped one critical industry. Both papers are available from the Microsoft Download Center (www.microsoft.com/download).

Exploits

An *exploit* is a piece of code that uses software vulnerabilities to access information on a computer or install malware. Exploits target vulnerabilities in operating systems, web browsers, applications, or software components that are installed on a computer.

In some scenarios, targeted components are add-ons that may be 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. In addition, 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.

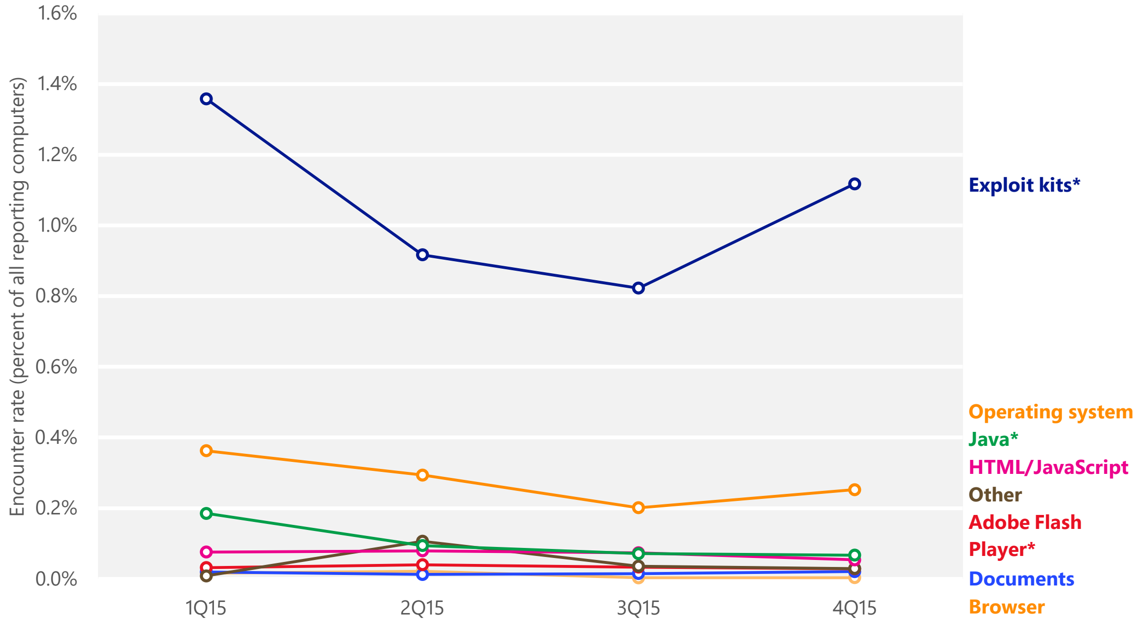
Software vulnerabilities are enumerated and documented in the Common Vulnerabilities and Exposures (CVE) list (cve.mitre.org), a standardized repository of vulnerability information. Here and throughout this report, exploits are labeled with the CVE identifier that pertains to the affected vulnerability, 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.[12]

Microsoft real-time security products can detect and block attempts to exploit known vulnerabilities whether the computer is affected by the vulnerabilities or not. For example, the CVE-2010-2568 CplLnk vulnerability has never affected Windows 8, but if a Windows 8 user receives a malicious file that attempts to exploit that vulnerability, Windows Defender is designed to detect and block it anyway. Encounter data provides important information about which products and vulnerabilities are being targeted by attackers, and by what means. However, the statistics presented in this report should not be interpreted as evidence of successful exploit attempts, or of the relative vulnerability of computers to different exploits.

[12] See technet.microsoft.com/security/bulletin to search and read Microsoft Security Bulletins.

Figure 35 shows the prevalence of different types of exploits detected by Microsoft antimalware products each quarter in 2015, by encounter rate. *Encounter rate* is the percentage of computers running Microsoft real-time security products that report a malware encounter. For example, the encounter rate for operating system exploit attempts in 4Q15 was 0.25 percent, meaning that 0.25 percent of computers running Microsoft real-time security software in 4Q15 encountered operating system exploit attempts, and 99.75 percent did not. In other words, a computer selected at random would have had about a 0.25 percent chance of encountering an operating system exploit attempt in 4Q15. Only computers whose users have opted in to provide data to Microsoft are considered when calculating encounter rates.[13] See page 79 for more information about the encounter rate metric.

Figure 35. Encounter rates for different types of exploit attempts in 2015



* Figures for exploit kits, Java, and Adobe Flash Player exploits are affected by IExtensionValidation in Internet Explorer, which blocks many threats before they are encountered. See page 76 for more information.

- Computers that report more than one type of exploit are counted for each type detected.
- After decreasing steadily for more than a year, encounters with exploit kits increased by more than a third from 3Q15 to 4Q15. They remained the most

[13] For information about the products and services that provide data for this report, see "Appendix B: Data sources" on page 157.

commonly encountered type of exploit in the second half of the year, with an encounter rate more than four times that of the next most common type of exploit. See "Exploit kits" on page 66 for more information about these exploits.

- The number of encounters with exploits that target operating systems increased slightly in 4Q15, but remained lower than in the first half of the year. Operating system exploits were the second most commonly encountered type of exploits during the period. See "Operating system exploits" on page 71 for more information.
- Encounters with Java exploits, Adobe Flash Player exploits, and other types of exploits each accounted for less than 0.1 percent of all malware encounters in 2H15. See the remainder of this section for more information about these exploits.

Exploit families

Figure 36 lists the exploit-related malware families that were detected most often during the second half of 2015.

Figure 36. Quarterly encounter rate trends for the exploit families most commonly detected and blocked by Microsoft real-time antimalware products in 2H15, shaded according to relative prevalence

Exploit	Type	1Q15	2Q15	3Q15	4Q15
Axpergle	Exploit kit	0.86%	0.66%	0.71%	0.92%
CVE-2010-2568 (CplLnk)	Operating system	0.30%	0.23%	0.18%	0.24%
HTML/Meadgive	Exploit kit	0.06%	0.05%	0.07%	0.17%
JS/NeutrinoEK	Exploit kit	0.06%	0.03%	0.01%	0.11%
HTML/IframeRef	Generic	0.07%	0.05%	0.04%	0.05%
JS/Neclu	Exploit kit	0.03%	0.15%	0.05%	0.01%
ShellCode	Other	0.01%	0.02%	0.01%	0.03%
Win32/Sdbby	Other	0.00%	0.09%	0.02%	0.01%
CVE-2012-1723	Java	0.04%	0.02%	0.02%	0.02%
Java/Obfuscator	Java	0.04%	0.05%	0.02%	0.01%

Totals for individual vulnerabilities do not include exploits that were detected as part of exploit kits.

- Exploit kits accounted for four of the 10 most commonly encountered exploits during 2H15. See "Exploit kits" on page 66 for more information about exploit kits.

- CVE-2010-2568, the most commonly targeted individual vulnerability in 1H15, is a vulnerability in Windows Shell. Detections are often identified as variants in the Win32/CplLnk family, although several other malware families attempt to exploit the vulnerability as well. An attacker exploits CVE-2010-2568 by creating a malformed shortcut file—typically distributed through social engineering or other methods—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 other families, many of which predated the disclosure of the vulnerability and were subsequently adapted to attempt to exploit it. Microsoft published Security Bulletin MS10-046 in August 2010 to address the issue. Windows 8 and subsequently released versions of Windows have never been vulnerable to exploits of CVE-2010-2568.

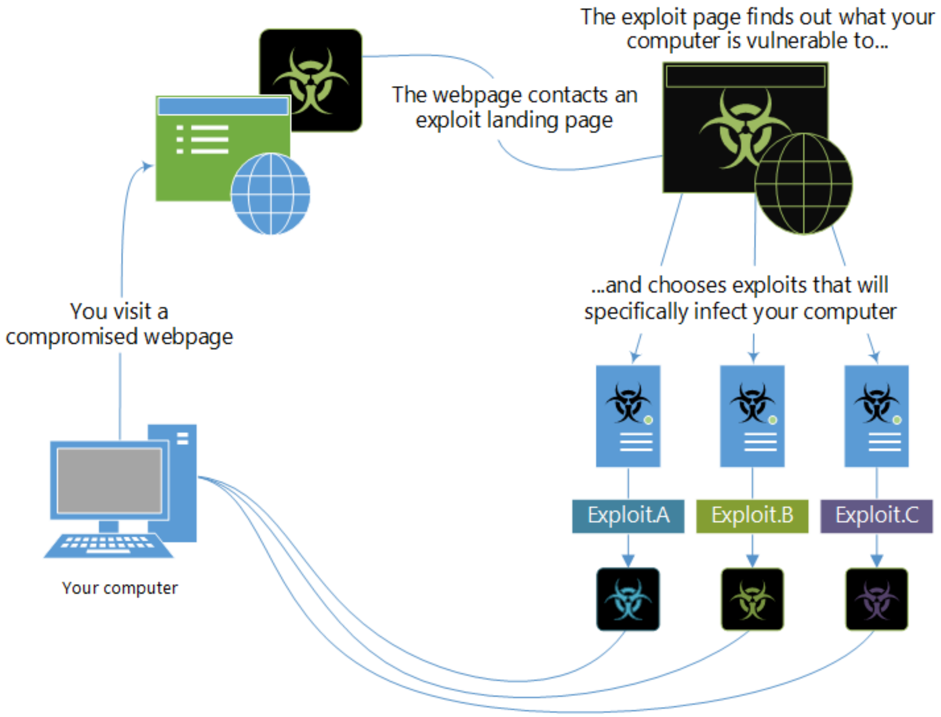
Exploit kits accounted for four of the 10 most common exploits during 2H15.

- HTML/IframeRef is a generic detection for specially formed HTML inline frame (IFrame) tags that redirect to remote websites that contain malicious content. More properly considered exploit downloaders than true exploits, these malicious pages use a variety of techniques to exploit vulnerabilities in browsers and plug-ins. The only commonality is that the attacker uses an inline frame to deliver the exploits to users. The exact exploit delivered and detected by one of these inline frames might be changed frequently.
- Win32/Sdbby is a generic detection for malware that bypasses the User Account Control (UAC) prompt to gain administrative privileges on a computer. After briefly becoming the fourth most commonly encountered exploit family in 2Q15, it decreased to much lower levels during the second half of the year.

Exploit kits

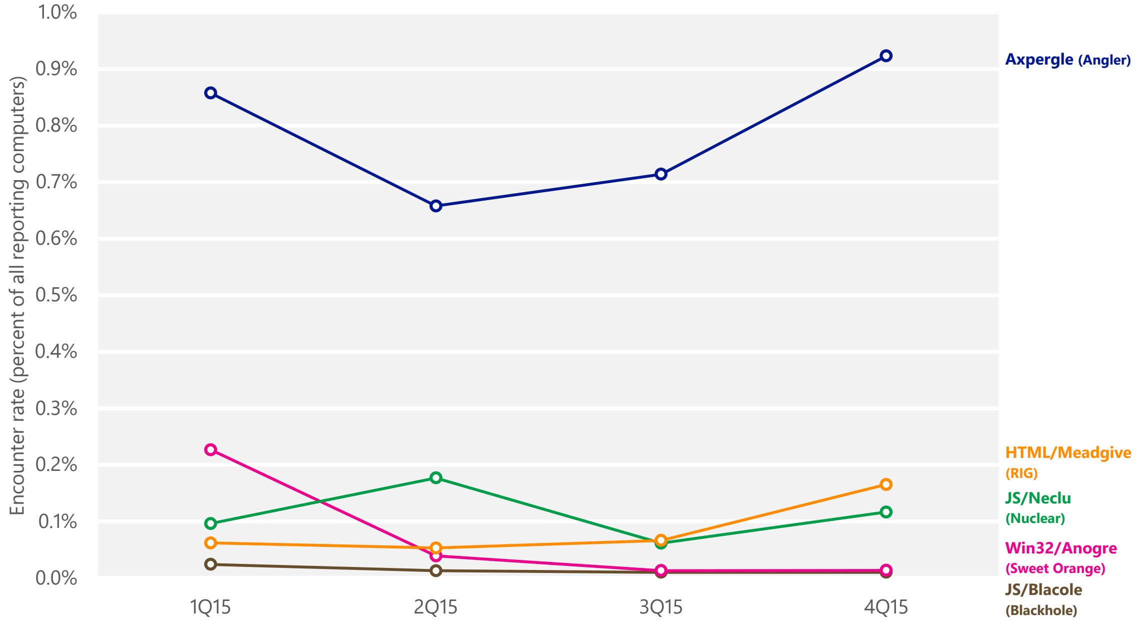
Exploit kits are collections of exploits bundled together and sold as commercial software or as a service. Prospective attackers buy or rent exploit kits on malicious hacker forums and through other illegitimate outlets. A typical kit comprises a collection of webpages that contain exploits for several vulnerabilities in popular web browsers and browser add-ons. When the attacker installs the kit on a malicious or compromised web server, visitors who don’t have the appropriate security updates installed are at risk of having their computers compromised through drive-by download attacks. (See page 133 for more information about drive-by downloads.)

Figure 37. How a typical exploit kit works


The exploit page finds out what your computer is vulnerable to...
The webpage contacts an exploit landing page
You visit a compromised webpage
...and chooses exploits that will specifically infect your computer
Exploit.A
Exploit.B
Exploit.C
Your computer

Microsoft security products detect and block the characteristic techniques that a number of common exploit kits use to infect computers, along with several generic HTML and JavaScript exploit techniques. Figure 38 shows the prevalence of several top web-based exploit kits and techniques during each of the four most recent quarters.

Figure 38. Trends for the top exploit kit-related threats detected and blocked by Microsoft real-time antimalware products in 2H15



- JS/Axpergle, a detection for the so-called Angler exploit kit, was the most commonly encountered exploit kit family in 2H15. It is known to target a number of vulnerabilities in Silverlight (CVE-2013-0074), Internet Explorer (CVE-2013-2551), Adobe Flash Player (CVE-2015-0310, CVE-2015-0311, and CVE-2015-0313, among others), and Java (CVE-2013-2460), although exploit kit authors frequently change the exploits included in their kits in an effort to stay ahead of software publishers and security software vendors. Exploits targeting *zero-day vulnerabilities*—those for which no security update has yet been made available by the vendor—are highly sought after by attackers, and the Axpergle authors added several zero-day Flash Player exploits to the kit in 2015, including CVE-2015-5122 and CVE-2015-7645.

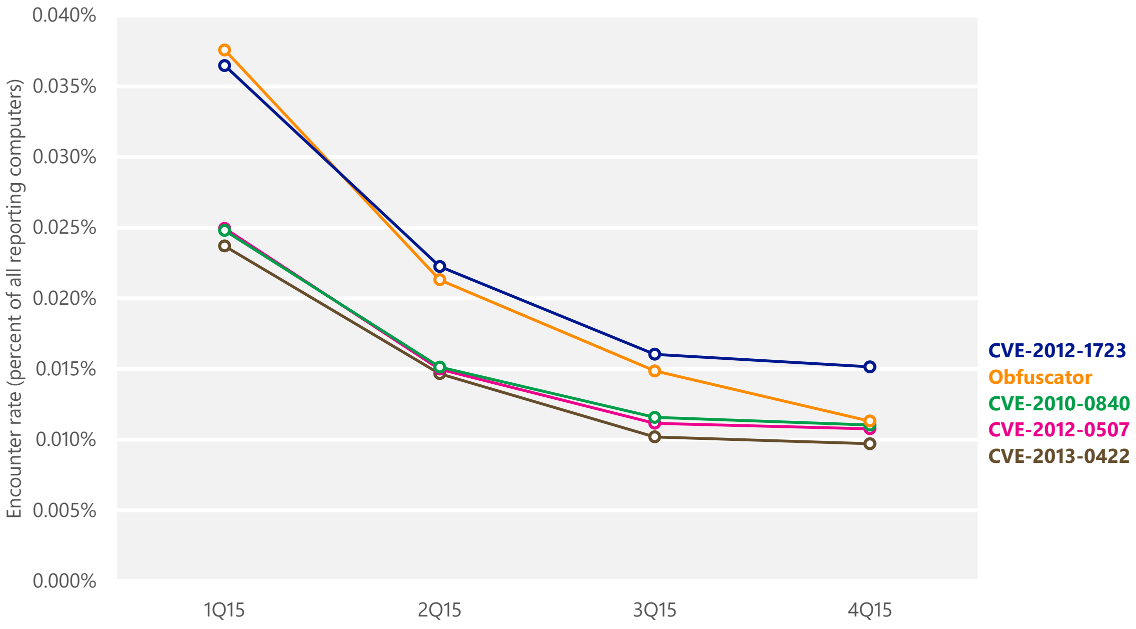
- Other exploit kits were encountered at much lower levels in 2H15. Encounters involving the RIG exploit kit (also known as Redkit, Infinity, and Goon, and detected as HTML/Meadgive) more than doubled between 3Q15 and 4Q15, but remained far below those involving Angler. Encounters involving the Nuclear kit (detected as JS/Neclu) increased between the third and fourth quarters, but remained below their 2Q15 levels.

- Encounters involving the Sweet Orange kit (detected as Win32/Anogre), the second most commonly encountered exploit kit in 1Q15, decreased to negligible levels by the end of the year.

Java exploits

Figure 39 shows the prevalence of different Java exploits by quarter.

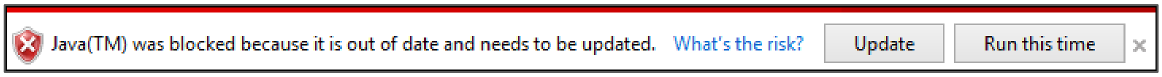
Figure 39. Trends for the top Java exploits detected and blocked by Microsoft real-time antimalware products in 2H15



- Overall, encounters with Java exploits continued to decrease significantly in 2H15. This decrease is likely caused by several important changes in the way web browsers evaluate and execute Java applets:
    - The IExtensionValidation interface in Internet Explorer 11, released in late 2013, provides a mechanism for security software to validate that a webpage is safe before allowing instantiation of ActiveX controls, such as the control that hosts embedded Java applets. If a webpage is determined to be malicious, the ActiveX controls are blocked from loading, and the actual Java exploit itself is therefore never encountered. (See "Exploit detection with Internet Explorer and IExtensionValidation" on page 76 for more information.) Subsequent Internet Explorer security updates released in 2014 added an isolated heap mechanism and a deferred-free method to mitigate use-after-free bugs, which further hardened Internet Explorer against Java exploitation.
    - Beginning with Java 7 update 51, released in January 2014, the Java Runtime Environment (JRE) requires Java applets running in web browsers to be digitally signed by default.

  - In September 2014, Microsoft published updates for versions 8 through 11 of Internet Explorer to begin blocking out-of-date ActiveX controls, including controls that host older versions of the JRE in the browser. As explained in this section, the most commonly encountered Java exploits all target vulnerabilities that were addressed with security updates years ago, but remain present in out-of-date Java installations. When a webpage attempts to load one of the vulnerable versions of Java in Internet Explorer with the update applied, the control is blocked by default and the user is urged to update Java to a more secure version.

Figure 40. Internet Explorer blocks out-of-date ActiveX controls from running



Java(TM) was blocked because it is out of date and needs to be updated.

  - Microsoft Edge, the newest Microsoft web browser and the default browser in Windows 10, does not support Java or other ActiveX plugins at all, which eliminates the possibility of Java exploits being delivered within the browser. See "A break from the past, part 2: Saying goodbye to ActiveX, VBScript, attachEvent…" (May 6, 2015) at the Microsoft Edge Dev Blog at blogs.windows.com/msedgedev for more information.

- CVE-2012-1723, the most commonly encountered individual Java exploit in 2H15, is a type-confusion vulnerability in the Java Runtime Environment (JRE) that is exploited by tricking the JRE into treating one type of variable like another type. Oracle confirmed the existence of the vulnerability in June 2012, and addressed it the same month with its June 2012 Critical Patch Update. The vulnerability was observed being exploited in the wild beginning in early July 2012, and has been used in a number of exploit kits.

  For more information about this exploit, see the entry "The rise of a new Java vulnerability - CVE-2012-1723" (August 1, 2012) in the Microsoft Malware Protection Center (MMPC) blog at blogs.technet.com/mmpc.

- Obfuscator is a generic detection for programs that have been modified by malware obfuscation, often in an attempt to avoid detection by security software. Files identified as Java/Obfuscator can represent exploits that target many different Java vulnerabilities.

- CVE-2010-0840 is a JRE vulnerability that was first disclosed in March 2010 and addressed by Oracle with a security update the same month. The vulnerability was previously exploited by some versions of the Blackhole exploit kit (detected as JS/Blacole), which has been inactive in recent years.

- CVE-2012-0507 allows an unsigned Java applet to gain elevated permissions and potentially have unrestricted access to a host system outside its sandbox environment. The vulnerability is a logic error that allows attackers to run code with the privileges of the current user, which means that an attacker can use it to perform reliable exploitation on other platforms that support the JRE, including Apple Mac OS X, Linux, VMWare, and others. Oracle released a security update in February 2012 to address the issue.

> CVE-2013-0422 first appeared in January 2013 as a zero-day vulnerability.

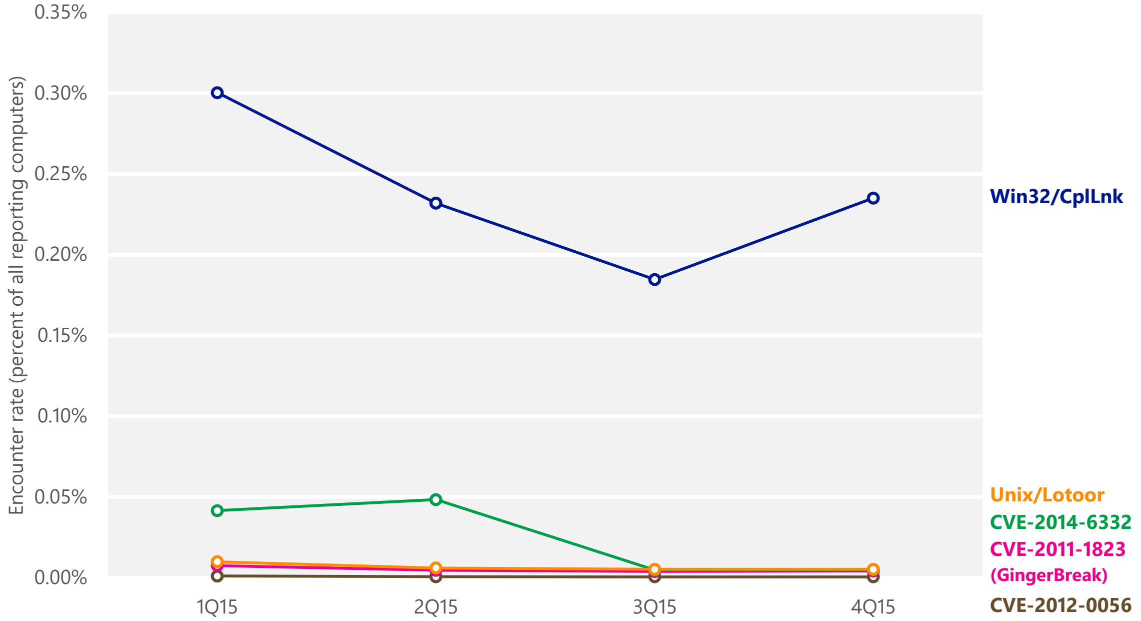
- CVE-2013-0422 first appeared in January 2013 as a zero-day vulnerability. CVE-2013-0422 is a package access check vulnerability that allows an untrusted Java applet to access code in a trusted class, which then loads the attacker's own class with elevated privileges. Oracle published a security update to address the vulnerability on January 13, 2013.

  For more information about CVE-2013-0422, see the entry "A technical analysis of a new Java vulnerability (CVE-2013-0422)" (January 20, 2013) in the MMPC blog at blogs.technet.com/mmpc. .

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, malicious or infected files that affect other operating systems are sometimes downloaded. Figure 41 shows trends for the individual exploits most commonly detected and blocked or removed during each of the past four quarters.

Figure 41. Trends for the top operating system exploits detected and blocked by Microsoft real-time antimalware products in 2015



- Win32/CplLnk, an exploit that targets a vulnerability in Windows Shell, remained the most commonly encountered operating system exploit in 2H15. An attacker exploits the vulnerability (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. Microsoft released Security Bulletin MS10-046 in August 2010 to address this issue.
- Two of the five most commonly encountered operating system exploits on Windows computers in 2H15 actually target the Android mobile operating system published by Google and the Open Handset Alliance. Microsoft security products detect these threats when Android devices or storage cards are connected to computers running Windows, or when Android users knowingly or unknowingly download infected or malicious programs to their computers before transferring the software to their devices. Most detections that affect Android involve exploits that enable an attacker or other user to obtain root privileges on vulnerable Android devices. Device owners sometimes use such exploits intentionally to gain access to additional functionality (a practice often called *rooting* or *jailbreaking*), but these exploits can also be used by attackers to infect devices with malware that bypasses many typical security systems.

  - Unix/Lotoor is an exploit family that exploits vulnerabilities in the Android operating system to gain root privileges on a mobile device. Google published a source code update in March 2011 to address the vulnerability.
  - CVE-2011-1823 is sometimes called the GingerBreak vulnerability because of its use by a popular rooting application of that name. It is also used by AndroidOS/GingerMaster, a malicious program that can allow a remote attacker to gain access to the mobile device. GingerMaster might be bundled with clean applications, and includes an exploit for the CVE-2011-1823 vulnerability disguised as an image file. Google published a source code update in May 2011 to address the vulnerability.

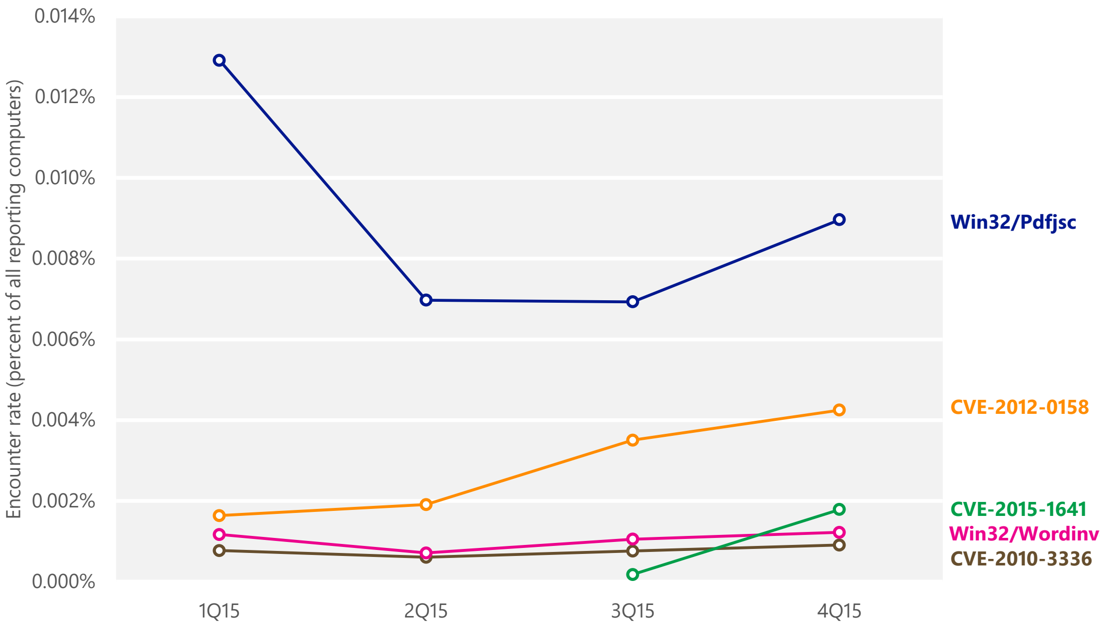
- CVE-2014-6332 is a vulnerability in Windows Object Linking and Embedding (OLE) that can be used to launch remote attacks on a computer through Internet Explorer in some circumstances. Microsoft released Security Bulletin MS14-064 in November 2014 to address this issue. See "The life and times of an exploit" on pages 3–10 of *Microsoft Security Intelligence Report, Volume 18 (July–December 2014)*, available from the Microsoft Download Center, for more information about this vulnerability and what Microsoft has done to mitigate it.

> Most detections that affect Android involve exploits that enable an attacker or other user to obtain root privileges on vulnerable Android devices.

Document exploits

Document exploits are exploits that target vulnerabilities in the way a document editing or viewing application processes a particular file format. Figure 42 shows encounter rates for individual exploits.

Figure 42. Trends for the top document exploits detected and blocked by Microsoft real-time antimalware products in 2015

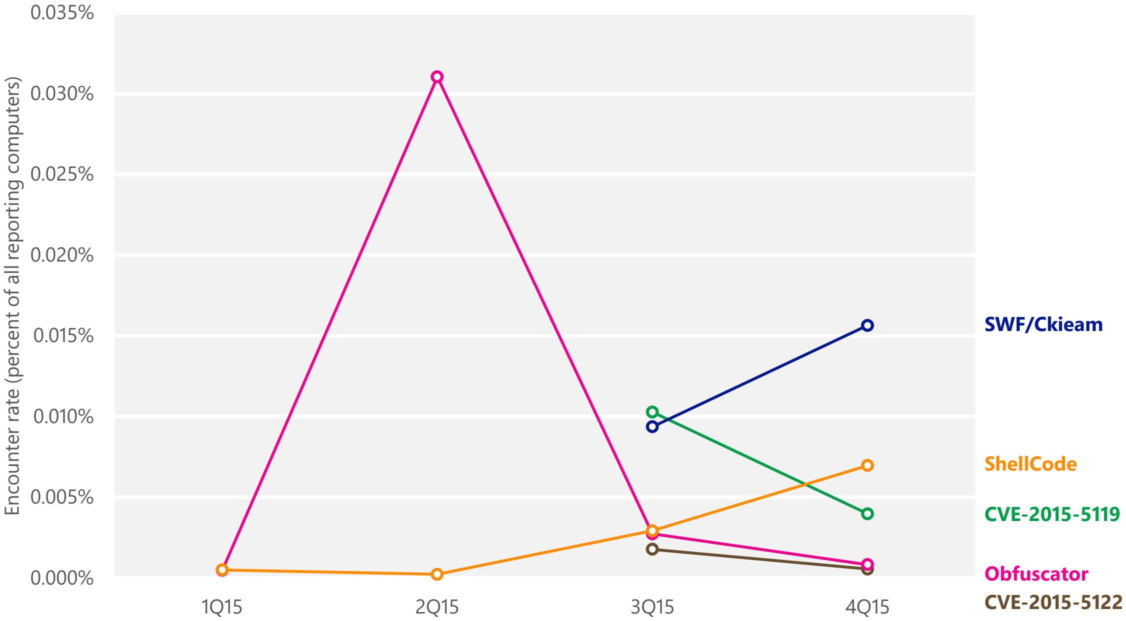


- Most detections of exploits that affect Adobe Reader and Adobe Acrobat were associated with the exploit family Win32/Pdfjsc, a detection for PDF files containing malicious JavaScript that targets CVE-2010-0188 and other vulnerabilities. Adobe released Security Bulletin APSB10-07 in February 2010 to address CVE-2010-0188. Pdfjsc and related exploits were particularly prevalent in eastern Europe. Pdfjsc mostly targets older Java vulnerabilities, so attackers may find it less useful as more computers are updated to newer versions of Java, which could explain the decrease in encounters over the past several quarters.
- CVE-2012-0158 is a remote code execution in Windows Common Controls that affects certain older versions of Microsoft Office. Microsoft released Security Bulletin MS12-027 in April 2015 to address the issue.
- CVE-2015-1641 is a memory corruption vulnerability in several versions of Microsoft Office and Microsoft Word that allows a remote attacker to execute arbitrary code via a malicious Rich Text Format (RTF document). Microsoft released Security Bulletin MS15-033 in April 2015 to address the issue.

Adobe Flash Player exploits

Figure 43 shows the prevalence of different Adobe Flash Player exploits by quarter.

Figure 43. Adobe Flash Player exploits detected and blocked by Microsoft real-time antimalware products in 2015

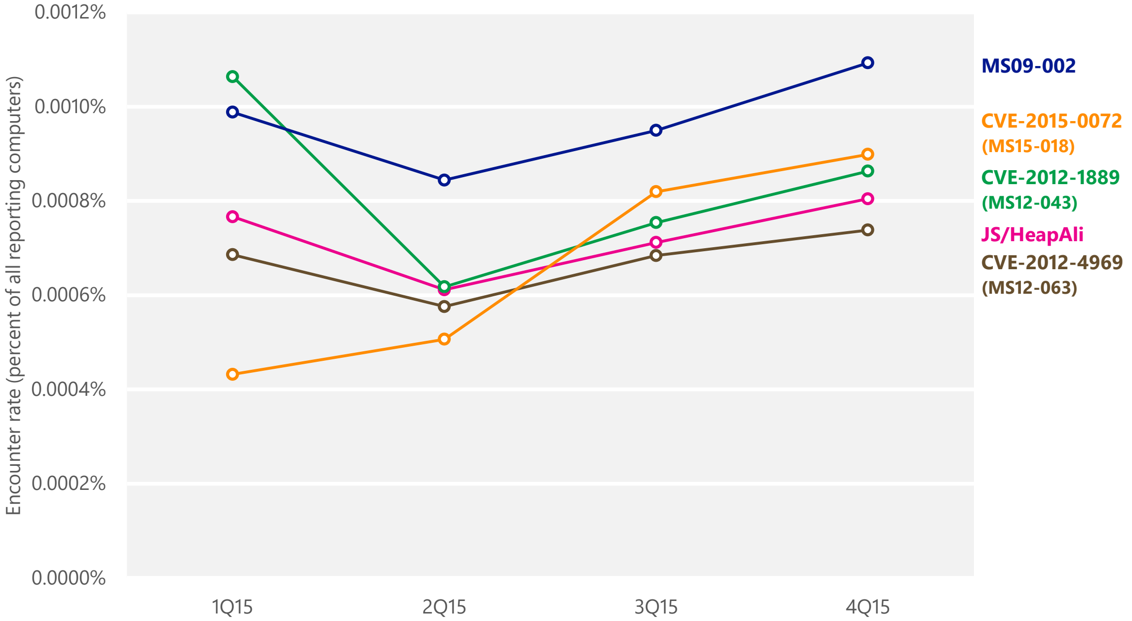


- Exploits targeting CVE-2015-5119, a use-after-free vulnerability in the ActionScript interpreter in some versions of Adobe Flash Player, was the most commonly encountered Flash Player exploit in 2H15. (SWF/Ckiem is another detection for CVE-2015-5119 exploits.) Adobe released Security Bulletin APSB15-16 in July to address the issue.
- After increasing sharply in 2Q15, encounters involving Obfuscator variants that target Adobe Flash Player declined to much lower levels in 3Q15, signaling a change in tactics on the part of attackers.

Browser exploits

Figure 44 shows the prevalence of different browser exploits by quarter.

Figure 44. Browser exploits detected and blocked by Microsoft real-time antimalware products in 2015

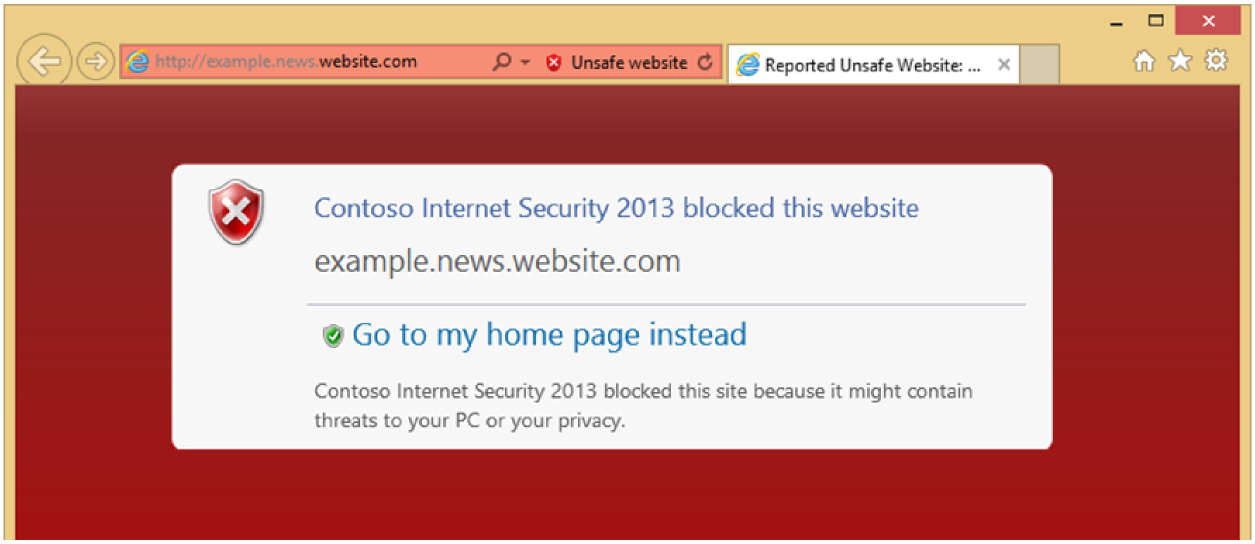


- Exploits targeting vulnerabilities addressed by Security Bulletin MS09-002, published by Microsoft in February 2009, accounted for the largest share of browser-related exploits encountered in 2H15. Of these, most exploits targeted CVE-2009-0075, an uninitialized memory corruption vulnerability in Internet Explorer 7.
- CVE-2015-0072 is a cross-site scripting (XSS) vulnerability in Internet Explorer versions 9 through 11 that can allow remote attackers to bypass the same-origin policy, which is intended to prevent malicious scripts on compromised pages from accessing resources located elsewhere. Microsoft released Security Bulletin MS15-018 in March 2015 to address the issue.
- CVE-2012-1889, a memory corruption vulnerability that affects older versions of Microsoft XML Core Services, was addressed by Microsoft with Security Bulletin MS12-043, released in July 2012.
- CVE-2012-4969, a use-after-free vulnerability in Internet Explorer versions 6 through 9, was addressed by Microsoft with Security Bulletin MS12-063, released in September 2012.

Exploit detection with Internet Explorer and IExtensionValidation

IExtensionValidation is an interface introduced in Internet Explorer 11 that real-time security software can implement to block ActiveX controls from loading on malicious pages. (Microsoft Edge, the newest Microsoft web browser and the default browser in Windows 10, does not support ActiveX plug-ins at all, and therefore does not use IExtensionValidation.) When Internet Explorer loads a webpage that includes ActiveX controls, if the security software has implemented IExtensionValidation, the browser calls the security software to scan the HTML and script content on the page before loading the controls themselves. If the security software determines that the page is malicious (for example, if it identifies the page as an exploit kit landing page), it can direct Internet Explorer to prevent individual controls or the entire page from loading.

Figure 45. Internet Explorer 11 can block pages that contain ActiveX controls if security software determines that the page is malicious



Contoso Internet Security 2013 blocked this website

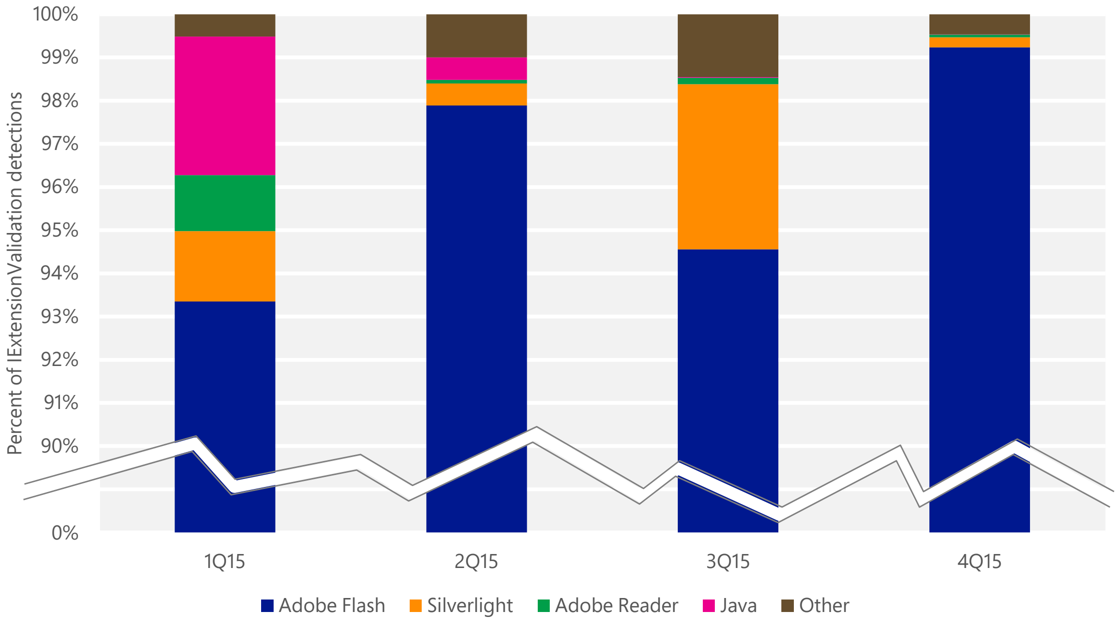
example.news.website.com

Go to my home page instead

Contoso Internet Security 2013 blocked this site because it might contain threats to your PC or your privacy.

Figure 46 shows the types of ActiveX controls identified on malicious webpages in Internet Explorer 11 for each quarter in 2015.

Figure 46. ActiveX controls detected on malicious webpages through IExtensionValidation in 2015, by control type



- Adobe Flash Player objects were the most commonly detected type of object hosted on malicious pages by an overwhelming margin in each of the past four quarters, from a low of 93.3 percent in 1Q15 to a high of 99.2 percent in 4Q15.

Detections of Java applets on malicious pages decreased to negligible levels by 4Q15.

- After accounting for almost half of object detections during some quarters in 2014, detections of Java applets on malicious pages decreased to negligible levels by 4Q15. A number of changes that have been made to Java and Internet Explorer over the past two years have made it much more difficult for attackers to take advantage of Java-based vulnerabilities, which is the most likely explanation for this significant decrease. (See "Java exploits" on page 69 for more information.)

- Detections of malicious Silverlight objects increased from 0.5 percent in 2Q15 to 3.8 percent in 3Q15, with most of the increase targeting CVE-2015-1671, a TrueType font parsing vulnerability addressed by Security Bulletin MS15-044 in May 2015. Silverlight object detections decreased again in 4Q15 as attackers focused on Flash Player almost exclusively.

Malware and unwanted software

Most attempts by malware to infect computers are unsuccessful. More than three-quarters of Internet-connected personal computers worldwide are protected by real-time security software that constantly monitors the computers and network traffic for threats and blocks them before they can infect the computers, if possible. Therefore, a comprehensive understanding of the malware landscape requires consideration of infection attempts that are blocked as well as infections that are removed.

Microsoft uses two different metrics to measure malware and unwanted software prevalence:[14]

- *Encounter rate* is simply the percentage of computers running Microsoft real-time security products that report a malware encounter.[15] For example, the encounter rate for the malware family Win32/Banload in Brazil in 4Q15 was 4.5 percent. This data means that, of the computers in Brazil that were running Microsoft real-time security software in 4Q15, 4.5 percent reported encountering the Banload family, and 95.5 percent did not. Encountering a threat does not mean the computer has been infected. Only computers whose users have opted in to provide data to Microsoft are considered when calculating encounter rates.[16]

[14] Microsoft regularly reviews and refines its data collection methodology to improve its scope and accuracy. For this reason, the statistics presented in this volume of the *Microsoft Security Intelligence Report* may differ slightly from comparable statistics in previous volumes.

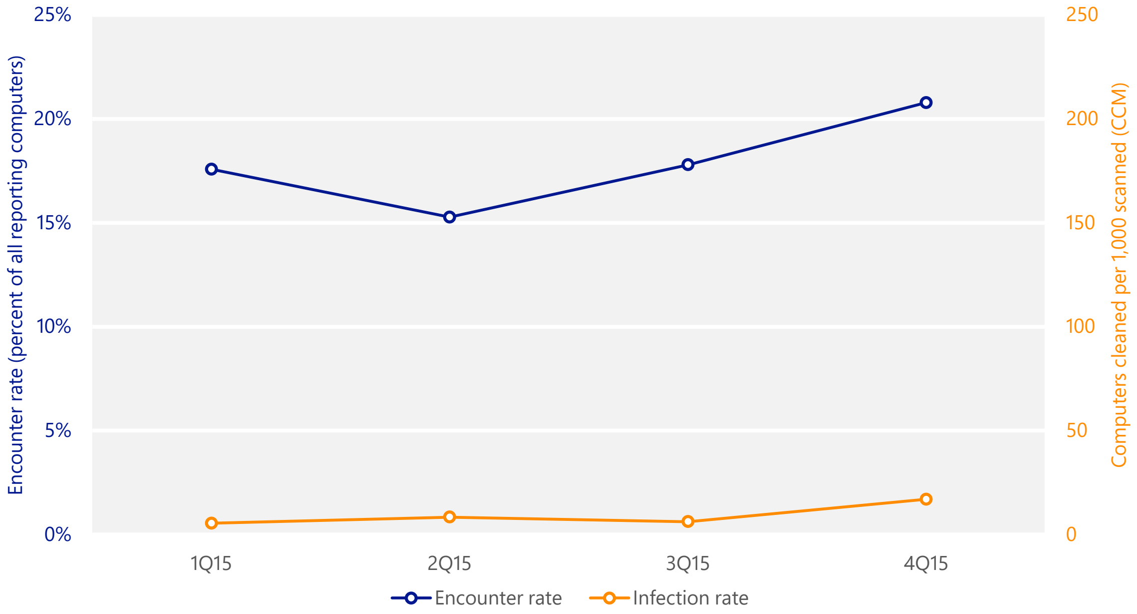
[15] Encounter rate does not include threats that are blocked by a web browser before being detected by antimalware software. In particular, IExtensionValidation in Internet Explorer 11 enables security software to block pages that contain exploits from loading. (See "Exploit detection with Internet Explorer and IExtensionValidation" on page 77 for information about IExtensionValidation and the threats it blocks.) For this reason, encounter rate figures may not fully reflect all of the threats encountered by computer users.

[16] For information about the products and services that provide data for this report, see "Appendix B: Data sources" on page 157.

- *Computers cleaned per mille*, or *CCM*, is an infection rate metric that is defined as the number of computers cleaned for every 1,000 unique computers that run the Malicious Software Removal Tool (MSRT), a free tool distributed through Microsoft update services that removes more than 200 highly prevalent or serious threats from computers. Because it is not a real-time tool, the MSRT only detects and removes threats that are already present on the computer; it does not block infection attempts as they happen.

Figure 47 illustrates the difference between these two metrics.

Figure 47. Worldwide encounter and infection rates in 2015, by quarter

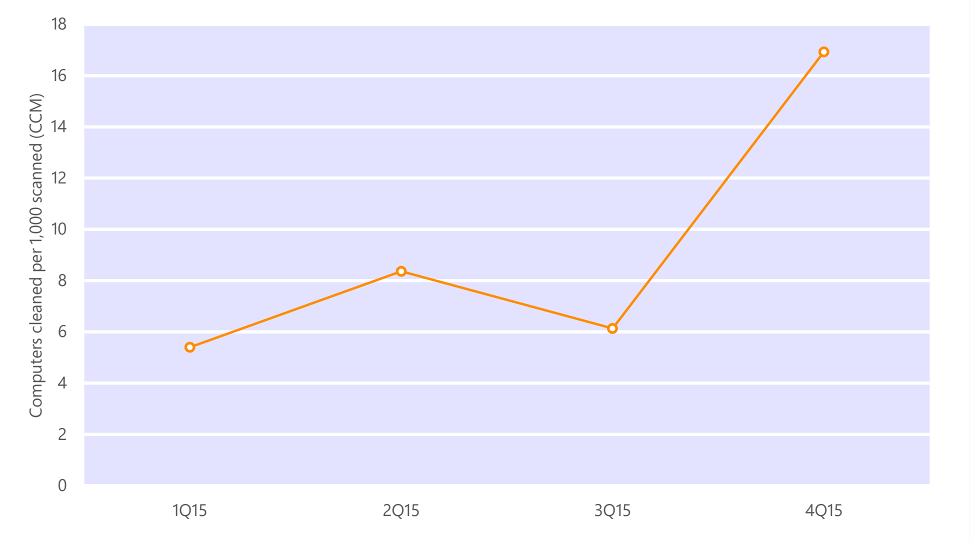


Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

As Figure 47 shows, and as one would expect, encounters are much more common than infections. On average, about 17.9 percent of reporting computers worldwide encountered threats over the past four quarters. At the same time, the MSRT removed threats from about 9.2 out of every 1,000 computers, or 0.92 percent. Together, encounter and infection rate information can help provide a broader picture of the threat landscape by offering different perspectives on how threats propagate and how computers get infected.

Diplugem and infection rates

Figure 48. Worldwide infection rates in 2015, by quarter



The worldwide infection rate increased 175.9 percent in the final quarter of the year, from a CCM of 6.1 in 3Q15 to 16.9 in 4Q15. Almost all of this increase was due to the unwanted software family Win32/Diplugem, a browser modifier that shows extra advertisements as the user browses the web. The CCM for Diplugem alone in 4Q15 was 11.7, nine times as high as the CCM for the next most prevalent family, Win32/Gamarue.

Diplugem was added to the MSRT in October 2015, causing a sharp increase in the worldwide infection rate as the MSRT detected and removed a backlog of millions of Diplugem infections that may have been present for many months or longer. Diplugem was the family most commonly detected and removed by the MSRT in 4Q15 by a large margin on all versions of Windows and in most countries and regions. Microsoft expects the worldwide infection rate to decrease to more typical levels in 2016 as the existing backlog of Diplugem infections is dealt with.

Brantall, Rotbrow, and Filcout

Where noted, the figures in this report omit detections of Win32/Brantall, Win32/Rotbrow, and Win32/Filcout. These three families were involved in an incident in which a rogue developer with access to commercial source code modified the source code to serve as a stealth distribution method for malware without being detected by major security software vendors. When the modification was discovered, it resulted in a significant installed base of commercial software being reclassified as malicious, which had an outsized effect on infection rates. Microsoft believes that the unmodified infection and encounter figures do not create an accurate picture of the worldwide threat landscape over the past year and a half. As a result, totals for the Brantall, Filcout, and Rotbrow families have been removed from the infection and encounter figures presented here where appropriate, as noted.

See "The Sefnit saga: a timeline" on pages 57–64 of *Microsoft Security Intelligence Report, Volume 17 (January–June 2014)*, available from the Microsoft Download Center, for a more in-depth explanation of the incident, along with detection statistics and a timeline of events.

Malware and unwanted software worldwide

The telemetry data generated by Microsoft security products from computers whose administrators or users choose to opt in to provide data to Microsoft includes information about the location of the computer, as determined by IP geolocation. This data makes it possible to compare infection and encounter rates, patterns, and trends in different locations around the world.[17]

[17] For more information about this process, see the entry "Determining the Geolocation of Systems Infected with Malware" (November 15, 2011) in the Microsoft Cyber Trust Blog (blogs.microsoft.com/cybertrust).

Figure 49. Encounter rate trends for the locations with the most computers reporting malware and unwanted software encounters in 2H15, by number of computers reporting

Country/Region	1Q15	2Q15	3Q15	4Q15
United States	11.2%	9.7%	10.8%	12.5%
Brazil	21.0%	20.4%	29.2%	34.4%
China	13.5%	13.6%	14.9%	18.9%
Russia	23.2%	17.7%	22.8%	28.7%
France	16.0%	13.3%	18.8%	19.4%
Germany	11.2%	8.9%	12.2%	13.8%
United Kingdom	12.9%	11.7%	11.9%	13.9%
Italy	19.8%	15.3%	19.8%	22.3%
Canada	14.2%	12.5%	13.1%	15.5%
Japan	5.6%	5.4%	6.3%	7.8%
Worldwide	*17.6%*	*15.3%*	*17.8%*	*20.8%*

Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

- Locations in Figure 49 are ordered by the number of computers reporting detections in 2H15.
- Consistent with the general increase in encounter rate shown in Figure 47, all of the locations in Figure 49 experienced increased encounter rates from 2Q15 to 3Q15, and from 3Q15 to 4Q15.
- The encounter rate in the United States was about 40 percent (or approximately 8 percentage points) lower than the worldwide encounter rate in 2H15. The browser modifier Win32/Diplugem and the exploit kit JS/Axpergle were the most common families encountered in the US during the period. The browser modifier Win32/Suptab, the most commonly encountered threat family worldwide in 2H15, only ranked 20th in the US, far lower than it ranked in any other location in Figure 49 except China and Canada; the worm family Win32/Gamarue, ranked third worldwide, only ranked 70th in the US.

  Families that were unusually common in the US included the rogue security software family JS/FakeCall (ranked third in the US, 35th worldwide) and the adware family Win32/Peapoon (15th in the US, 47th worldwide). See "Threat families" beginning on page 97 for more information about commonly encountered malware and unwanted software families.

- The encounter rate in Brazil was about 65 percent higher than the worldwide encounter rate in 2H15. Encounters in Brazil were led by Suptab and the downloader/dropper families Win32/Sventore and Win32/Banload. (See "Win32/Banload and Banking Malware" on page 21 of *Microsoft Security Intelligence Report, Volume 19 (January–June 2015)* for more information about Banload and related families in Brazil.) Families that were unusually common in Brazil included Banload (ranked third in Brazil, 49th worldwide), the software bundler Win32/Fourthrem (13th in Brazil, 107th worldwide), and the trojan family Win32/Banker (15th in Brazil, 88th worldwide).

> The encounter rate in Brazil was about 65 percent higher than the worldwide encounter rate in 2H15.

- The encounter rate in China was about 13 percent lower than the worldwide encounter rate in 2H15. The threat landscape in China is typically dominated by malware families that are much less common worldwide, and 2H15 was no exception. Several of the most prevalent families worldwide, including Suptab, Axpergle, the software bundler Win32/Outbrowse, and the browser modifier Win32/CouponRuc were not among the 100 most commonly encountered families in China in 2H15.

  Unusually common families in China included the viruses DOS/JackTheRipper (ranked third in China, 70th worldwide) and ALisp/Bursted (12th in China, 104th worldwide) and the worm ALisp/Kenilfe (ninth in China, 125th worldwide). Only two of the most common families in China were unwanted software families, and they, too, were largely confined to China: the browser modifier Win32/Hao123 ranked fourth in China and 68th worldwide, and the software bundler Win32/Xiazai ranked fifth in China and 63rd worldwide.

- The encounter rate in Russia was about 33 percent higher than the worldwide encounter rate in 2H15. Five of the ten most commonly encountered families in Russia in 2H15 were trojans, including Win32/Peals, Win32/Skeeyah, Win32/Dynamer, and Win32/Spursint. The exploit kit family Axpergle, ranked tenth worldwide, only ranked 243rd in Russia. Families that were unusually common in Russia in 2H15 included the downloader families Win32/Ogimant (ranked seventh in Russia, 75th worldwide) and Win32/Mytonel (ranked 14th in Russia, 90th worldwide).

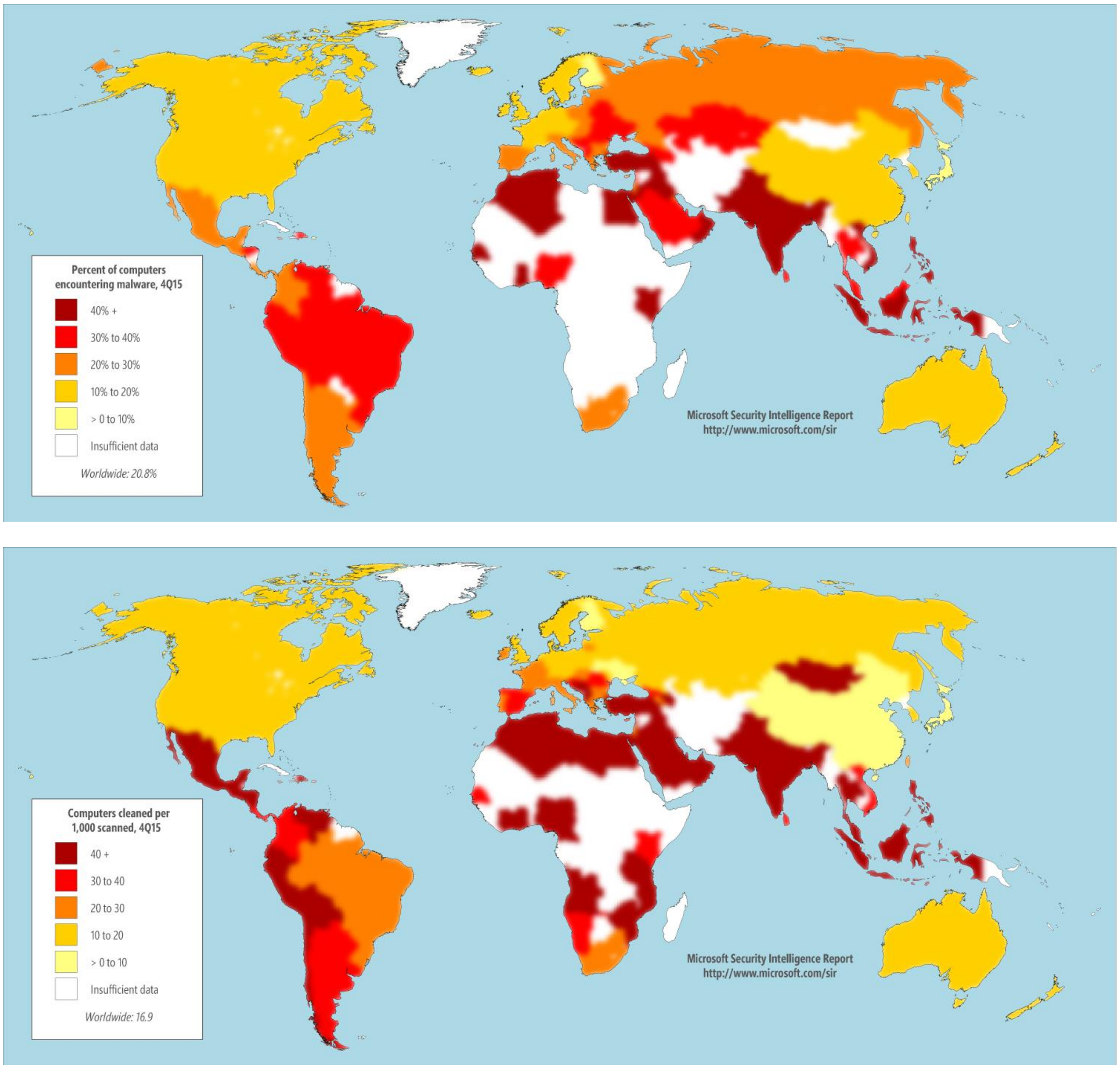
- The encounter rate in France was close to the worldwide average in 2H15, as was the overall mix of threats: all of the ten most common families in France were also among the top 20 threats worldwide. The most significant difference involved Gamarue, which ranked third worldwide but only 82nd in France.

- Gamarue was also relatively uncommon in Germany, the UK, and Italy in 2H15, all of which otherwise displayed a similar mix of threats to that of the world overall. The adware family Win32/Putalol was unusually common in Germany, where it ranked tenth in 2H15, compared to 69th worldwide. Encounter rates in these locations ranged between 33 percent lower and 9 percent higher than the world overall in 2H15.
- The encounter rate in Canada was about 26 percent lower than the worldwide encounter rate in 2H15. The list of the most common families in Canada was similar to that of the United States. Gamarue, which ranked third worldwide, ranked 81st in Canada. As in the US, unusually common families in Canada included Fakecall (fifth in Canada, 35th worldwide) and Peapoon (14th in Canada, 47th worldwide).
- The encounter rate in Japan was about 64 percent lower than the worldwide encounter rate in 2H15, giving it one of the lowest encounter rates of any country or region (see page 90 for more information). Despite Japan's geographic and cultural distance from most of the locations listed in Figure 49, the mix of threats encountered there was quite similar to that of the world overall; all of the 15 most commonly encountered families in Japan in 2H15 were also among the top 20 families encountered worldwide.

> The mix of threats encountered in Japan was quite similar to that of the world overall.

For a different perspective on threat patterns worldwide, Figure 50 shows the infection and encounter rates in locations around the world in 4Q15.

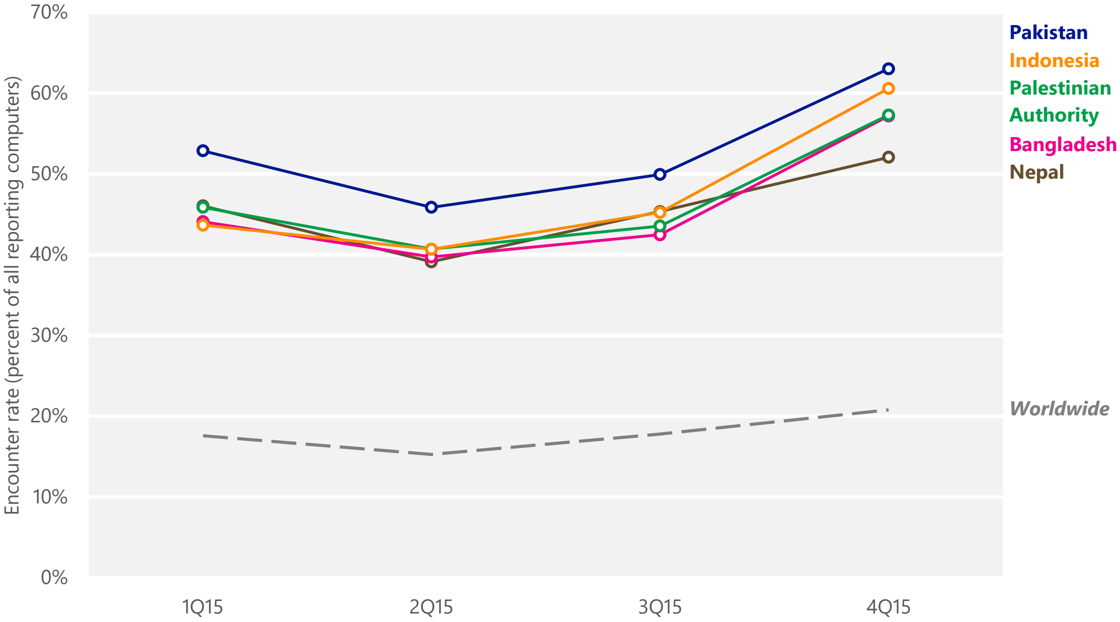
Figure 50. Encounter rates (top) and infection rates (bottom) by country/region in 4Q15


Percent of computers encountering malware, 4Q15
40% +
30% to 40%
20% to 30%
10% to 20%
> 0 to 10%
Insufficient data
Worldwide: 20.8%
Microsoft Security Intelligence Report
http://www.microsoft.com/sir
Computers cleaned per 1,000 scanned, 4Q15
40 +
30 to 40
20 to 30
10 to 20
> 0 to 10
Insufficient data
Worldwide: 16.9
Microsoft Security Intelligence Report
http://www.microsoft.com/sir

Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

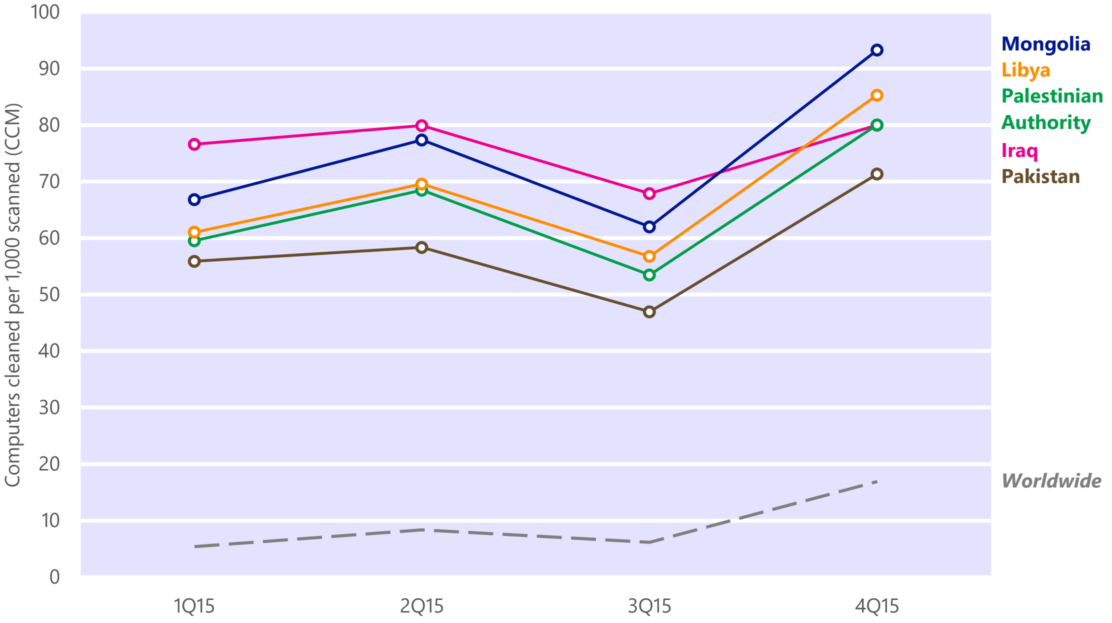
The next several figures illustrate trends for specific locations around the world with particularly high or low incidences of threat detection. Figure 51 and Figure 52 show trends for the locations with the highest rates of detection as determined by encounter rate and CCM, respectively.

Figure 51. Trends for the five locations with the highest encounter rates in 2H15 (100,000 reporting computers minimum)



Figures do not include Brantall, Rotbrow, and Filcout. See “Brantall, Rotbrow, and Filcout” on page 82 for more information.

Figure 52. Trends for the five locations with the highest infection rates in 2H15, by CCM (100,000 MSRT computers minimum)



Figures do not include Brantall, Rotbrow, and Filcout. See “Brantall, Rotbrow, and Filcout” on page 82 for more information.

- The locations with the highest encounter rates were Pakistan, Indonesia, the Palestinian territories, Bangladesh, and Nepal.

- Pakistan, Indonesia, Bangladesh, and Nepal were also among the locations with the highest encounter rates in 1H15.
- As in 1H15, exploit kits were relatively rare in the locations with the highest encounter rates. JS/Axpergle, the most commonly encountered exploit kit worldwide in 2H15 and the 10th most commonly encountered family overall, ranked no higher than 100th in any of the locations with the highest encounter rates.
- Families that were unusually common in Pakistan included the worm families Win32/Ippedo (ranked third in Pakistan, 28th worldwide) and Win32/Nuqel (ninth in Pakistan, 71st worldwide).
- Families that were unusually prevalent in Indonesia included the exploit family Win32/CplLnk (ranked sixth in Indonesia, 37th worldwide) and the virus family Win32/Virut (seventh in Indonesia, 43rd worldwide).

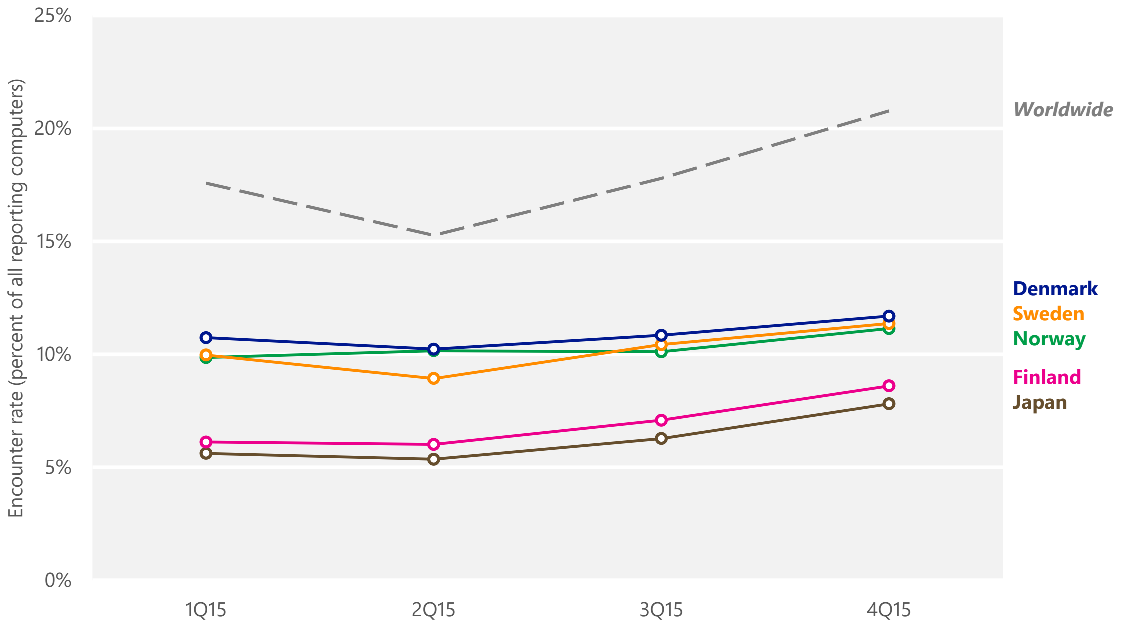
About 15 percent of all Jeefo encounters took place in Nepal.

- Families that were unusually common in Bangladesh included Ippedo (ranked first in Bangladesh, 28th worldwide), CplLnk (eighth in Bangladesh, 37th worldwide), and the virus family Win32/Sality (tenth in Bangladesh, 27th worldwide).
- Families that were unusually common in Nepal included the virus family Win32/Jeefo (ranked 10th in Nepal, 239th worldwide). About 15 percent of all Jeefo encounters worldwide in 2H15 took place in Nepal, where the encounter rate for the family was about 20 times higher than in any other country or region.

- The locations with the highest infection rates were Mongolia, Libya, the Palestinian territories, Iraq, and Pakistan.
    - Win32/Diplugem, the family removed from the most computers worldwide in 4Q15 by a significant margin, had a less dramatic impact in these locations because of their generally high infection rates overall. In Mongolia, in fact, Diplugem was only the second most common infecting family in 4Q15, behind Win32/Gamarue. (See “Diplugem and infection rates” on page 81 for more information about Diplugem and its effect on CCM.)

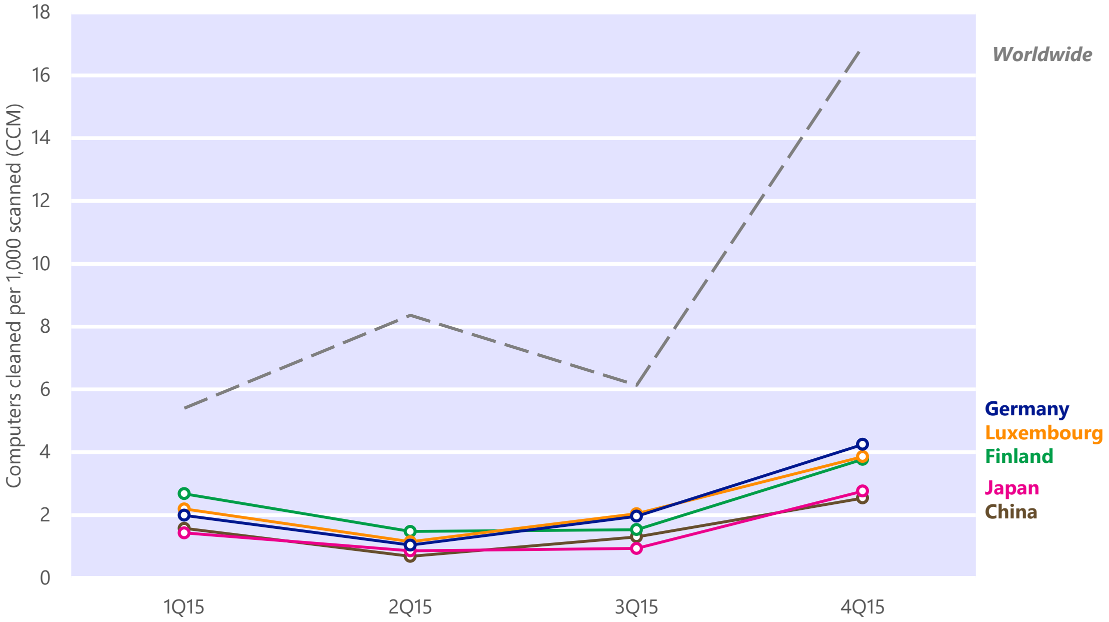
- The worm family Win32/Vobfus, the 25th most common infecting family worldwide, was unusually common in Mongolia (where it ranked 11th), Libya (ninth), and the Palestinian territories (15th).
- Infections involving the backdoor family MSIL/Bladabindi, which ranked 14th among infecting families worldwide, were particularly common in Iraq and Libya (where it ranked third) and the Palestinian territories (where it ranked sixth).
- Gamarue was particularly prevalent in Mongolia, where it was found to be infecting about 35 out of every 1000 computers running the MSRT in 2H15.
- Families that were unusually prevalent in Iraq included the worm family Win32/Wecykler (ranked fifth in Iraq, 59th worldwide) and the trojan family Win32/Sulunch (13th in Iraq, 141st worldwide).
- Infecting families that were unusually prevalent in Pakistan included the worm family Win32/Tupym (ranked 13th in Pakistan, 110th worldwide) and the backdoor family Win32/Bifrose (15th in Pakistan, 115th worldwide).

Figure 53. Trends for locations with low encounter rates in 2H15 (100,000 reporting computers minimum)



Figures do not include Brantall, Rotbrow, and Filcout. See “Brantall, Rotbrow, and Filcout” on page 82 for more information.

Figure 54. Trends for locations with low infection rates in 2H15, by CCM (100,000 reporting computers minimum)



Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

- The Nordic countries, including Denmark, Finland, Iceland, Norway, and Sweden, have perennially been among the healthiest locations in the world with regard to malware exposure, as has Japan. In 2H15, the infection and encounter rates for these locations were typically about half of the worldwide averages. (See the blog entry series "Lessons from Least Infected Countries" at blogs.technet.com/b/security/p/series-lessons-from-least-infected-countries.aspx for more information about locations that typically have low infection and encounter rates.)
- All of the locations shown in Figure 53 and Figure 54 had similar encounter and infection statistics in 2H15, with relatively few families that were particularly common or uncommon compared to the world as a whole. A significant exception was Win32/Gamarue, a worm that is particularly prevalent in parts of the Middle East and Asia. Gamarue was the third most commonly encountered family worldwide in 2H15, but ranked 38th in Japan, and 74th or lower in Sweden, Finland, Denmark, and Norway.
- As in most of the rest of the world, the browser modifier Win32/Diplugem heavily influenced the 4Q15 infection rates in all five locations shown in Figure 54. The MSRT found Diplugem infecting between 5.5 and 13.2 of every 1000 computers in 4Q15 in all five places, compared to about 0.5 of every 1000 computers for the next most prevalent family in each location.

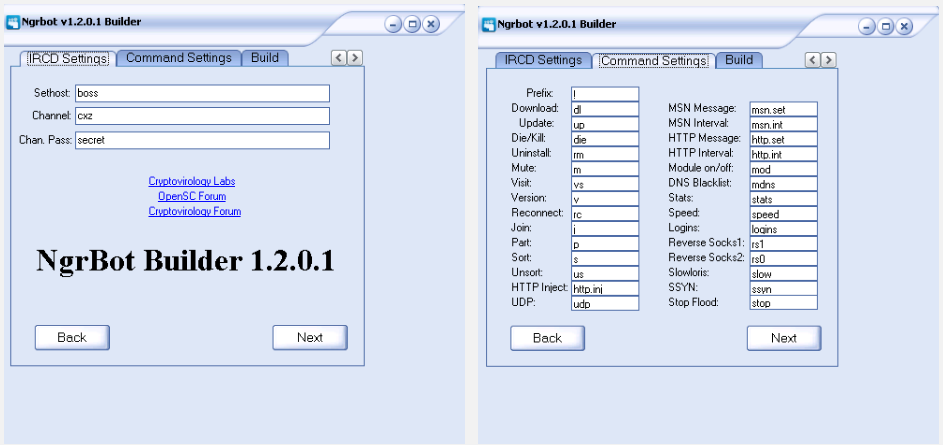
(See "Diplugem and infection rates" on page 81 for more information about Diplugem and its effect on CCM.)

Microsoft and partners disrupt Dorkbot botnets

On December 2, 2015, Microsoft and its partners in industry and law enforcement announced the disruption of Win32/Dorkbot, a "botnet-in-a-box" malware family that had infected more than one million computers in more than 190 countries and regions.

Dorkbot is the Microsoft detection name for NgrBot, a commercial botnet kit that prospective computer criminals buy from its creator through underground online forums. The kit includes a bot builder utility as well as documentation on how to create a Dorkbot botnet. The bot malware is spread in a number of different ways, including removable drives, social networks, or drive-by downloading via an exploit kit. The purchaser controls the resulting botnet over Internet Relay Chat (IRC), and can command bots to download other malware to the infected computer, to spread to other computers, or take other malicious actions.

Figure 55. The Win32/Dorkbot administrative interface



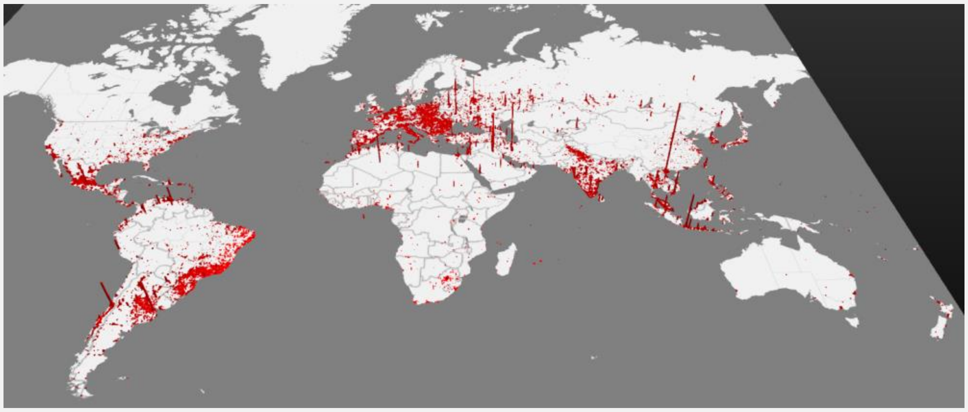
Sethost: boss
Channel: cxz
Chan. Pass: secret

Prefix:	!		
Download:	dl	MSN Message:	msn.set
Update:	up	MSN Interval:	msn.int
Die/Kill:	die	HTTP Message:	http.set
Uninstall:	rm	HTTP Interval:	http.int
Mute:	m	Module on/off:	mod
Visit:	vs	DNS Blacklist:	mdns
Version:	v	Stats:	stats
Reconnect:	rc	Speed:	speed
Join:	j	Logins:	logins
Part:	p	Reverse Socks1:	rs1
Sort:	s	Reverse Socks2:	rs0
Unsort:	us	Slowloris:	slow
HTTP Inject:	http.inj	SSYN:	ssyn
UDP:	udp	Stop Flood:	stop

The Microsoft Malware Protection Center (MMPC) and the Microsoft Digital Crimes Unit (DCU) led the analysis of the Dorkbot malware in partnership with ESET and Computer Emergency Response Team Polska (CERT Polska, NASK). Microsoft activated a Coordinated Malware Eradication (CME) campaign to coordinate the takedown, and provided research help, telemetry, and other assistance to a number of industry partners and law enforcement agencies, including CERT Polska, ESET, the Canadian Radio-television and Telecommunications Commission (CRTC), the Department of Homeland

Security's United States Computer Emergency Readiness Team (DHS/USCERT), Europol, the Federal Bureau of Investigation (FBI), Interpol, and the Royal Canadian Mounted Police (RCMP).

Figure 56. Dorkbot-infected computers connecting to the sinkhole during the first week of the takedown in December 2015



During the six months prior to the takedown, Microsoft detected Dorkbot on an average of 100,000 infected computers each month, with the top 10 countries accounting for 61 percent of all infected computers.

Figure 57. Computers infected by Dorkbot, May–October 2015

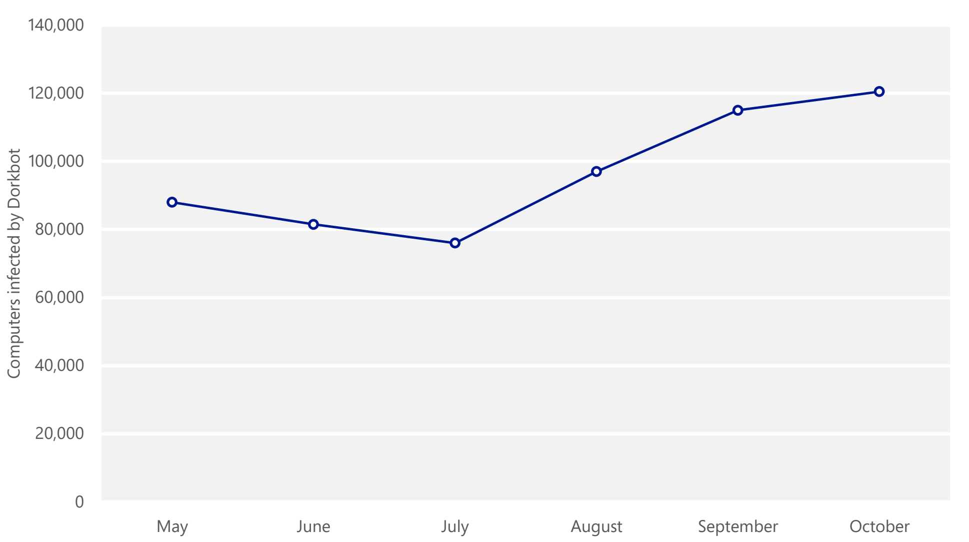
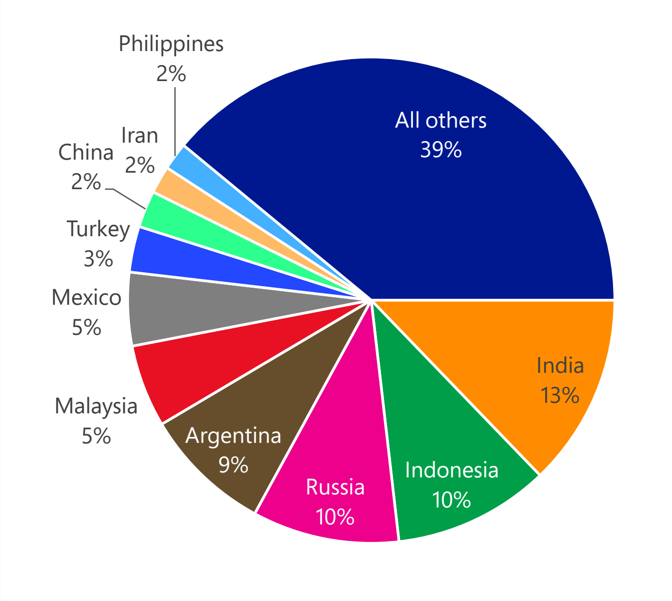


Figure 58. Top countries with Dorkbot infections, May–October 2015



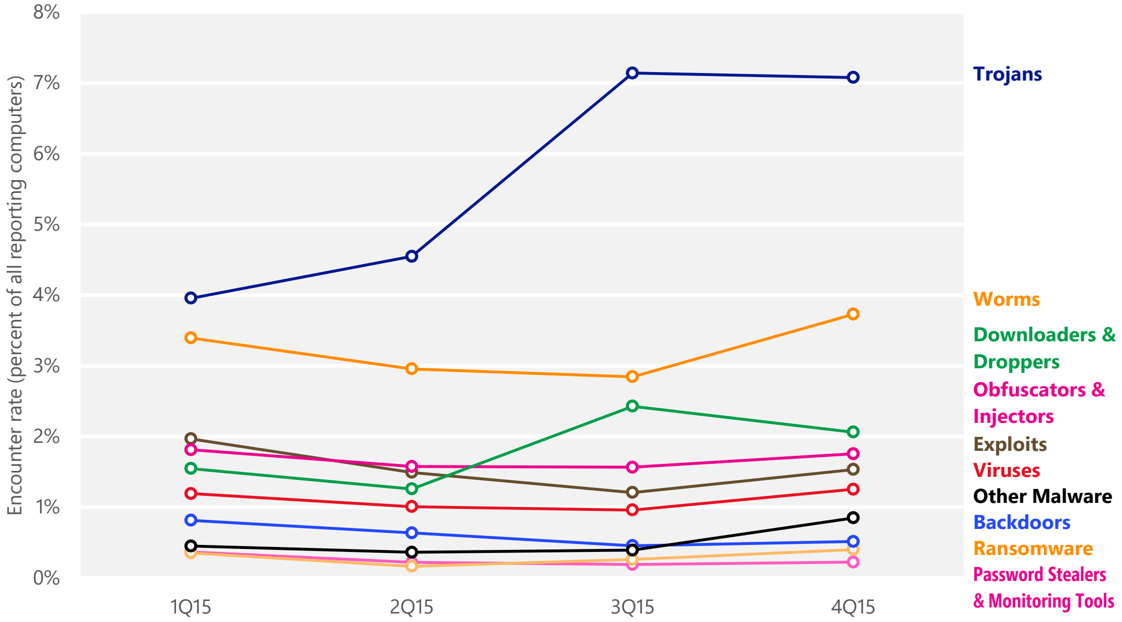
For more information about Dorkbot and the takedown effort, see the following entries on the MMPC blog at blogs.technet.com/mmpc:

- MSRT March 2012: Breaking bad (March 12, 2012)
- An analysis of Dorkbot's infection vectors (part 1) (November 14, 2012)
- An analysis of Dorkbot's infection vectors (part 2) (November 21, 2012)
- Microsoft assists law enforcement to help disrupt Dorkbot botnets (December 2, 2015)

Threat categories

The 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 categories based on similarities in function and purpose.

Figure 59. Encounter rates for significant malware categories in 2015



Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

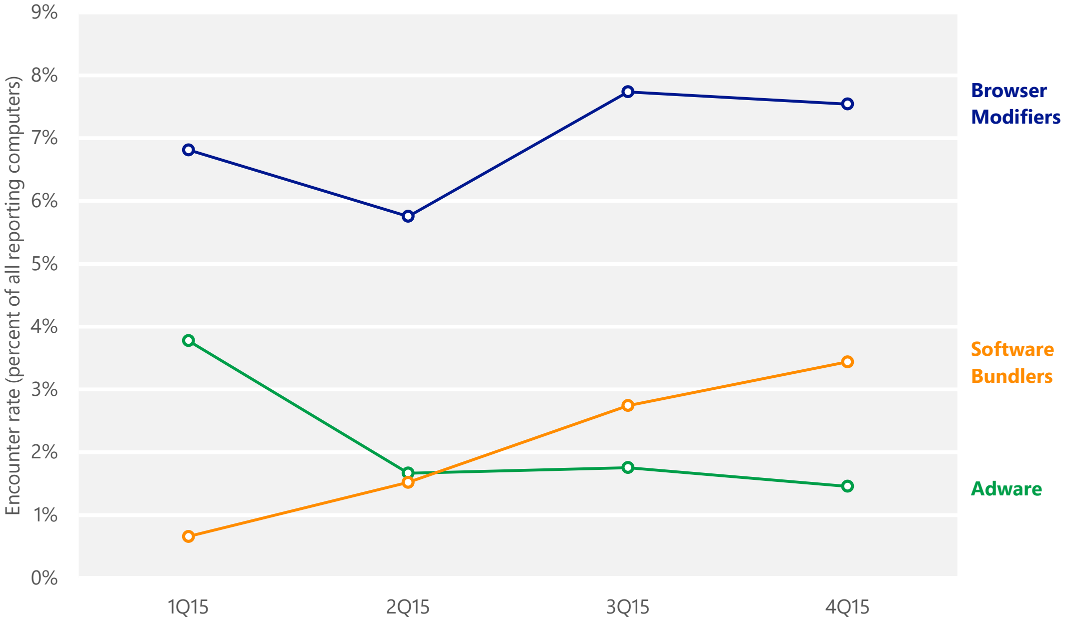
- Encounters involving trojans increased 57 percent from 2Q15 to 3Q15 and remained at an elevated level through the end of the year. Much of the rise was due to increased detections of Win32/Peals, Win32/Skeeyah, Win32/Colisi, and Win32/Dynamer, as well as a pair of newly detected trojans, Win32/Dorv and Win32/Spursint. See "Threat families" beginning on page 97 for more information about these and other malware and unwanted software families.

- Increased detections of Win32/Gamarue were principally responsible for the rise in encounters involving worms in 4Q15.

- Encounters involving downloaders and droppers increased significantly in 3Q15 before retreating slightly in 4Q15. Almost all of the increase was due to Win32/Sventore, which first appeared in the third quarter and was responsible for more than a third of the downloader/dropper encounters that quarter.

- The other categories of malware all remained relatively stable throughout 2H15, with most showing small increases in the fourth quarter.

Figure 60. Encounter rates for unwanted software categories in 2015



Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

- Two new browser modifiers, Win32/Diplugem and Win32/SupTab, were primarily responsible for the increased encounter rate for that category in 3Q15. See "Threat families" beginning on page 97 for more information about these and other malware and unwanted software families.
- Encounters involving software bundlers rose throughout 2H15, primarily because of increased detections of Win32/OutBrowse beginning in 3Q15 and because of two new software bundlers, Win32/Mizenota and Win32/Dowadmin, in 4Q15.

> Diplugem and SupTab were largely responsible for the increased encounter rate for browser modifiers.

Threat categories by location

Significant differences exist in the types of threats that affect users in different parts of the world. The spread of malware can be highly dependent on language and socioeconomic factors as well as on 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 world.

Figure 61 shows the relative prevalence of different categories of malware in several locations around the world in 4Q15.

Figure 61. Threat category prevalence worldwide and in the 10 locations with the most computers reporting encounters in 4Q15

Category	Worldwide	United States	Brazil	China	Russia	France	Germany	United Kingdom	Italy	Canada	Japan
Browser Modifiers	7.6%	9.1%	11.8%	0.6%	7.0%	14.3%	8.7%	10.9%	15.3%	11.3%	4.2%
Trojans	7.1%	4.2%	12.7%	10.2%	20.8%	5.7%	4.3%	4.4%	7.0%	5.1%	1.5%
Worms	3.3%	0.6%	8.9%	5.6%	4.6%	1.9%	1.1%	0.8%	3.9%	0.6%	0.7%
Software Bundlers	3.1%	1.7%	1.5%	0.2%	0.5%	2.2%	0.9%	2.3%	2.5%	2.5%	0.5%
Downloaders & Droppers	2.2%	2.3%	6.5%	3.2%	6.6%	2.8%	1.5%	3.2%	3.1%	3.3%	0.4%
Obfuscators & Injectors	1.7%	1.0%	5.3%	5.2%	7.3%	1.9%	1.6%	1.7%	2.8%	1.6%	0.6%
Adware	1.6%	4.5%	7.1%	0.2%	5.2%	7.8%	4.1%	4.7%	7.2%	5.3%	2.0%
Exploits	1.4%	3.4%	2.4%	1.7%	1.3%	2.5%	3.2%	4.4%	4.3%	5.7%	3.2%
Viruses	1.1%	0.4%	2.2%	7.4%	1.5%	0.4%	0.3%	0.3%	0.8%	0.4%	0.2%
Other Malware	0.6%	0.9%	0.3%	1.2%	0.3%	0.5%	0.5%	0.6%	0.7%	1.5%	0.2%
Backdoors	0.5%	0.7%	1.4%	1.8%	2.0%	0.9%	0.6%	0.9%	1.0%	0.7%	0.3%
Ransomware	0.3%	0.6%	0.5%	0.0%	0.6%	0.7%	0.6%	0.4%	1.4%	0.7%	0.4%
Password Stealers & Monitoring Tools	0.2%	0.4%	1.0%	0.5%	0.8%	0.3%	0.4%	0.4%	0.6%	0.6%	0.3%

Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

- Within each row of Figure 61, 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. As in Figure 49 on page 83, the locations in the table are ordered by number of computers reporting detections in 2H15.
- France and Italy had high encounter rates for Browser Modifiers, led by Win32/SupTab and Win32/Diplugem.

- Russia had a significantly higher encounter rate for Trojans than the other locations listed in Figure 61, led by Win32/Peals, Win32/Skeeyah, Win32/Dynamer, and Win32/Spursint. All four trojans disproportionately affected computers in Russia and eastern Europe in 4Q15.
- Worms were particularly prevalent in Brazil, led by VBS/Jenxcus, Win32/Gamarue, and JS/Bondat.
- The highest encounter rates for Adware were in Brazil, France, and Italy. Win32/EoRezo was the most commonly encountered adware family in all three places.
- Viruses were particularly prevalent in China, led by DOS/JackTheRipper and Win32/Ramnit.

> Peals, Skeeyah, Dynamer, and Spursint disproportionately affected computers in Russia and eastern Europe in 4Q15.

See "Appendix C: Worldwide encounter and infection rates" on page 158 for more information about malware around the world. Also, see "Linking Cybersecurity Policy and Performance" at aka.ms/securityatlas for an in-depth examination of the socioeconomic factors that correlate with high infection rates in different parts of the world.

Threat families

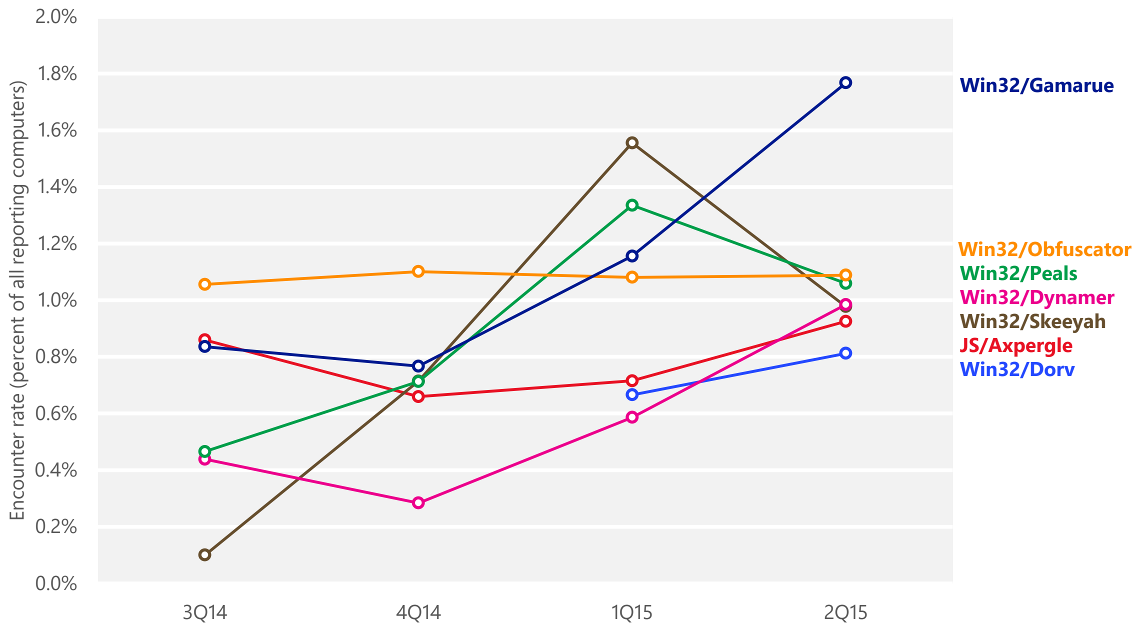
Figure 62 and Figure 63 show trends for the top malware families that were detected on computers by Microsoft real-time antimalware products worldwide in 2H15.

Figure 62. Quarterly trends for the top 10 malware families encountered by Microsoft real-time antimalware products in 2H15, shaded according to relative encounter rate

Rank	Family	Most significant category	1Q15	2Q15	3Q15	4Q15
1	Win32/Gamarue	Worms	0.84%	0.77%	1.16%	1.77%
2	Win32/Skeeyah	Trojans	0.10%	0.71%	1.56%	0.98%
3	Win32/Peals	Trojans	0.47%	0.71%	1.34%	1.06%
4	Win32/Obfuscator	Obfuscators & Injectors	1.06%	1.10%	1.08%	1.09%
5	JS/Axpergle	Exploits	0.86%	0.66%	0.72%	0.93%
6	Win32/Dynamer	Trojans	0.44%	0.28%	0.59%	0.98%
7	Win32/Dorv	Trojans	—	—	0.67%	0.81%
8	Win32/Sventore	Downloaders & Droppers	—	—	0.84%	0.60%
9	Win32/Colisi	Trojans	0.00%	0.01%	1.26%	0.01%
10	VBS/Jenxcus	Worms	0.93%	0.78%	0.55%	0.67%

Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

Figure 63. Encounter rate trends for a number of notable malware families in 2H15



- Win32/Gamarue, the most commonly encountered threat in 2H15, is a worm that is commonly distributed via exploit kits and social engineering. Variants have been observed stealing information from the local computer and communicating with command-and-control (C&C) servers managed by attackers. Gamarue was especially prevalent in southeast Asia and the Middle East, with computers in some heavily affected locations, such as

Indonesia, reporting Gamarue encounter rates in excess of 20 percent in 4Q15—close to the worldwide encounter rate for *all* threat families combined for the quarter. Despite its prevalence worldwide, Gamarue was rarely detected in most countries and regions in North America and western Europe, including the United States, where it was only the 70th most commonly encountered family in 2H15; Canada, where it ranked 81st; France, where it ranked 82nd; and Norway, where it ranked 86th.

For more information about Gamarue, see the following entries in the MMPC blog at blogs.technet.com/mmpc:

- Get gamed and rue the day… (October 25, 2011)
- The strange case of Gamarue propagation (February 27, 2013)

- Win32/Skeeyah, Win32/Peals, and Win32/Dynamer are generic detections for a variety of threats that share certain characteristics. All three detections disproportionately affected computers in Russia and Eastern Europe.
- Win32/Obfuscator is a generic detection for programs that have been modified by malware obfuscation tools. These tools typically use a combination of methods, including encryption, compression, and anti-debugging or anti-emulation techniques, to alter malware programs in an effort to hinder analysis or detection by security products. The output is usually another program that keeps the same functionality as the original program but with different code, data, and geometry.
- JS/Axpergle, a detection for the Angler exploit kit, is the only exploit-related family in the top ten in 2H15. See "Exploit kits" on page 66 for more information about Axpergle and other exploit kits.
- Win32/Sventore is a trojan that connects to a remote host, potentially to download other files or receive additional instructions from the attacker. Some Sventore variants make an effort to determine whether the computer is a virtual machine or exhibits other characteristics of a malware research environment, and terminates execution if they detect such characteristics.

> Axpergle, a detection for the Angler exploit kit, is the only exploit-related family in the top ten in 2H15.

- VBS/Jenxcus is a worm coded in VBScript that opens a backdoor on an infected computer, enabling an attacker to control it remotely. In addition to spreading via removable drives, Jenxcus is often transmitted via a fake

Adobe Flash Player update from spoofed YouTube webpages. Encounters involving Jenxcus decreased significantly after the Microsoft Digital Crimes Unit launched a takedown operation in June of 2014 that successfully disrupted the Jenxcus botnet. The original owners of the botnet subsequently left the project, but the Jenxcus code is now being used by other criminal organizations.

See "The Microsoft DCU and the legal side of fighting malware" on pages 29–32 of *Microsoft Security Intelligence Report, Volume 17 (January–June 2014)*, available from the Microsoft Download Center, for more information about the Microsoft takedown of the Jenxcus botnet. For additional technical information about Jenxcus, see the following entries in the MMPC blog (blogs.technet.com/mmpc):

- MSRT February 2014 – Jenxcus (February 11, 2014)
- Microsoft Digital Crimes Unit disrupts Jenxcus and Bladabindi malware families (June 30, 2014)

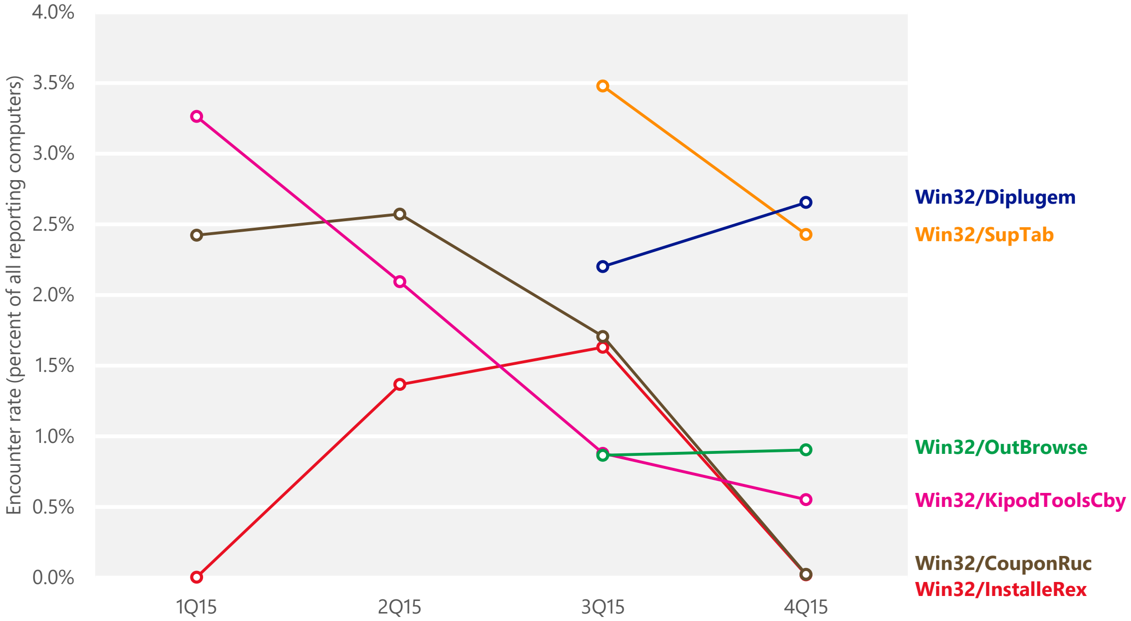
Figure 64 and Figure 65 show trends for the top unwanted software families that were detected on computers by Microsoft real-time antimalware products worldwide in 2H15. [18]

Figure 64. Quarterly trends for the top five unwanted software families encountered by Microsoft real-time antimalware products in 2H15, shaded according to relative encounter rate

Rank	Family	Most Significant Category	1Q15	2Q15	3Q15	4Q15
1	Win32/SupTab	Browser Modifiers	—	—	3.48%	2.43%
2	Win32/Diplugem	Browser Modifiers	—	—	2.20%	2.65%
3	Win32/OutBrowse	Software Bundlers	—	—	0.87%	0.90%
4	Win32/CouponRuc	Browser Modifiers	2.42%	2.57%	1.71%	0.02%
5	Win32/InstalleRex	Software Bundlers	0.00%	1.37%	1.63%	0.02%

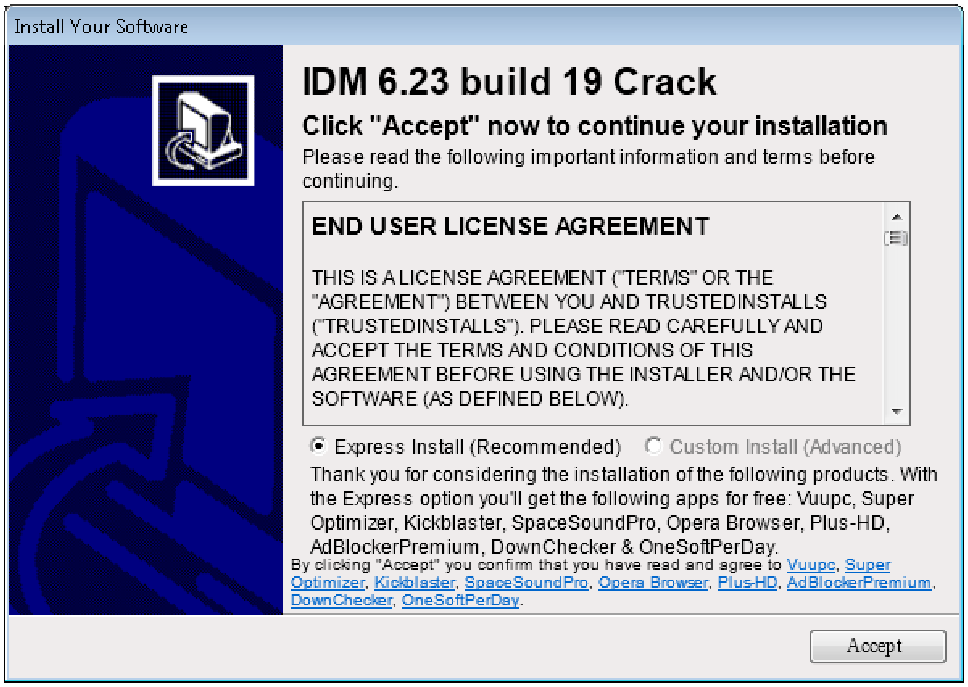
[18] Microsoft has published the criteria that the company uses to classify programs as unwanted software at www.microsoft.com/security/portal/mmpc/shared/objectivecriteria.aspx. For programs that have been classified as unwanted software, Microsoft provides a dispute resolution process to allow for reporting of potential false positives and to provide software vendors with the opportunity to request investigation of a rating with which they do not agree.

Figure 65. Encounter rate trends for the top unwanted software families in 2H15



- The three most commonly encountered unwanted software families in 2H15 were all first encountered in 3Q15.
  - Win32/SupTab is a browser modifier that installs itself and changes the browser's default search provider without obtaining the user's consent for either action.
  - Win32/Diplugem installs browser extensions without obtaining the user's consent. The browser extensions show extra advertisements as the user browses the web and can inject additional advertisements into web search results pages.
  - Win32/OutBrowse is a software bundler that installs additional unwanted programs alongside software that the user wishes to install. It can remove or hide the installation program's close button, leaving no option for users to close or decline the installation of offered applications.

Figure 66. Win32/OutBrowse installs software without a close or Cancel button to allow the user to decline installation



IDM 6.23 build 19 Crack

Click "Accept" now to continue your installation

Please read the following important information and terms before continuing.

END USER LICENSE AGREEMENT

THIS IS A LICENSE AGREEMENT ("TERMS" OR THE "AGREEMENT") BETWEEN YOU AND TRUSTEDINSTALLS ("TRUSTEDINSTALLS"). PLEASE READ CAREFULLY AND ACCEPT THE TERMS AND CONDITIONS OF THIS AGREEMENT BEFORE USING THE INSTALLER AND/OR THE SOFTWARE (AS DEFINED BELOW).

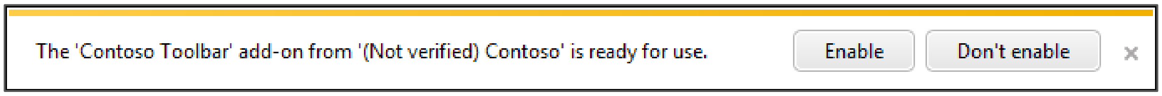
● Express Install (Recommended) ○ Custom Install (Advanced)

Thank you for considering the installation of the following products. With the Express option you'll get the following apps for free: Vuupc, Super Optimizer, Kickblaster, SpaceSoundPro, Opera Browser, Plus-HD, AdBlockerPremium, DownChecker & OneSoftPerDay.

By clicking "Accept" you confirm that you have read and agree to Vuupc, Super Optimizer, Kickblaster, SpaceSoundPro, Opera Browser, Plus-HD, AdBlockerPremium, DownChecker, OneSoftPerDay.

- Win32/KipodToolsCby is a browser modifier that bypasses user consent dialogs to install software without the user's explicit permission. Microsoft security products started detecting such browser modifiers in January after Microsoft changed its unwanted software detection criteria to include attempts to bypass user consent for actions such as installing new browser add-ons. The encounter rate for KipodToolsCby was highest in 1Q15 as Microsoft security products detected and removed large numbers of installations from previous periods, and decreased significantly in every subsequent quarter.

Figure 67. An add-on consent dialog bar from Internet Explorer 11. Add-ons that disable consent dialogs are now detected as unwanted software.



The 'Contoso Toolbar' add-on from '(Not verified) Contoso' is ready for use.

For more information about this change and its ramifications, see the following entries on the MMPC blog at blogs.technet.com/mmpc:

- Staying in control of your browser: New detection changes (October 17, 2014)
- A timeline of consent and control (December 11, 2014)

Threat families by platform

Malware does not affect all platforms equally. Some threats are spread by exploits that are ineffective against one or more operating system versions. Some threats are more common in parts of the world where specific platforms are more or less popular than elsewhere. In other cases, differences between platforms might be caused by simple random variation.

As Figure 68 demonstrates, the threats encountered by client and server platforms tend to be quite different.

Figure 68. The malware and unwanted software families most commonly encountered on supported Windows client and server platforms in 4Q15

	Client family	Most significant category	4Q15	Server family	Most significant category	4Q15
1	Win32/Diplugem	Browser Modifiers	2.59%	Win32/Peals	Trojans	0.61%
2	Win32/SupTab	Browser Modifiers	2.38%	Win32/Diplugem	Browser Modifiers	0.60%
3	Win32/Gamarue	Worms	1.66%	Win32/Crowti	Ransomware	0.47%
4	Win32/Obfuscator	Obfuscators & Injectors	1.10%	Win32/Dynamer	Trojans	0.46%
5	Win32/Peals	Trojans	1.02%	Win32/Dorv	Trojans	0.43%
6	Win32/Dynamer	Trojans	0.97%	Win32/Conficker	Worms	0.42%
7	Win32/Skeeyah	Trojans	0.97%	Win32/Sality	Viruses	0.33%
8	JS/Axpergle	Exploits	0.95%	Win32/Gamarue	Worms	0.32%
9	Win32/OutBrowse	Software Bundlers	0.89%	INF/Autorun	Obfuscators & Injectors	0.30%
10	Win32/Pokki	Browser Modifiers	0.86%	Win32/Skeeyah	Trojans	0.29%

Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

- Unwanted software was encountered significantly more often on client platforms than on server platforms. Four of the top ten families encountered by client versions of Windows in 2Q15—Win32/Diplugem, Win32/SupTab, Win32/OutBrowse, and Win32/Pokki—were unwanted software families, compared to just one (Diplugem) of the top ten families encountered on servers. The discrepancy reflects the very different ways servers are used to access the Internet, enforced by features such as Enhanced Security Configuration in Internet Explorer.
- Win32/Conficker was only the 39th most prevalent family overall in 4Q15, but ranked sixth on server platforms. Conficker is a worm that was disrupted several years ago, but continues to be encountered in enterprise

environments relatively frequently because of its use of a built-in list of common and weak passwords to spread between computers.

- PHP/SimpleShell was only the 709th most prevalent family overall in 4Q15, but ranked 23rd on server platforms. When installed on a compromised web server, it creates a webpage that an attacker can use to run shell commands on the server. A number of popular content management systems (CMSes) are written in the PHP scripting language, including WordPress, Drupal, and MediaWiki, and attackers often use PHP-based malware to compromise vulnerable servers for purposes such as sending spam and hosting exploit kit landing pages.

Figure 69 and Figure 70 demonstrate how detections of the most prevalent malware and unwanted software families in 4Q15 ranked differently on different operating system/service pack combinations.

Figure 69. The malware families most commonly encountered by Microsoft real-time antimalware solutions in 4Q15, and how they ranked in prevalence on different platforms

	Family	Most significant category	Rank					
			Win. Vista SP2	Win. 7 SP1	Win. 8 RTM	Win. 8.1 RTM	Win. 10 TH1	Win. 10 TH2
1	Win32/Gamarue	Worms	9	2	1	1	2	7
2	Win32/Obfuscator	Obfuscators & Injectors	4	9	4	2	1	3
3	Win32/Peals	Trojans	3	3	2	3	5	6
4	Win32/Dynamer	Trojans	8	5	7	5	4	1
5	Win32/Skeeyah	Trojans	6	6	5	4	3	2
6	JS/Axpergle	Exploits	124	1	307	26	156	48
7	Win32/Dorv	Trojans	1	4	8	11	6	4
8	VBS/Jenxcus	Worms	18	7	3	6	9	11
9	INF/Autorun	Obfuscators & Injectors	10	10	6	7	8	10
10	Win32/Spursint	Trojans	11	11	11	10	7	9
13	JS/FakeCall	Other Malware	2	13	15	9	15	23

Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

- Encounters involving JS/Axpergle, a detection for the Angler exploit kit and the only exploit-related family in the top ten in 2H15, were almost entirely confined to computers running Windows 7; although Axpergle ranked first on that platform, it ranked 26th on Windows 8.1, 48th on the Windows 10 November Update (sometimes referred to as TH2), and ranked outside the top 100 on all other supported client platforms. The malicious webpages that exploit kits use to spread malware often include scripts that detect certain aspects of the computer's computing environment and only present their exploits to computers that meet criteria specified by the attacker. The Angler exploit kit clearly affects Windows 7 far more than other platforms, which may partially be caused by the integration of Adobe Flash Player into Internet Explorer in Windows 8 and subsequent versions. The Angler exploit kit relies heavily on exploiting vulnerabilities in old, out-of-date versions of Flash Player, which must be installed as an add-on and updated separately from Internet Explorer in versions of Windows prior to Windows 8. Because Flash Player is integrated into Internet Explorer in Windows 8 and subsequent versions, it receives security updates through Windows Update and Microsoft Update along with other operating system components, which makes it easier for users to stay current on security updates for the component.

> Angler clearly affects Windows 7 far more than other platforms.

- The list of the most commonly encountered malware families was otherwise largely consistent from platform to platform. All of the ten most commonly encountered families apart from Axpergle were within the top 20 families on every supported platform. Windows Vista (the oldest currently supported client platform) and the Windows 10 November Update (the newest) displayed the most dissimilarities with the other platforms, probably because of their relatively small installed bases. As Figure 70 illustrates, unwanted software is generally consistent between platforms as well.

Figure 70. The unwanted software families most commonly encountered by Microsoft real-time antimalware solutions in 4Q15, and how they ranked in prevalence on different platforms

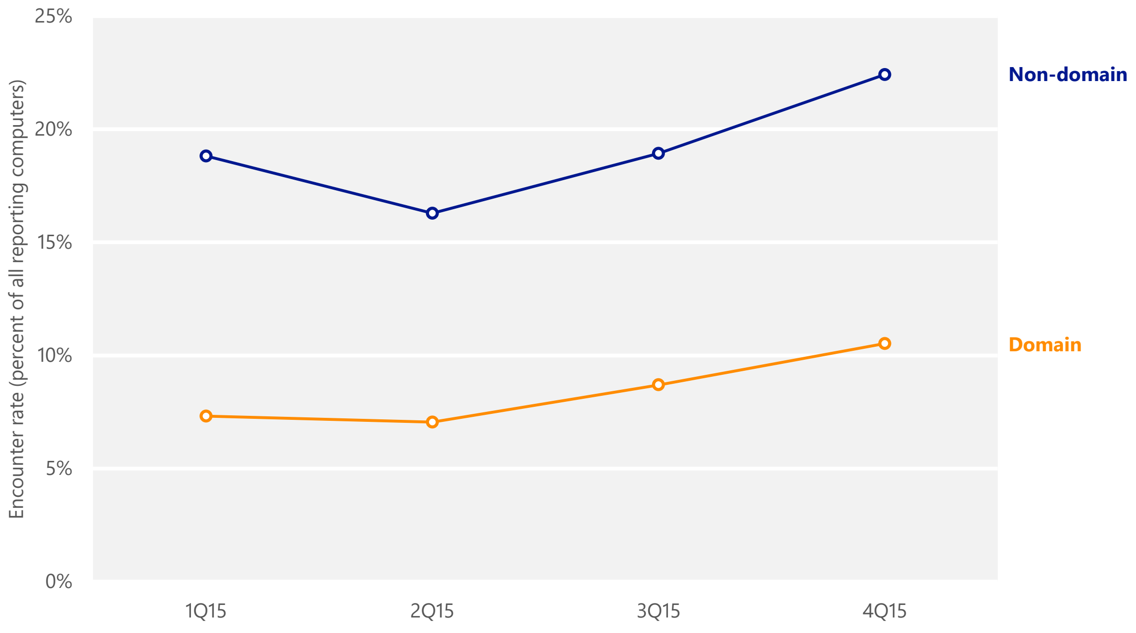
	Family	Most significant category	Rank					
			Win. Vista SP2	Win. 7 SP1	Win. 8 RTM	Win. 8.1 RTM	Win. 10 TH1	Win. 10 TH2
1	Win32/Diplugem	Browser Modifiers	1	1	1	2	2	3
2	Win32/SupTab	Browser Modifiers	2	2	2	1	1	1
3	Win32/OutBrowse	Software Bundlers	3	3	4	4	4	6
4	Win32/Pokki	Browser Modifiers	101	46	23	3	3	90
5	Win32/Bayads	Adware	5	4	3	5	19	30
11	Win32/Tillail	Software Bundlers	14	12	15	12	9	2

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.

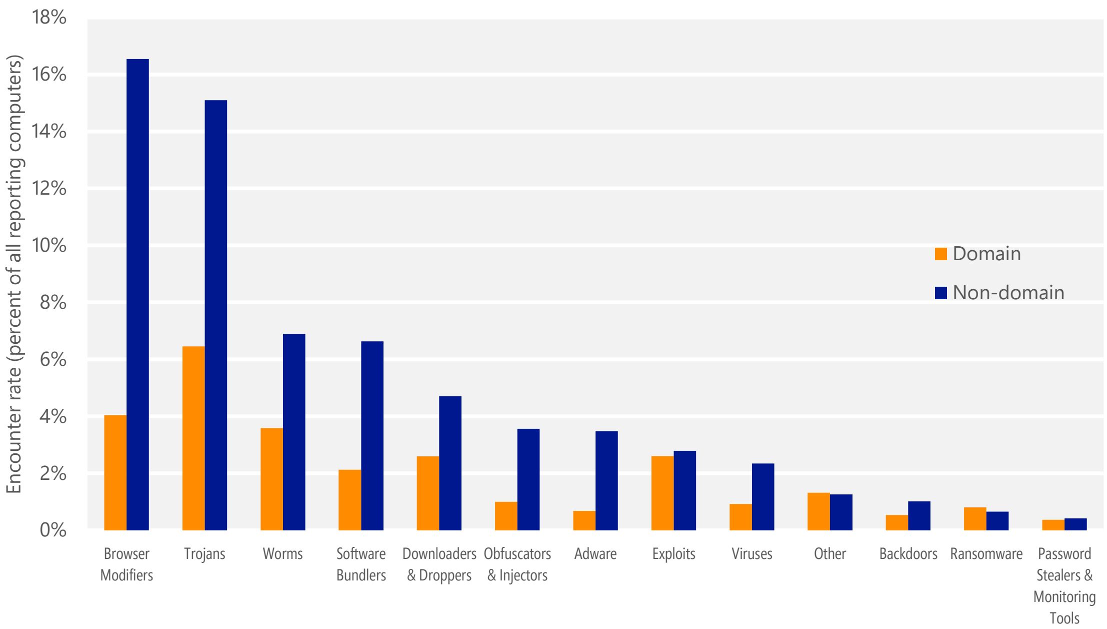
The infection telemetry data produced by Microsoft antimalware products and tools includes information about whether the infected computer belongs to an Active Directory Domain Services (AD DS) domain. Such domains are used almost exclusively in enterprise environments, and computers that do not belong 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 71. Malware encounter rates for domain-based and non-domain computers in 2015



Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

Figure 72. Malware and unwanted software encounter rates for domain-based and non-domain computers, 2H15, by category



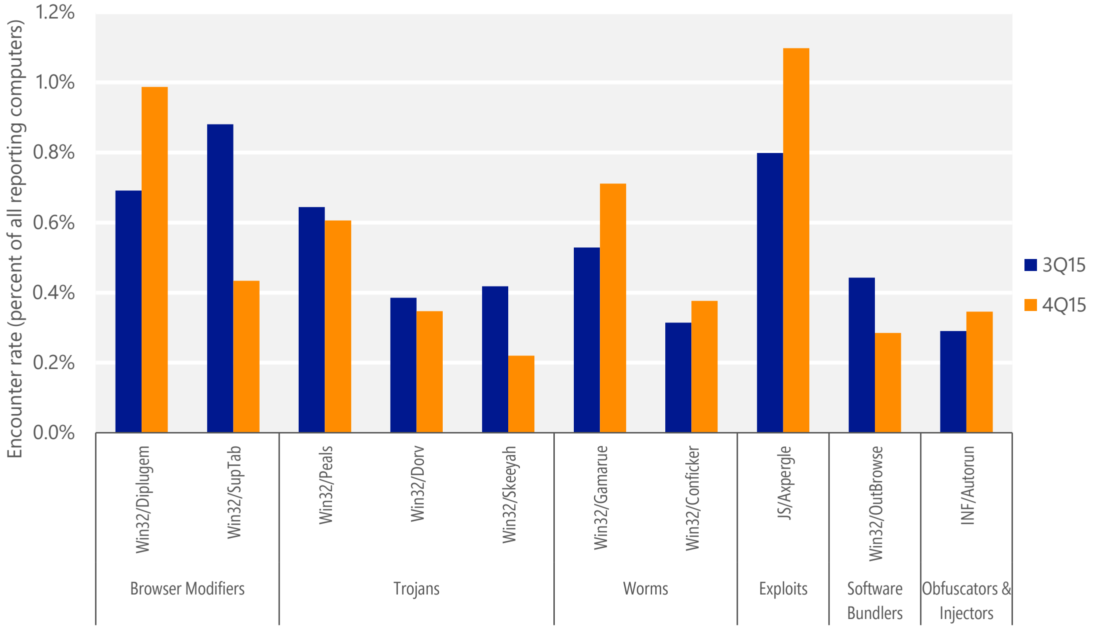
Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

- Enterprise environments typically implement defense-in-depth measures, such as enterprise firewalls, that prevent a certain amount of malware from reaching users' computers. Consequently, enterprise computers tend to encounter malware at a lower rate than consumer computers. As Figure 71 shows, the encounter rate for consumer computers was about 2.2 times as high as the rate for enterprise computers in 2H15.
- In addition to encountering less malware in general, computers in enterprise environments tend to encounter different kinds of threats than consumer computers, as shown in Figure 72. Non-domain computers encountered disproportionate amounts of unwanted software compared to domain-based computers, with Adware, Browser Modifiers, and Software Bundlers each appearing between three and six times as often on non-domain computers. Meanwhile, domain-based computers encountered exploits nearly as often as their non-domain counterparts, despite encountering less than half as much malware as non-domain computers overall.

Figure 73 and Figure 74 list the top 10 malware families detected on domain-joined and non-domain computers, respectively, in 2H15.

Figure 73. Quarterly trends for the top 10 malware and unwanted software families detected on domain-joined computers in 2H15, by percentage of computers encountering each family

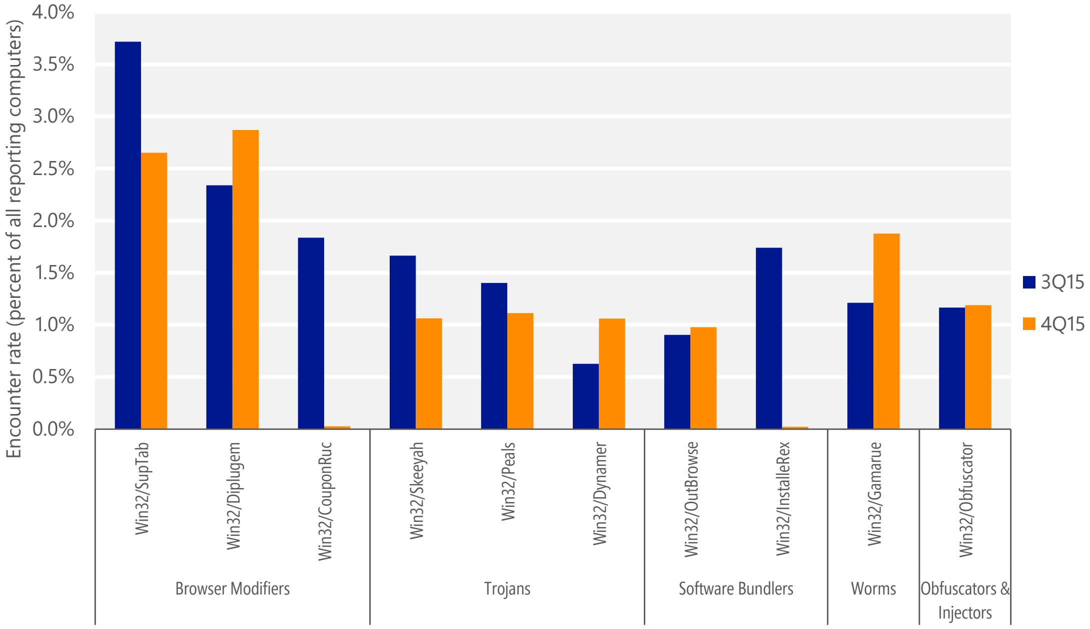
Family	Most significant category	3Q15	4Q15
JS/Axpergle	Exploits	0.80%	1.10%
Win32/Diplugem	Browser Modifiers	0.69%	0.99%
Win32/SupTab	Browser Modifiers	0.88%	0.43%
Win32/Peals	Trojans	0.64%	0.61%
Win32/Gamarue	Worms	0.53%	0.71%
Win32/Dorv	Trojans	0.39%	0.35%
Win32/OutBrowse	Software Bundlers	0.44%	0.28%
Win32/Conficker	Worms	0.31%	0.38%
Win32/Skeeyah	Trojans	0.42%	0.22%
INF/Autorun	Obfuscators & Injectors	0.29%	0.35%



Figures do not include Brantall, Rotbrow, and Filcout. See “Brantall, Rotbrow, and Filcout” on page 82 for more information.

Figure 74. Quarterly trends for the top 10 malware and unwanted software families detected on non-domain computers in 2H15, by percentage of computers encountering each family

Family	Most significant category	3Q15	4Q15
Win32/SupTab	Browser Modifiers	3.72%	2.65%
Win32/Diplugem	Browser Modifiers	2.34%	2.87%
Win32/Gamarue	Worms	1.21%	1.88%
Win32/Skeeyah	Trojans	1.66%	1.06%
Win32/Peals	Trojans	1.40%	1.11%
Win32/Obfuscator	Obfuscators & Injectors	1.17%	1.19%
Win32/OutBrowse	Software Bundlers	0.91%	0.98%
Win32/CouponRuc	Browser Modifiers	1.84%	0.03%
Win32/InstalleRex	Software Bundlers	1.74%	0.02%
Win32/Dynamer	Trojans	0.63%	1.06%



Figures do not include Brantall, Rotbrow, and Filcout. See “Brantall, Rotbrow, and Filcout” on page 82 for more information.

- Six families—Win32/SupTab, Win32/Diplugem, Win32/Gamarue, Win32/Skeeyah, Win32/Peals, and Win32/OutBrowse—were common to both lists. All were more frequently encountered on non-domain computers than on domain-joined computers. See “Threat families” on page 97 for more information about these families.

- The four families that were unique to the top 10 list for domain-joined computers but not for non-domain computers are the exploit kit JS/Axpergle, the trojan family Win32/Dorv, the worm family Win32/Conficker, and the generic detection INF/Autorun.
    - Axpergle is the Microsoft detection name for the Angler exploit kit. See "Exploit kits" on page 66 for more information about Axpergle and other exploit kits.
    - Conficker is a worm that was disrupted several years ago, but continues to be encountered in domain environments because of its use of a built-in list of common and weak passwords to spread between computers.
    - Autorun is a detection for threats that spread by copying themselves to the mapped drives of an infected computer, which may include network and removable drives. Changes to the way the AutoRun feature works make it more difficult for this technique to succeed in recent versions of Windows, but attackers continue to attempt to use it against older installations.
- Outside the top 10 are a number of threats that are encountered significantly more frequently on domain-joined computers than on non-domain computers, often because of factors that make it easier for them to spread in enterprise environments.
    - W97M/Donoff was the 18th most commonly encountered family on domain-joined computers in 2H15, on which it was encountered about three times as frequently as on non-domain computers. Donoff is a malicious macro script for Microsoft Office files, which are commonly used in enterprise environments.
    - W97M/Adnel, another malicious Microsoft Office script, was the 21st most commonly encountered family on domain-joined computers in 2H15, and was encountered there about four times as frequently as on non-domain computers.

Donoff, a malicious macro script for Microsoft Office files, was encountered three times as frequently on domain-joined computers as on non-domain computers.

See "Malware at Microsoft: Dealing with threats in the Microsoft environment" on page 141 for information about the threat landscape for computers at

Microsoft and to learn about the actions Microsoft IT takes to protect users, data, and resources.

Threats from targeted attackers

Although using a real-time security software product from a reputable vendor and keeping the detection signatures up-to-date remains one of the best ways individuals and organizations can protect themselves against known threats, conventional antimalware software is often less effective against advanced attacks, such as those conducted by targeted attack groups. These groups, which focus on targeting computers at specific institutions, often use specially crafted threats that they test against popular antimalware solutions ahead of time to ensure that they will not be detected. By the time detection signatures are available to stop such a threat, it may have already compromised the organization. To help organizations combat such attacks, Office 365 Advanced Threat Protection, available with select Office 365 plans, provides an additional layer of defense against threats and malicious links that have never been encountered before.

When an incoming message includes a potentially dangerous attached file, Exchange Online launches it in a detonation chamber—a virtual sandboxed environment in which potential threats can run without posing harm to any other resources—and monitors it for malicious behavior such as suspicious registry changes, attempts to access memory dumps, changes to executables, and other actions that malware characteristically takes. This monitoring makes it possible to detect and block threats that have never been seen before and for which no detection signatures are available. Office 365 Advanced Threat Protection includes anti-sandbox detection features to combat advanced threats that avoid taking malicious actions when they determine they are being run in a virtual machine.

Figure 75. How Advanced Threat Protection works with Exchange Online

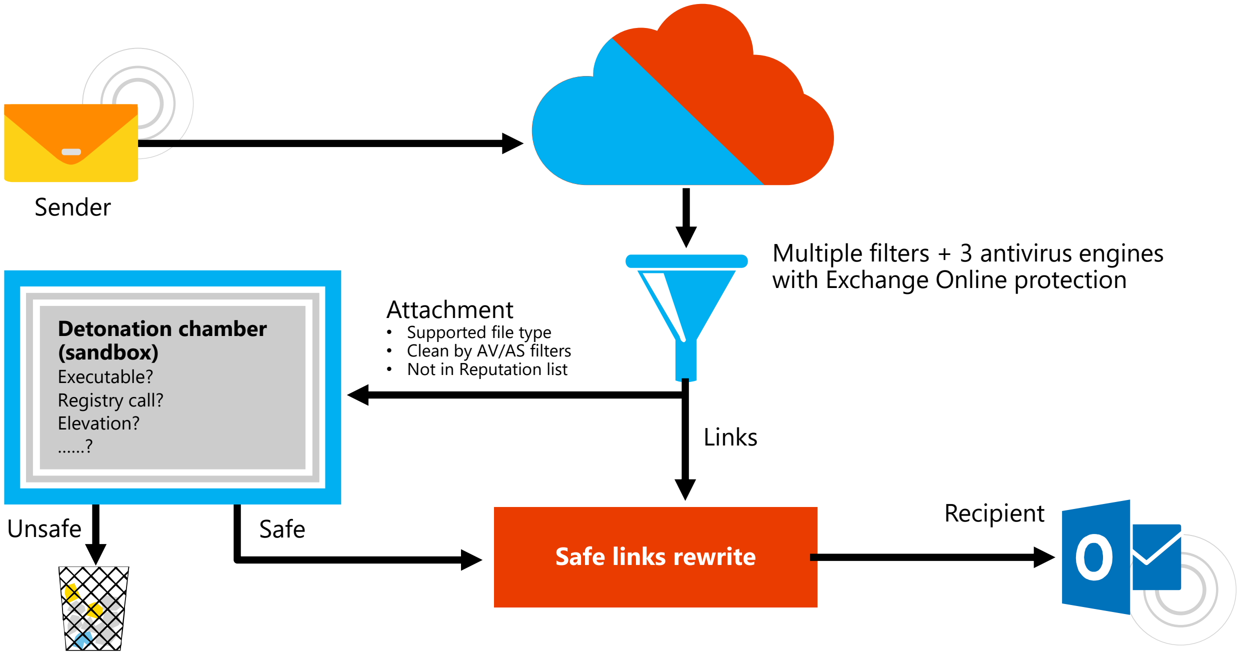
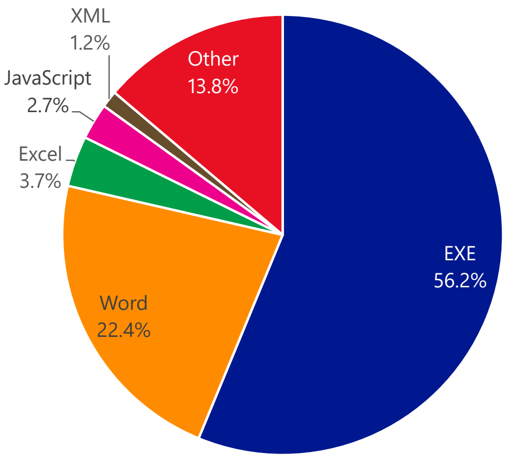


Figure 76 illustrates the file types of the malicious attachments blocked by Office 365 Advanced Threat Protection in 2H15.

Figure 76. Types of malicious files blocked by Office 365 Advanced Threat Protection in 2H15

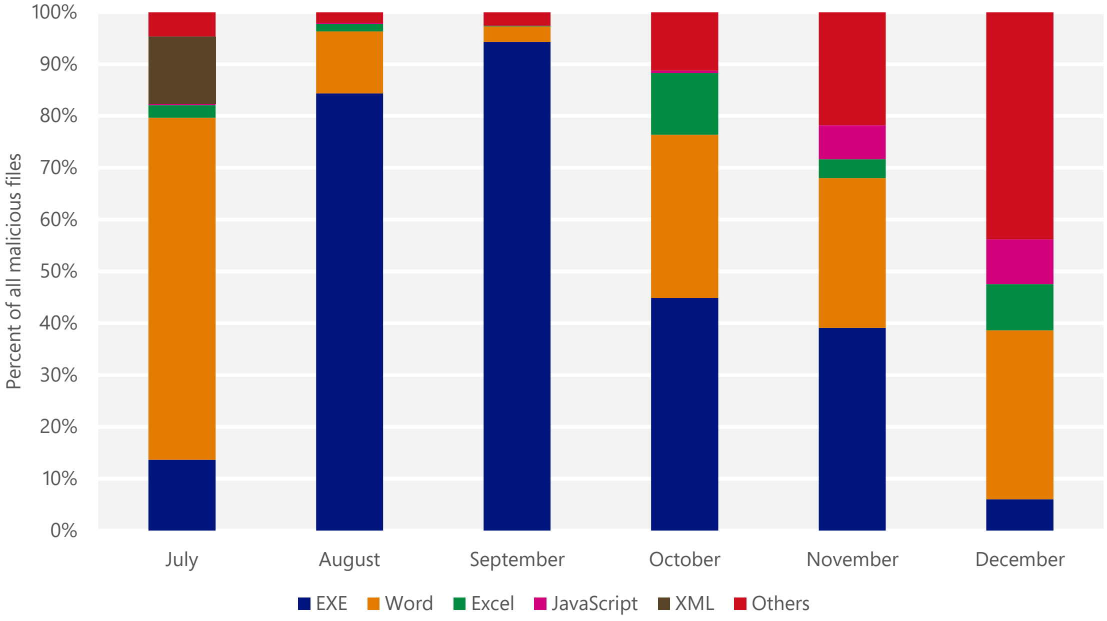


- Executable files accounted for the largest share of malicious files, at 56.2 percent of the total. This type includes the familiar .exe extension used by most executable programs in Windows, along with a number of other extensions, such as .scr, .com, .pif, and .bat. Executable files can provide an attacker with an easy way to compromise a computer without relying on exploiting a vulnerability, but most enterprise email servers and programs are configured to block them by default.

- Microsoft Word files accounted for 22.4 percent of malicious files. Of these, the most common file extensions were .doc, used for the binary file format used in Word 97-2003, and .docm, used for Word documents that contain macros.
- Microsoft Excel, JavaScript, and XML files each accounted for a small percentage of the total.
- Other file types accounted for 13.8 percent of the total. Some of the more common file extensions here were .eml, used by Microsoft Outlook to save email messages to disk; .rar, a popular compression and archive format; .vbs, used by VBScript, and .jar, a package format used for Java files.

As Figure 77 demonstrates, the file types used for advanced threats changes significantly from month to month, as targeted attack groups shift between different victims and tactics.

Figure 77. Malicious files blocked by Office 365 Advanced Threat Protection in 2H15, by month



- Detections of malicious executable files peaked in August and September at more than 80 percent of the total, as one group mounted a sustained attack campaign that dwarfed all other activity. By December, executable files had decreased to just 6 percent of the total.

- Malicious Word files accounted for two-thirds of the total in July, then faded to relative insignificance during the executable file campaign, then finally recovered to about a third of all malicious files by the end of the year.
- All of the malicious XML files detected by Office 365 Advanced Threat Protection were sent in July.

Potentially unwanted applications in the enterprise

Microsoft has published the criteria used to classify programs as unwanted software at www.microsoft.com/security/portal/mmpc/shared/objectivecriteria.aspx. Characteristics of unwanted software can include depriving users of adequate choice or control over what the software does to the computer, preventing users from removing the software, or displaying advertisements without clearly identifying their source. Microsoft security products classify unwanted software as threats, and block or remove them when they are encountered.

Some programs don't meet the criteria to be considered unwanted software but still exhibit behaviors that may be considered undesirable, particularly in enterprise environments. Microsoft classifies these programs as *potentially unwanted applications* (PUA). For example, a program that displays additional advertisements in the browser might not be classified as unwanted software if it clearly identifies itself as the source of the ads, but may be considered potentially unwanted. Users often end up installing these programs because they were installing an application that they wanted, and the installer offered to install additional software—usually with the offer acceptance checked by default and often without the user realizing they are agreeing to install the additional software. These programs can also cause problems for network administrators—they can affect computer performance, increase the workload for the IT help desk, put computers and data at risk of being compromised through exploits, and make it more difficult to identify malware infections among the noise. To provide organizations with additional options for dealing with programs classified as PUA, Microsoft is now offering enterprise users of System Center Endpoint Protection (SCEP) the ability to block them from being installed on their networks.

PUA statistics

The statistic presented here come from a pilot of the PUA functionality conducted on several thousand enterprise computers during 2H15.

Figure 78. PUA, malware, and unwanted software blocked during 2H15 pilot project, by month

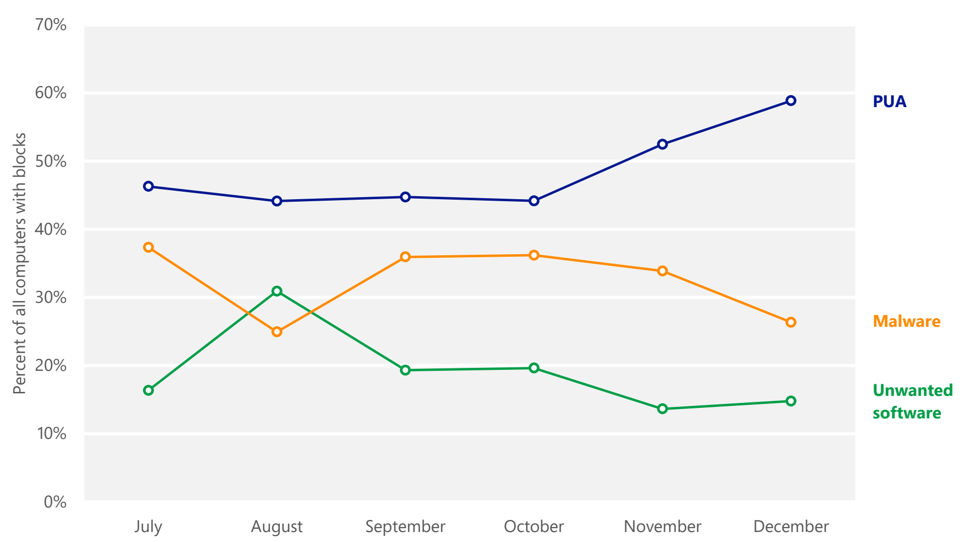
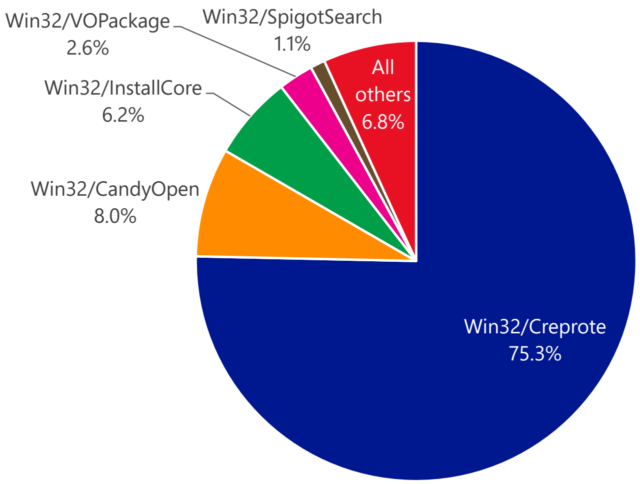


Figure 78 demonstrates the impact that PUA can potentially have in an enterprise environment. PUA was responsible for more detections each month during the pilot project than either malware or unwanted software. In fact, approximately half of the detections during the pilot involved PUA, with malware and unwanted software making up the rest.

Figure 79. PUA families blocked during 2H15 pilot project



The generic detection PUA:Win32/Creprote was responsible for about three-fourths of all PUA detections during the pilot, as shown in Figure 79. Creprote is a generic detection for software signed with certificates that fail the Microsoft reputation-based system for distinguishing PUA from other programs.

PUA:Win32/CandyOpen and PUA:Win32/InstallCore are detections for installer programs that were built with software bundler utilities (called OpenCandy and InstallCore, respectively) that offer monetization opportunities to software developers, such as pay-per-install services for programs that offer to download other programs alongside the requested application.

PUA:Win32/VOPackage is a software bundler that can tamper with system settings such as the Windows hosts file to prevent computer users from accessing websites that belong to competitors. It can also force users to install unwanted applications by disabling the Cancel button in the installer.

PUA:Win32/SpigotSearch is a toolbar that can automatically change the browser's default search provider, and advertises that it will periodically restore this changed search provider if it has been changed to something else.

Looking at the names of files installed with PUA software bundlers can help administrators understand which outside product names are being used as installation vectors. Figure 80 lists the filenames most commonly detected as part of OpenCandy and InstallCore installation packages.

Figure 80. Top filenames used by OpenCandy and installCore software bundlers during 2H15 pilot project

	PUA:Win32/CandyOpen (OpenCandy)		PUA: Win32/InstallCore	
	Filename	**% of CandyOpen**	**Filename**	**% of InstallCore**
1	OCSetupHlp.dll	38.0%	FileZilla_3.exe	23.7%
2	uTorrent.exe	21.8%	setup.exe	8.6%
3	FreemakeVideoConverterSetup.exe	5.8%	adobe_flash_player.exe	8.2%
4	epm.exe	3.3%	GoogleChromeSetup.exe	5.8%
5	CheatEngine64.exe	3.0%	DownloadManagerSetup.exe	4.5%
6	avc-free.exe	3.0%	ZipOpenerSetup.exe	4.2%
7	KeyFinderInstaller.exe	2.7%	FlvPlayerSetup.exe	4.2%
8	cdbxp_setup_4.5.6.5931.exe	2.4%	adobe_flash_setup.exe	3.8%
9	cdbxp_setup_4.5.6.5844.exe	2.4%	FirefoxSetup.exe	3.3%
10	youtube_downloader_hd_setup.exe	2.4%	Uninstall.exe	3.2%

After OCSetupHlp.dll, a code library file used by the OpenCandy installation platform, the most commonly detected filename associated with OpenCandy is utorrent.exe, an installer for the popular µTorrent file sharing client. The µTorrent website uses an OpenCandy installer to distribute the program along with other offers to generate revenue. OpenCandy installers are also used frequently to distribute audio and video file conversion programs such as Freemake Video Converter (FreemakeVideoConverterSetup.exe), Any Video Converter (avc-free.exe), and CDBurnerXP (cdbxp_setup_4.5.6.5931.exe, cdbxp_setup_4.5.6.5844.exe).

FileZilla, a popular FTP client application, was responsible for almost one-fourth of the InstallCore detections. As with µTorrent and OpenCandy, the official FileZilla website distributes the application using an InstallCore installer. In addition, a number of dubious software distributors use InstallCore to create and distribute monetized installers for popular programs that can be downloaded for free elsewhere without the bundled programs, such as Adobe Flash Player, Google Chrome, and Mozilla Firefox. To convince users to install them, distributors often purchase ad placements on search terms that are related to these popular programs. Some such distributors use deceptive advertisements that claim that the user's browser is out of date, or that the user must install a video player or other component to view website content, while offering links to the distributors' bundled versions.

As Figure 81 demonstrates, the majority of PUA programs are digitally signed by their creators.

Figure 81. The digital signers with the most PUA detections during 2H15 pilot project

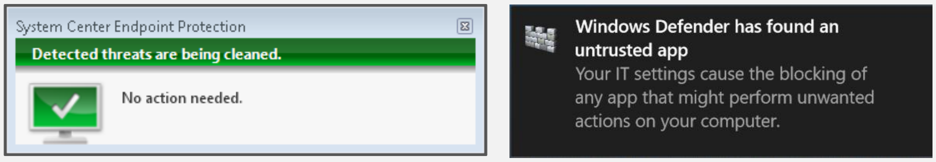
	Signer	Percent of all detections
1	*(Unsigned)*	11.7%
2	OpenCandy*	6.5%
3	Taiwan Shui Mu Chih Ching Technology Limited	4.0%
4	Spigot, Inc.	3.3%
5	ProInstall Applications SRL	2.0%
6	Ellora Assets Corporation	1.8%
7	Mindspark Interactive Network	1.6%
8	ClientConnect LTD	1.5%
9	CHIP Digital GmbH	1.4%
10	Canneverbe Limited	1.2%

Only 11.7 percent of PUA detections during the pilot project were unsigned. Of the ones that were signed, the largest share (6.5 percent of all PUA detections) were signed by SweetLabs (formerly OpenCandy, Inc.), the company that publishes the OpenCandy software bundler. Most of these files were components of the OpenCandy installation platform itself, predominantly OCSetupHlp.dll, discussed earlier. Taiwan Shui Mu Chih Ching Technology Limited, which distributes a number of utility programs for Windows, was next, followed by Spigot, Inc., publisher of PUA:Win32/SpigotSearch.

Blocking PUA with System Center Endpoint Protection

System administrators can configure the PUA protection feature through System Center Configuration Manager (SCCM) or Microsoft Intune. For more information about creating a configuration item to enable PUA protection in System Center Configuration Manager, see How to Configure Endpoint Protection in Configuration Manager on Microsoft TechNet (technet.microsoft.com). For more information about configuring policy settings in Microsoft Intune, see Windows 10 configuration policy settings in Microsoft Intune, also on Microsoft TechNet.

Figure 82. PUA is blocked by System Center Endpoint Protection (left) and Windows Defender in Windows 10 (right)



System Center Endpoint Protection

Detected threats are being cleaned.

No action needed.

Windows Defender has found an untrusted app

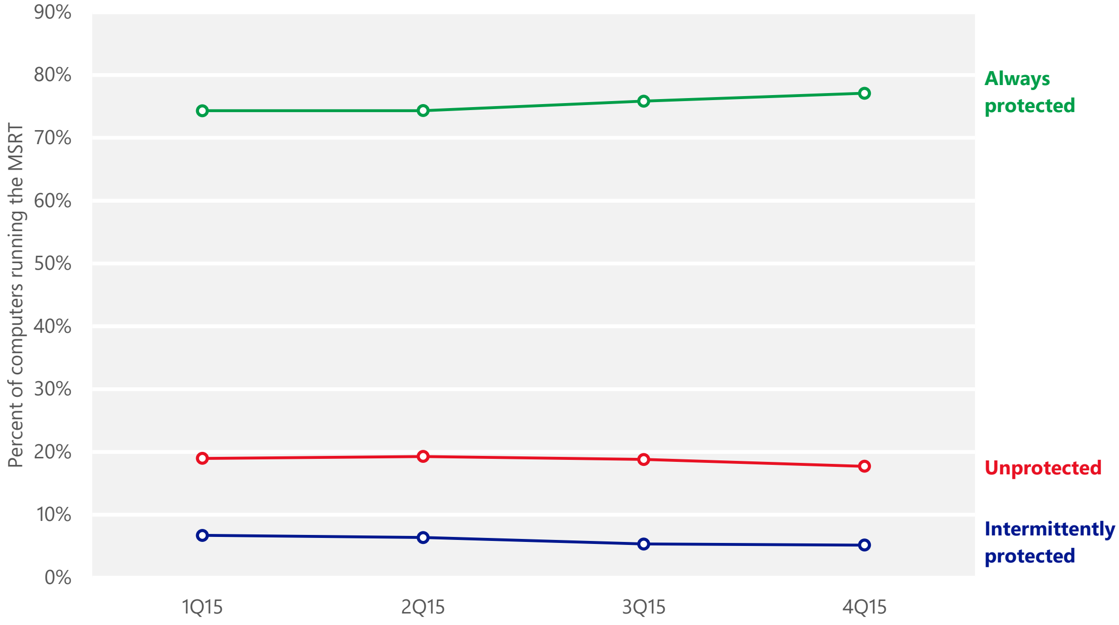
Your IT settings cause the blocking of any app that might perform unwanted actions on your computer.

When enabled, PUA is blocked and automatically quarantined; users who request more information online are informed that the program was blocked from running on the network because it has a poor reputation. PUA that is already installed on the computer will not be removed.

Security software use

Recent releases of the MSRT collect and report details about the state of real-time antimalware software on a computer, if the computer's administrator has chosen to opt in to provide data to Microsoft. This telemetry data makes it possible to analyze security software usage patterns around the world and correlate them with infection rates. Figure 83 shows the percentage of computers worldwide that the MSRT found to be protected and unprotected by real-time security software each quarter in 1H15 and 2H15.

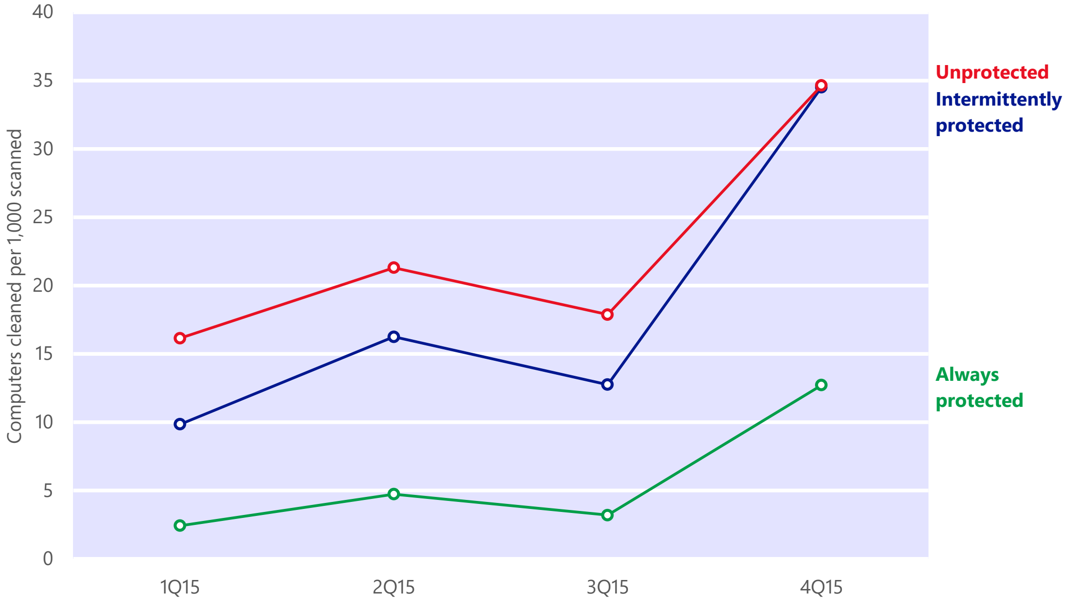
Figure 83. Percentage of computers worldwide protected by real-time security software in 2015



- A typical computer runs the MSRT three times each quarter, once for each monthly version of the tool that Microsoft releases. In Figure 83, "Protected" represents computers that had real-time security software active and up-to-date every time the MSRT ran during a quarter; "Intermittently protected" represents computers that had security software active during one or more MSRT executions, but not all of them. "Unprotected" represents computers that did not have security software active during any MSRT executions that quarter.
- Overall, about three-fourths of computers worldwide were found to be always protected at every monthly MSRT execution in each of the past four quarters, varying between 74.3 percent and 77.1 percent.
- Computers that never reported running security software accounted for between 17.7 and 19.3 percent of computers worldwide each quarter. Intermittently protected computers—those that were found to be running real-time security software during at least one MSRT execution in a quarter, but not all of them—accounted for between 5.2 and 6.7 percent of computers each quarter.

Computers that do not run real-time security software are at significantly greater risk of malware infection than computers that do. Figure 84 compares infection rates with protection levels worldwide for each of the last four quarters.

Figure 84. Infection rates for protected and unprotected computers in 2015



Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

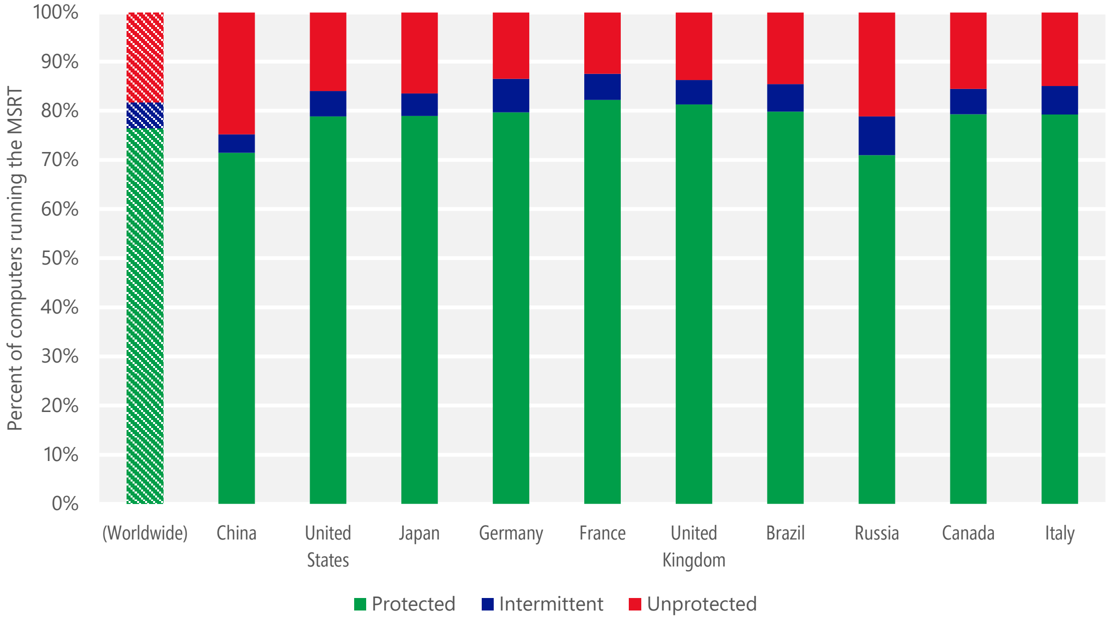
- Infection rates increased significantly for all protection levels in 4Q15 due to Win32/Diplugem. See "Diplugem and infection rates" on page 81 for more information.
- The MSRT reported that computers that were never found to be running real-time security software during 2H15 were between 2.7 and 5.6 times as likely to be infected with malware as computers that were always found to be protected.
- Computers that were intermittently protected were between 2.7 and 4.0 times more likely to be infected with malware in 2H15 than computers that were always protected.
- Users who don't run real-time security software aren't always unprotected by choice: a number of prevalent malware families are capable of disabling some security products, potentially without the user even knowing. Other users might disable or uninstall security software intentionally because of perceived performance issues, a belief that protection is not necessary, or a desire to run programs that would be quarantined or removed by security software. In other cases, users lose up-to-date real-time protection when they don't renew paid subscriptions for their antimalware software, which might come pre-installed with their computers as limited-time trial software. (See "The challenge of expired security software" on pages 21–28 of *Microsoft Security Intelligence Report, Volume 17 (January–June 2014)*, available from the Microsoft Download Center, for more information about the causes and consequences of expired security software.) Whatever the reason, users who don't have functioning real-time antimalware protection face significantly greater risk from malware infection than users who do, as Figure 84 illustrates.

> Computers that were unprotected were between 2.7 and 5.6 times as likely to be infected with malware as computers that were protected.

Security software use worldwide

Just as infection and encounter rates differ from one country or region to another, so do security software usage rates, as shown in Figure 85.

Figure 85. Average security software protection state for the locations with the most computers executing the MSRT in 2H15



- Computers that reported being fully protected in these locations ranged between 71.0 percent and 82.2 percent, with all locations except China and Russia exceeding the worldwide rate of 76.5 percent of computers reporting as fully protected.
- Computers that reported being fully unprotected in these locations ranged between 12.5 percent and 24.8 percent, with Russia and China reporting larger percentages of fully unprotected computers than the world overall.
- Computers that were protected in some months but not in others accounted for between 3.7 percent and 7.9 percent in these locations.

The rate of security software usage in a country or region often correlates with its infection rate. Figure 86 and Figure 87 show the percentage of computers in different countries and regions that reported being fully protected and fully unprotected, respectively, in 4Q15.

Figure 86. Percent of computers reporting as Protected during every MSRT execution in 4Q15, by country/region

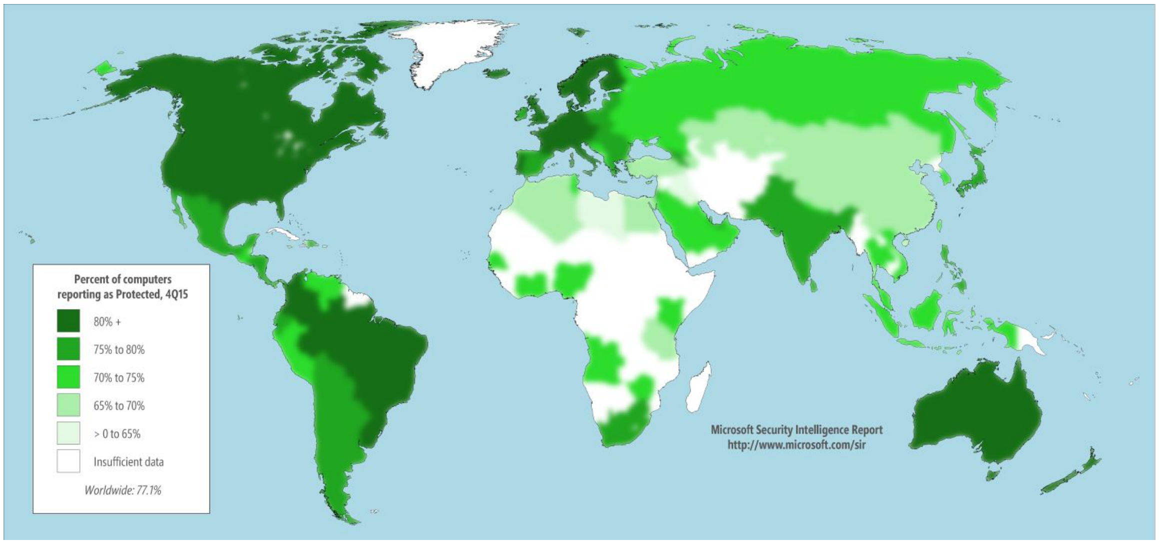
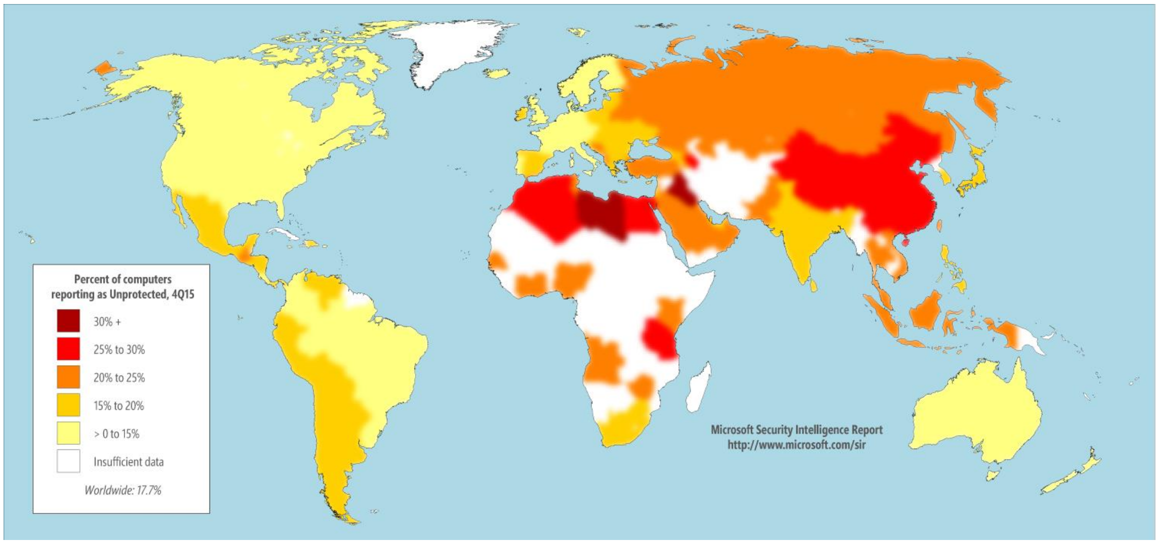

Percent of computers reporting as Protected, 4Q15
80% +
75% to 80%
70% to 75%
65% to 70%
> 0 to 65%
Insufficient data
Worldwide: 77.1%
Microsoft Security Intelligence Report
http://www.microsoft.com/sir

Figure 87. Percent of computers reporting as Unprotected during every MSRT execution in 4Q15, by country/region


Percent of computers reporting as Unprotected, 4Q15
30% +
25% to 30%
20% to 25%
15% to 20%
> 0 to 15%
Insufficient data
Worldwide: 17.7%
Microsoft Security Intelligence Report
http://www.microsoft.com/sir

- The locations with the most computers reporting as fully protected by real-time security software include Finland, with 87.9 percent of computers reporting as fully protected in 4Q15; Norway, at 84.5 percent; and Denmark, at 84.4 percent. Locations with the fewest computers reporting as fully protected include Libya, at 55.0 percent; Iraq, at 64.4 percent; and Algeria, at 66.3 percent.
- In general, the percentage of computers reporting as fully protected was significantly higher in most countries and regions in 4Q15 than in 2Q15,

which can be at least partially attributed to adoption of Windows 10. See "Security software use by platform" on page 125 for more information.

- The ranking of countries and regions by unprotected rate is largely an inverse of their ranking by protected rate. The locations with the fewest computers reporting as fully unprotected include Finland, at 8.8 percent; Denmark, at 11.5 percent; and Norway, at 11.6 percent. Locations with the most computers reporting as fully unprotected include Libya, at 37.5 percent; Iraq, at 30.8 percent; and Morocco, at 27.3 percent.

Countries and regions with high percentages of computers reporting as fully unprotected also tend to have high infection rates, as Figure 88 shows.

Figure 88. Infection rates for the locations with the highest percentage of computers reporting as fully unprotected in 2H15

Country/region	2H15 average unprotected %	CCM 3Q15	CCM 4Q15	Unprotected CCM 3Q15	Unprotected CCM 4Q15
Libya	39.00%	56.8	85.3	125.4	172.1
Iraq	33.70%	67.9	80.0	167.0	206.5
Morocco	28.47%	46.1	69.9	143.3	181.2
Azerbaijan	28.13%	23.0	43.7	60.7	93.7
Algeria	27.64%	43.3	62.6	123.3	147.6
Mongolia	26.96%	62.0	93.3	165.8	269.6
Palestinian Authority	26.69%	53.5	80.0	151.3	213.7
Egypt	26.64%	47.2	60.2	130.9	150.5
Indonesia	26.47%	31.4	72.1	88.2	177.9
Jordan	26.00%	33.6	64.3	94.9	160.8
Worldwide	*18.27%*	*6.1*	*16.9*	*17.6*	*34.2*

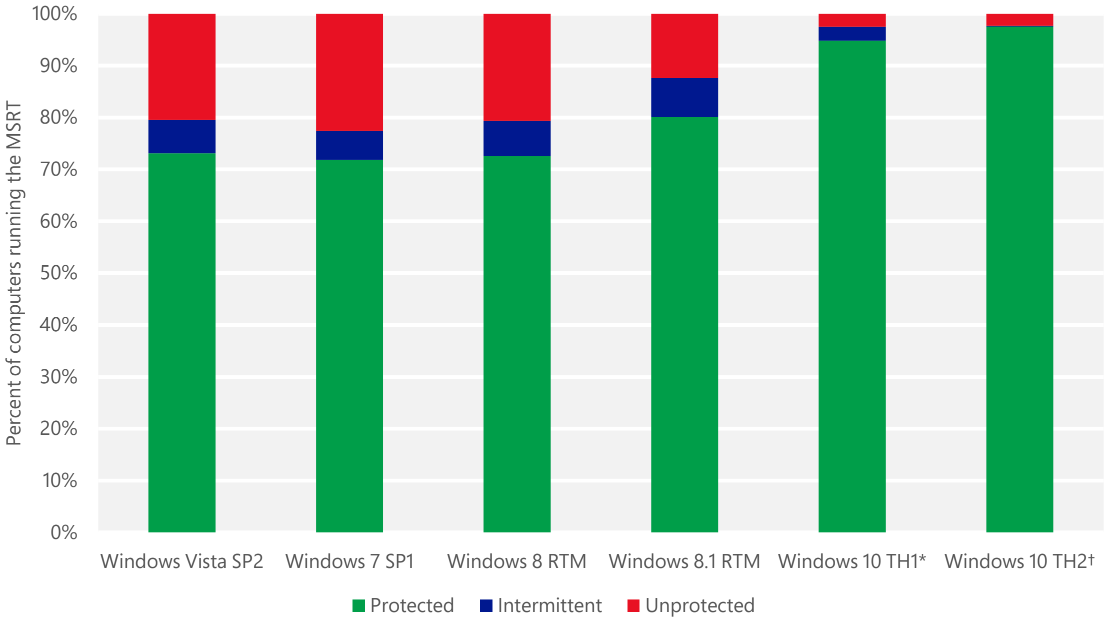
Figures do not include Brantall, Rotbrow, and Filcout. See "Brantall, Rotbrow, and Filcout" on page 82 for more information.

- The locations in the table all had overall infection rates ranging between 2.6 and 11.1 times as high as the worldwide average each quarter.
- The infection rates for fully unprotected computers in these locations ranged between 2.7 and 9.4 times as high as the infection rates for fully unprotected computers worldwide, and between 5.5 and 27.2 times as high as the infection rates for all computers worldwide. In Mongolia, the location with the highest infection rates in Figure 70, the MSRT detected and removed malware on 27.0 percent of the fully unprotected computers that executed it at least once in 2Q15 (a CCM of 269.6).

Security software use by platform

Protection rates can also vary by operating system, as shown in Figure 89.

Figure 89. Average quarterly security software protection state for supported client versions of Windows in 2H15



* Released July 29, 2015 † Released November 12, 2015

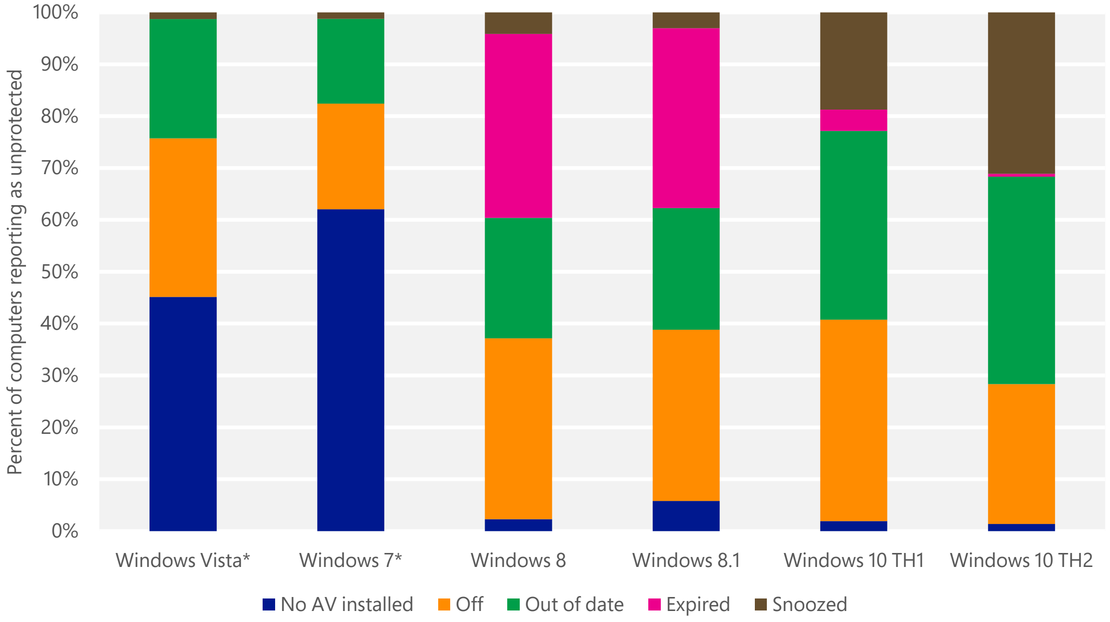
- In general, computers running newer versions of Windows tended to report being fully protected more often than computers running older versions, and to report being fully unprotected less often.

> Windows Defender is automatically activated upon installation of Windows 10 if no other real-time security product is installed.

- Both the initial release of Windows 10 (TH1) and the Windows 10 November Update (TH2) reported being fully protected on more than 94 percent of computers throughout the period, and fully unprotected on less than 3 percent of computers. (Note that most computers running the Windows 10 November Update only executed the MSRT one time before the end of the year, and would have been counted as fully protected if security software was running and up-to-date during that single MSRT execution.) The high rate of protection with Windows 10 is primarily due to a change in the way Windows Defender operates. To provide Windows 10 users with protection from malware out of the box, Windows Defender is automatically activated upon installation of Windows 10 if no other real-time security product is installed, as opposed to a few days after installation in Windows 8 and Windows 8.1.

- The reasons computers go unprotected can vary significantly by platform, as Figure 90 illustrates.

Figure 90. Status reported by unprotected computers running supported client versions of Windows in 2H15



* Windows Vista and Windows 7 do not report expired subscriptions.

- On Windows Vista and Windows 7, unprotected computers predominantly report having no antimalware software installed at all. On subsequent Windows versions, Windows Defender is enabled by default if no other antimalware software is present, so the number of computers reporting no antimalware software is very low.
- On Windows 8 and Windows 8.1, expired versions of commercial antimalware products that are no longer receiving signature updates account for the largest percentage of unprotected computers.
- On the initial release of Windows 10 and the Windows 10 November Update, out-of-date signatures were the most common reason computers lacked protection. Expired subscriptions accounted for a very low percentage of unprotected computers running Windows 10, probably because many trial subscriptions of commercial antimalware products that were pre-installed on new computers sold with Windows 10 had yet to expire during the period.

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 Help prevent malware infection on your PC at the Microsoft Malware Protection Center website at www.microsoft.com/mmpc.

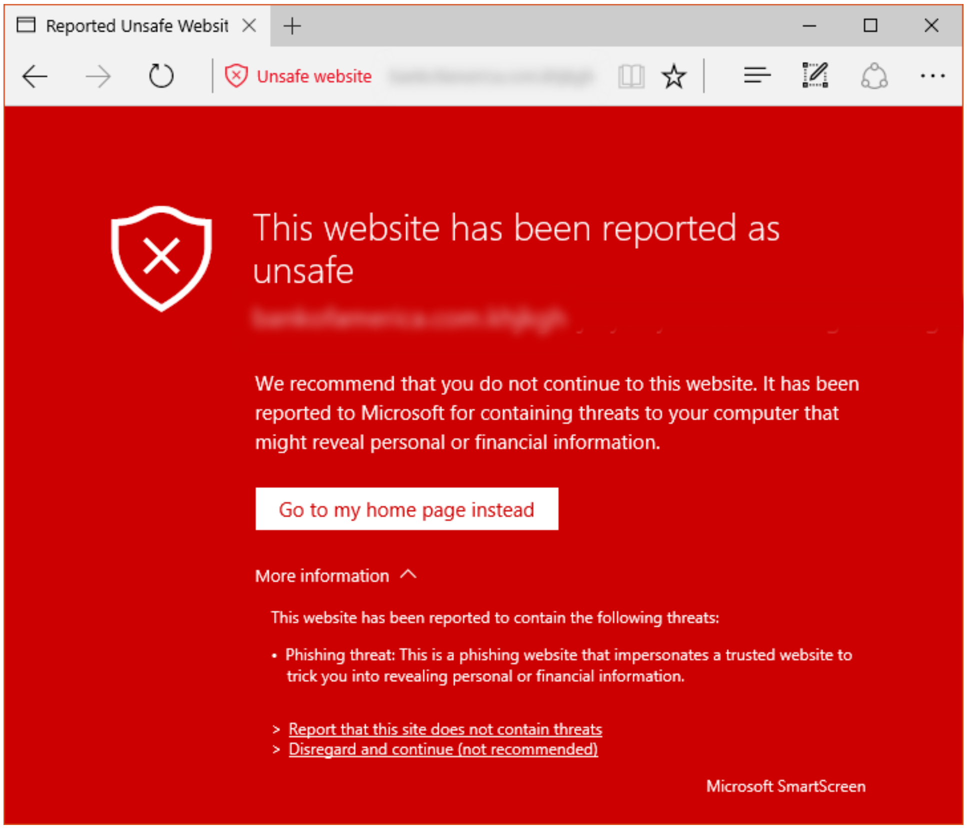
For help understanding the threats that pose the greatest risk to your environment and how to defend against them, see "Fixing the #1 Problem in Computer Security: A Data-Driven Defense," available from Microsoft TechNet.

Malicious websites

Attackers often use websites to conduct phishing attacks or distribute malware. Malicious websites typically appear to be completely legitimate, and provide no outward indicators of their malicious nature even to experienced computer users. In many cases, these sites are legitimate websites that have been compromised by malware, SQL injection, or other techniques in efforts by attackers to take advantage of the trust users have invested in such sites. 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 sources, including telemetry data produced by SmartScreen Filter in Internet Explorer versions 8 through 11 and Microsoft Edge, 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 156 for more information about the products and services that provided data for this report.)

Figure 91. SmartScreen Filter in Microsoft Edge and Internet Explorer blocks reported phishing and malware distribution sites to protect users



This website has been reported as unsafe

[redacted]

We recommend that you do not continue to this website. It has been reported to Microsoft for containing threats to your computer that might reveal personal or financial information.

This website has been reported to contain the following threats:

- Phishing threat: This is a phishing website that impersonates a trusted website to trick you into revealing personal or financial information.

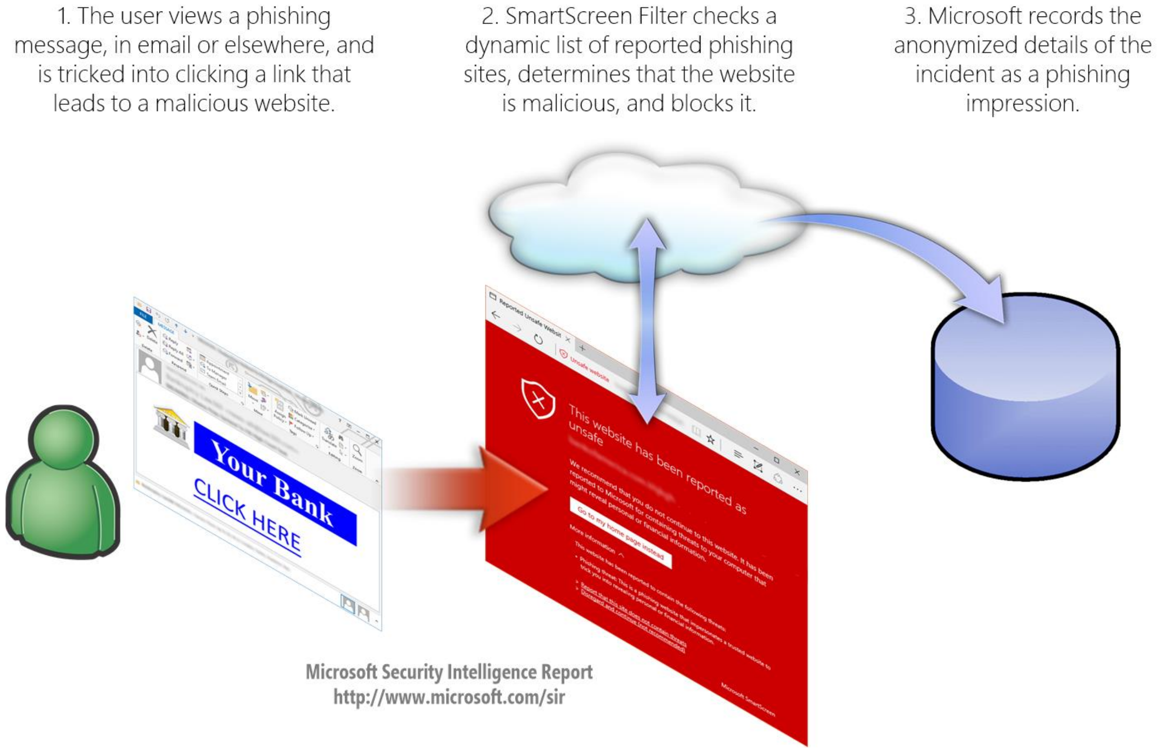
> Report that this site does not contain threats
> Disregard and continue (not recommended)

Phishing sites

Microsoft gathers information about phishing sites and impressions from *phishing impressions* that are generated by users who choose to enable SmartScreen Filter.[19] A phishing impression is a single instance of a user attempting to visit a known phishing site with SmartScreen Filter enabled and being warned, as illustrated in Figure 92.

[19] See "Appendix B: Data sources" on page 157 for information about the products and services used to provide data for this report.

Figure 92. 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.

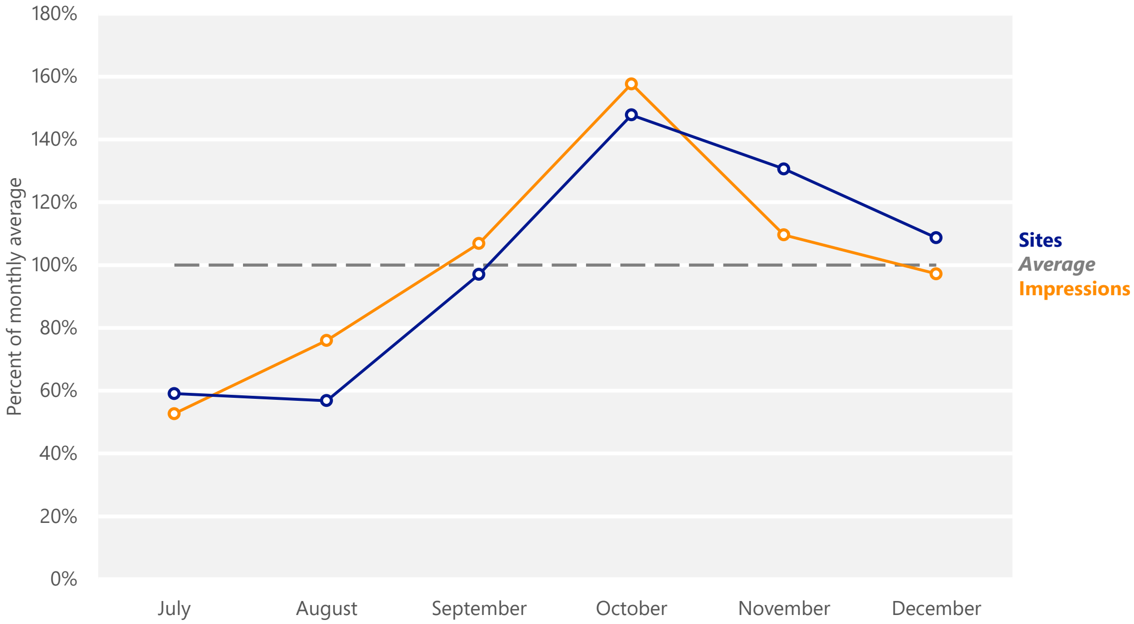
2. SmartScreen Filter checks a dynamic list of reported phishing sites, determines that the website is malicious, and blocks it.

3. Microsoft records the anonymized details of the incident as a phishing impression.

Your Bank
CLICK HERE
This website has been reported as unsafe
Go to my home page instead
Microsoft Security Intelligence Report
http://www.microsoft.com/sir

Figure 93 illustrates the volume of phishing impressions tracked by SmartScreen Filter each month in 2H15, compared to the volume of distinct phishing URLs visited.

Figure 93. Phishing sites and impressions reported by SmartScreen Filter each month in 2H15, relative to the monthly average for each

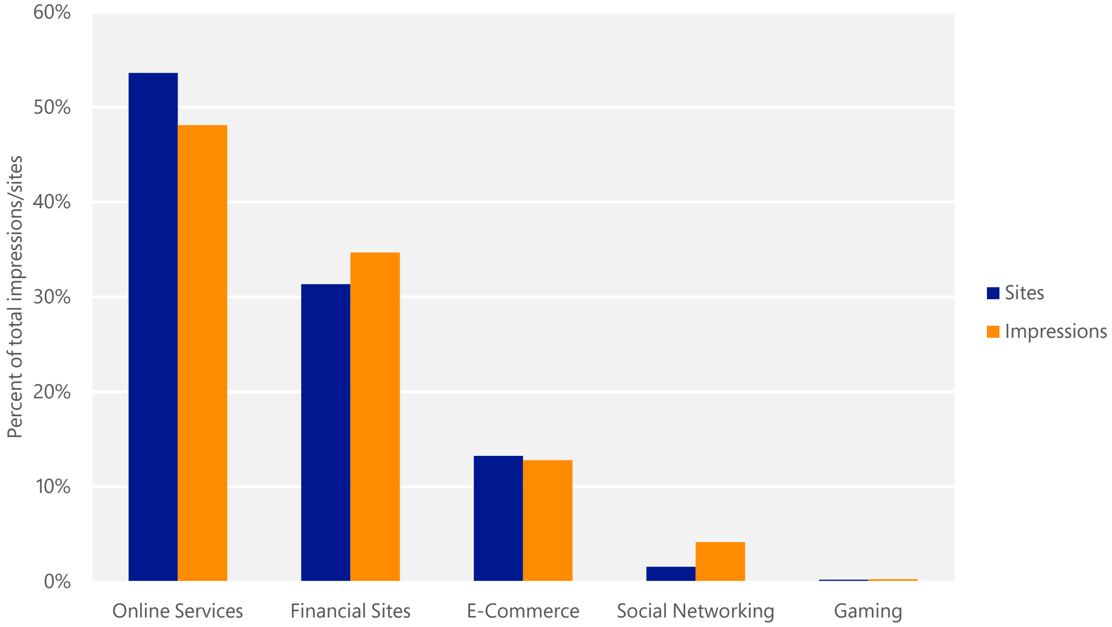


- Numbers of active phishing sites and phishing impressions both increased from July through October, which indicates a general increase in phishing activity, and then declined through the end of the year. However, because phishers are frequently observed using campaigns to drive large amounts of traffic to a relatively small number of pages, the two metrics are generally not strongly correlated, and the dual rise and fall may be at least partially coincidental.

Target institutions

Some types of sites tend to consistently draw many more impressions per site than others. Figure 94 shows the breakdown of phishing impressions by category as reported by SmartScreen Filter.

Figure 94. Phishing sites and impressions reported by SmartScreen Filter for each type of phishing site in 2H15

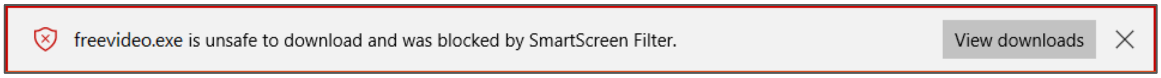


- Phishing sites that targeted online services received the largest share of impressions during the period, and accounted for the largest number of active phishing URLs.
- Financial institutions have always been popular phishing targets because of their potential for providing direct illicit access to victims' bank accounts. Sites that targeted financial institutions accounted for the largest number of active phishing attacks during the period, as well as the second largest number of impressions.
- The other three categories each accounted for a small percentage of both sites and impressions.

Malware hosting sites

SmartScreen Filter helps provide protection against sites that are known to host malware, in addition to phishing sites. SmartScreen Filter uses file and URL reputation data and Microsoft antimalware technologies to determine whether sites distribute unsafe content. As with phishing sites, Microsoft collects anonymized data regarding how many people visit each malware hosting site and uses the information to improve SmartScreen Filter and to better combat malware distribution.

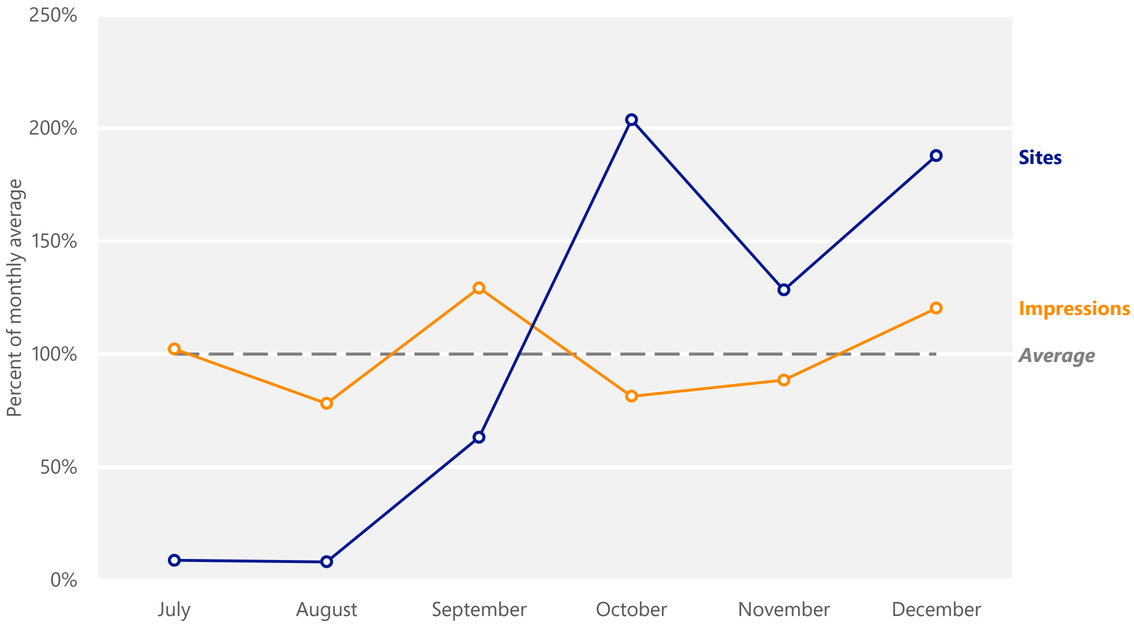
Figure 95. SmartScreen Filter in Microsoft Edge and Internet Explorer displays a warning when a user attempts to download an unsafe file



freevideo.exe is unsafe to download and was blocked by SmartScreen Filter.

Figure 96 compares the volume of active malware hosting sites in the Microsoft database each month with the volume of malware impressions tracked.

Figure 96. Malware hosting sites and impressions tracked each month in 2H15, relative to the monthly average for each



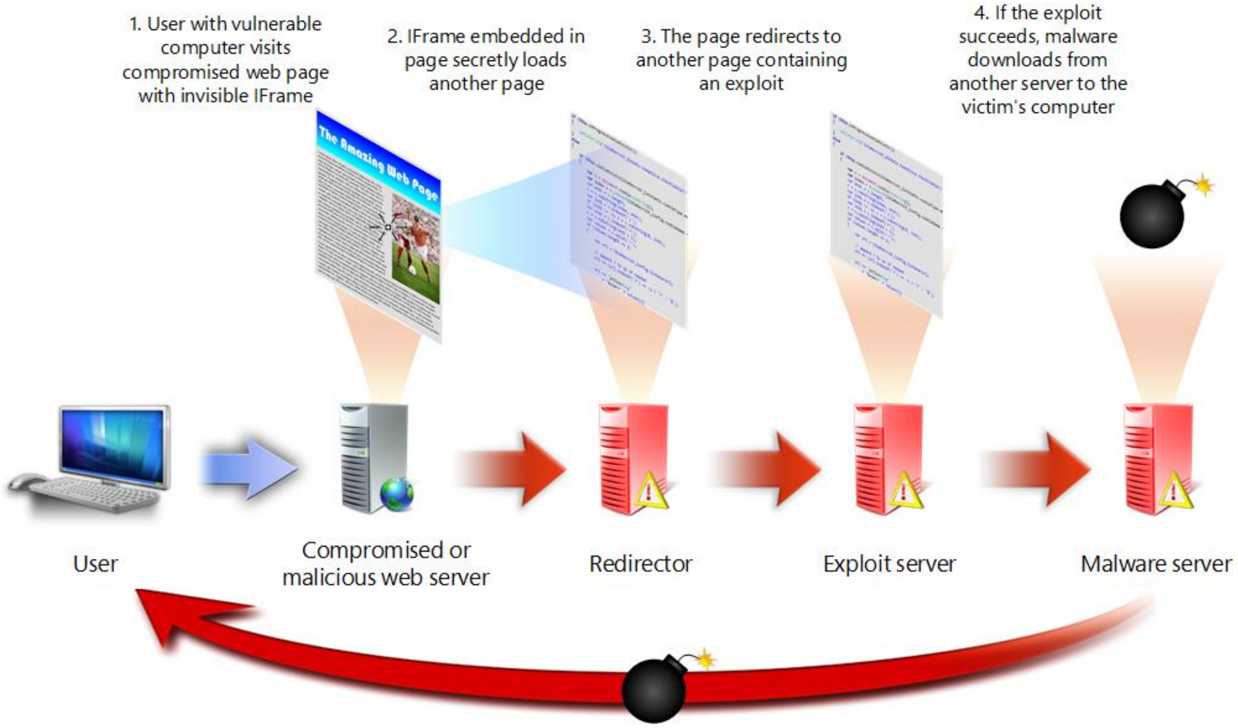
- The number of active malware hosting sites increased by more than 25 times between August and October, correlated with an attack campaign that compromised thousands of sites running the WordPress content management system (CMS) beginning in September, which resulted in large numbers of new exploit kit landing pages containing drive-by downloads for popular browser add-ons. (See "Exploit kits" on page 66 for more information about exploit kit landing pages.)

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.

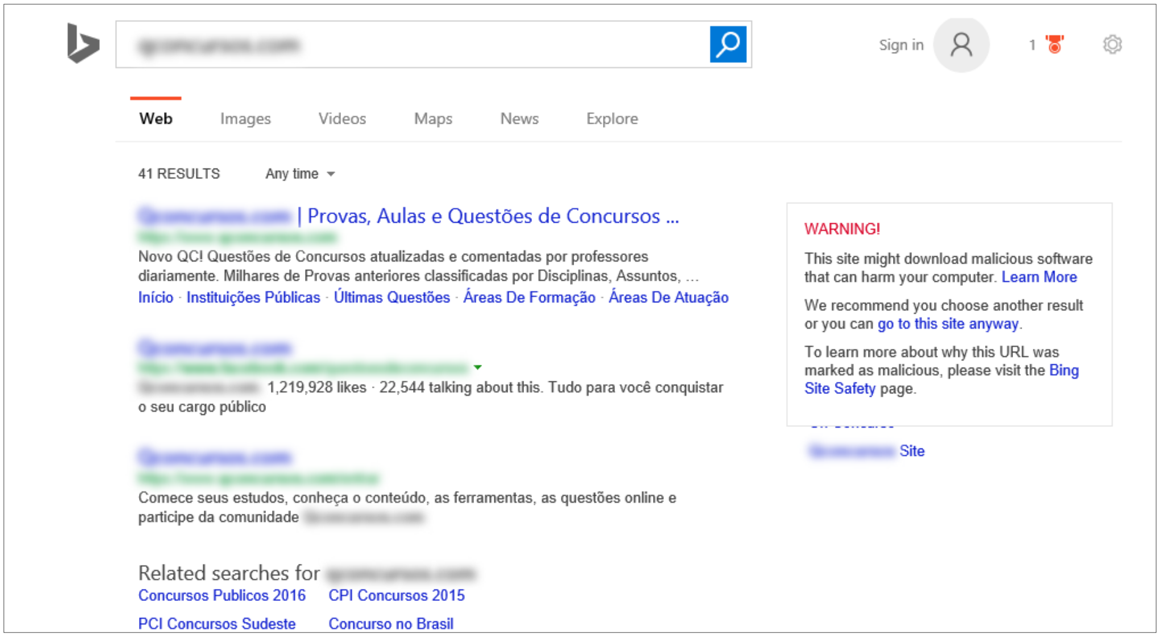
Drive-by download pages are usually hosted on legitimate websites to which an attacker has posted exploit code. Attackers gain access to legitimate sites through intrusion or by posting malicious code to a poorly secured web form, like a comment field on a blog. Compromised sites can be hosted anywhere in the world and concern nearly any subject imaginable, making it difficult for even an experienced user to identify a compromised site from a list of search results.

Figure 97. One example of a drive-by download attack


1. User with vulnerable computer visits compromised web page with invisible IFrame
2. IFrame embedded in page secretly loads another page
3. The page redirects to another page containing an exploit
4. If the exploit succeeds, malware downloads from another server to the victim's computer
The Amazing Web Page
User
Compromised or malicious web server
Redirector
Exploit server
Malware server

Search engines such as Bing have taken a number of measures to help protect users from drive-by downloads. As Bing indexes webpages, they are assessed for malicious elements or malicious behavior. Because the owners of compromised sites are usually victims themselves, the sites are not removed from the Bing index. Instead, clicking the link in the list of search results displays a prominent warning, saying that the page may contain malicious software, as shown in Figure 98.

Figure 98. A drive-by download warning from Bing



[redacted] | Provas, Aulas e Questões de Concursos ...
[redacted]
Novo QC! Questões de Concursos atualizadas e comentadas por professores diariamente. Milhares de Provas anteriores classificadas por Disciplinas, Assuntos, ...
Início · Instituições Públicas · Últimas Questões · Áreas De Formação · Áreas De Atuação

[redacted]
[redacted]
[redacted] 1,219,928 likes · 22,544 talking about this. Tudo para você conquistar o seu cargo público

[redacted]
[redacted]
Comece seus estudos, conheça o conteúdo, as ferramentas, as questões online e participe da comunidade [redacted]

Related searches for [redacted]
Concursos Publicos 2016 CPI Concursos 2015
PCI Concursos Sudeste Concurso no Brasil

WARNING!

This site might download malicious software that can harm your computer. Learn More

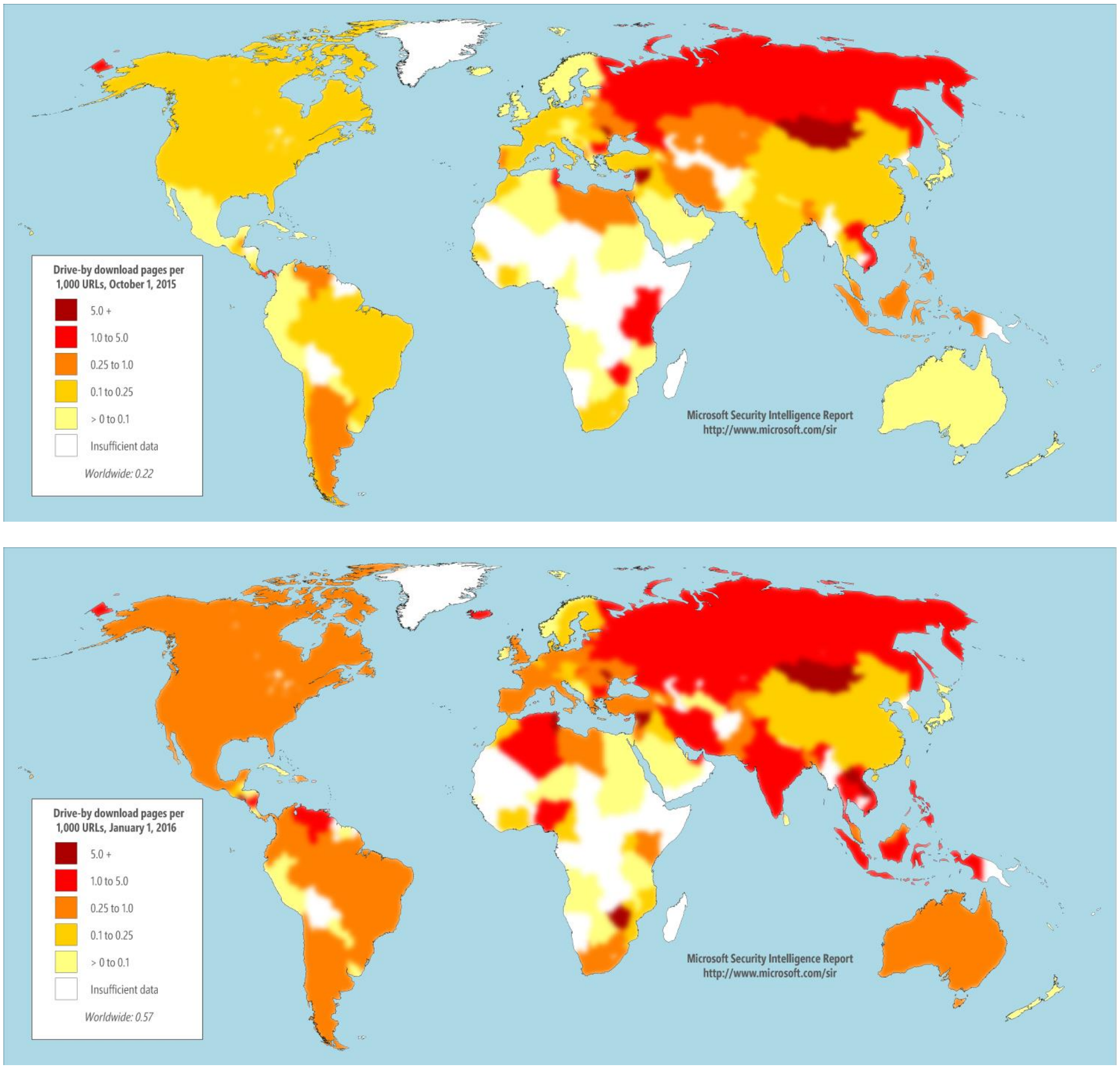
We recommend you choose another result or you can go to this site anyway.

To learn more about why this URL was marked as malicious, please visit the Bing Site Safety page.

[redacted] Site

Figure 99 shows the concentration of drive-by download pages in countries and regions throughout the world at the end of 3Q15 and 4Q15, respectively.

Figure 99. Drive-by download pages indexed by Bing at the end of 3Q15 (top) and 4Q15 (bottom), per 1,000 URLs in each country/region


Drive-by download pages per 1,000 URLs, October 1, 2015
5.0 +
1.0 to 5.0
0.25 to 1.0
0.1 to 0.25
> 0 to 0.1
Insufficient data
Worldwide: 0.22
Microsoft Security Intelligence Report
http://www.microsoft.com/sir
Drive-by download pages per 1,000 URLs, January 1, 2016
5.0 +
1.0 to 5.0
0.25 to 1.0
0.1 to 0.25
> 0 to 0.1
Insufficient data
Worldwide: 0.57
Microsoft Security Intelligence Report
http://www.microsoft.com/sir

- Each map shows the concentration of drive-by download URLs tracked by Bing in each country or region on a reference date at the end of the associated quarter, expressed as the number of drive-by download URLs per every 1,000 URLs hosted in the country/region.
- The overall number of active drive-by download URLs tracked by Bing at the end of 4Q15 was about 2.6 times as large as at the end of 3Q15. This increase correlated with an attack campaign that compromised thousands of sites running the WordPress content management system (CMS) beginning in September, which resulted in large numbers of new exploit kit

landing pages containing drive-by downloads for popular browser add-ons. (See "Exploit kits" on page 66 for more information about exploit kit landing pages.)

- Significant locations with high concentrations of drive-by download URLs in both quarters include Moldova, with 17.2 drive-by URLs for every 1,000 URLs tracked by Bing at the end of 4Q15; Cyprus, with 2.6; and Russia, with 1.8.

> The increase correlated with an attack campaign that compromised thousands of WordPress sites beginning in September.

Guidance: Protecting users from unsafe websites

One of the best ways organizations can protect their users from malicious and compromised websites is by mandating the use of web browsers with appropriate protection features built in and by promoting safe browsing practices. For in-depth guidance, see "Top security solutions" at www.microsoft.com/security/pc-security/solutions.aspx.

Mitigating risk

Malware at Microsoft: Dealing with threats in the Microsoft environment

Microsoft IT

Microsoft IT provides information technology services internally for Microsoft employees and resources. Microsoft IT manages more than 600,000 devices for more than 150,000 users across more than 100 countries and regions worldwide. Safeguarding a computing infrastructure of this size requires implementation of strong security policies, technology to help keep malware off the network and away from mission-critical resources, and dealing with malware outbreaks swiftly and comprehensively when they occur.

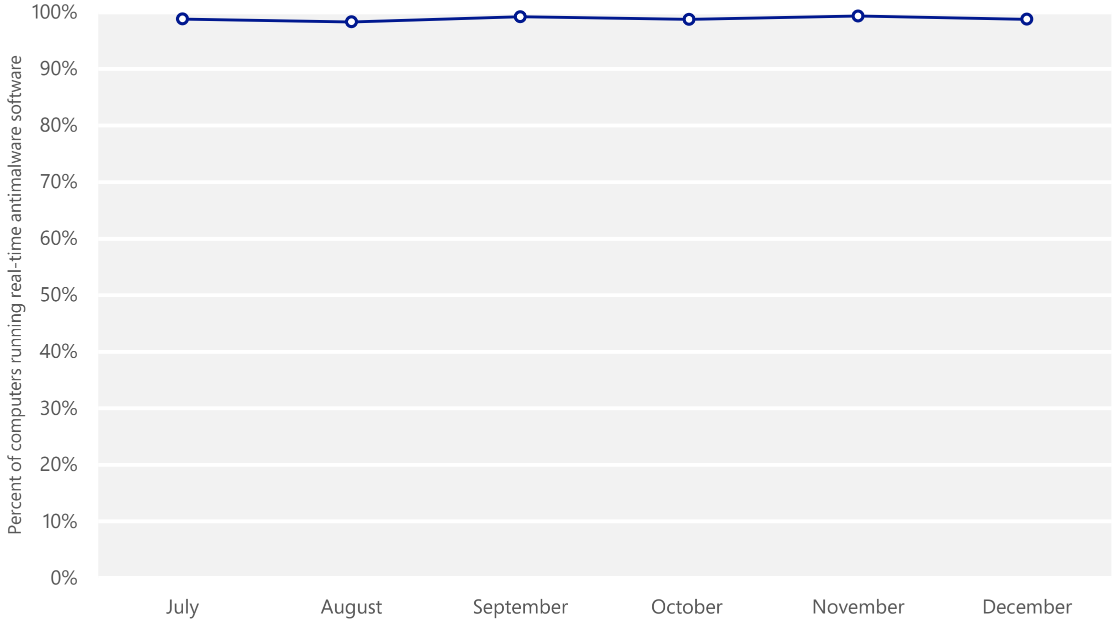
This section of the report compares the potential impact of malware to the levels of antimalware compliance from more than 600,000 workstation computers and devices managed by Microsoft IT between July and December 2015. This data is compiled from multiple sources, including Windows Defender, System Center Endpoint Protection (SCEP), DirectAccess, forensics, and manual submission of suspicious files. Comparing the nature and volume of the malware detected on these computers to the level of protection they receive can illustrate significant trends and provide insights as to the effectiveness of antimalware software and security best practices.

Antimalware usage

Real-time antimalware software is required on all user devices that connect to the Microsoft corporate network. Windows Defender and System Center Endpoint Protection 2012 (SCEP) are the antimalware solutions that Microsoft IT deploys to its users. To be considered compliant with antimalware policies and standards, user computers must be running the latest version of the Defender or SCEP client, antimalware signatures must be no more than six days old, and real-time protection must be enabled.

Figure 100 shows the level of antimalware compliance in the Microsoft user workstation environment for each month in 2H15.

Figure 100. Percentage of computers at Microsoft running real-time antimalware software each month in 2H15



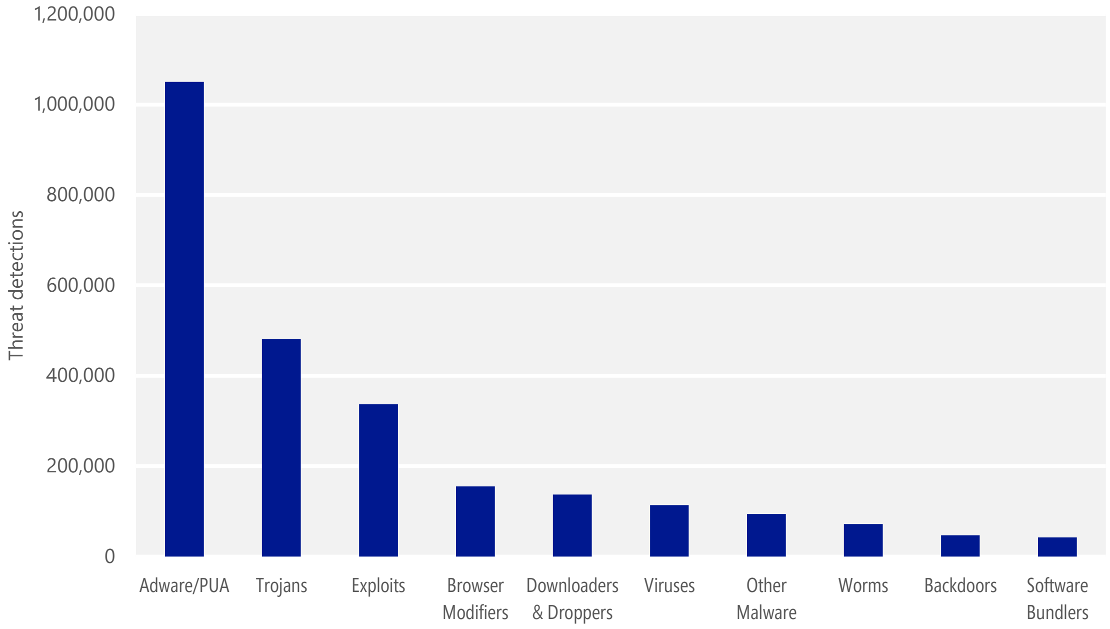
The average monthly compliance rate at Microsoft exceeded 98 percent each month during the second half of the year, reaching a high of 99.4 percent in November. In any network of this size, it is almost inevitable that a small number of computers will be in a noncompliant state at any given time. In most cases, these are computers that are being rebuilt or are otherwise in a state of change when online, rather than computers that have had their antimalware software intentionally disabled.

Microsoft IT believes that a compliance rate in excess of 98 percent among approximately half a million computers is an acceptable level of compliance. In most cases, attempting to boost a large organization’s compliance rate the rest of the way to 100 percent will likely be a costly endeavor, and the end result—100 percent compliance—will be unsustainable over time.

Malware detections

Figure 101 shows the categories of malware and unwanted software that were most frequently detected at Microsoft in 2H15.

Figure 101. Top categories of malware and unwanted software detected by Windows Defender and System Center Endpoint Protection at Microsoft in 2H15

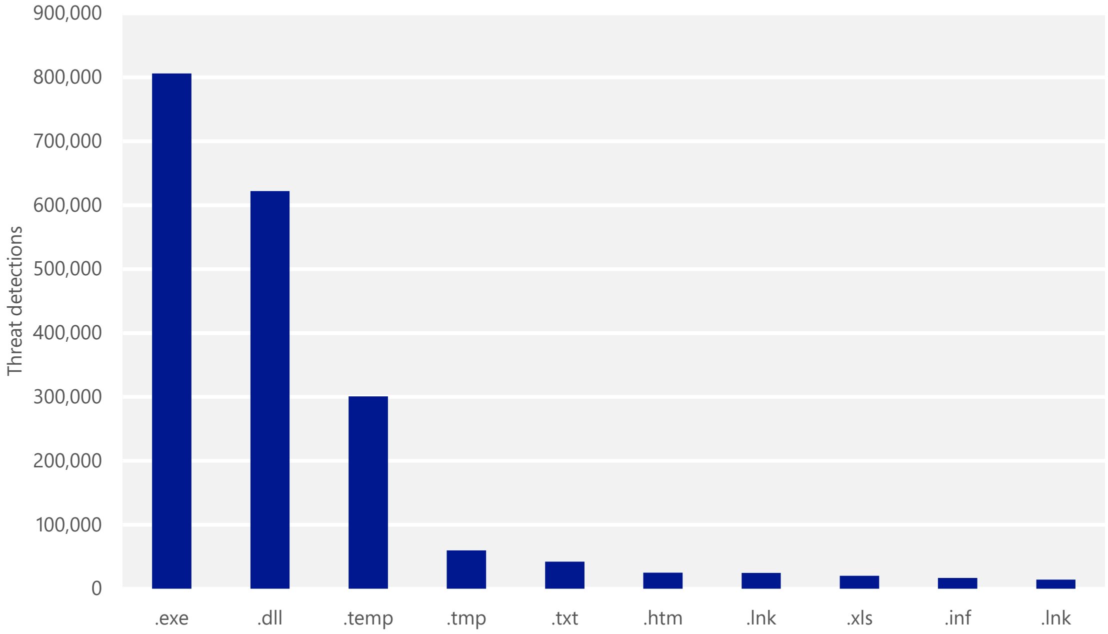


In this section, malware detections are defined as files and processes flagged by SCEP, regardless of the success or failure of automated containment or remediation. Malware detections are a measure of attempted malware activity, and do not necessarily indicate that a computer has been successfully infected. (Note that the methodology for assessing encounters used elsewhere in this report counts unique computers with detections, an approach that differs from the methodology used in this section, in which individual detections are counted. For example, if a computer encountered one trojan family in February and another one in June, it would only be counted once for the purposes of figures such as Figure 59 on page 94. In the preceding Figure 101, it would be counted twice, once for each detection.)

Adware and potentially unwanted applications (PUA) accounted for the largest number of detections, with twice as many detections as the next most prevalent category. The large number of internal adware and PUA detections is caused by a pilot project that MSIT has undertaken with the Microsoft Security Response Center (MSRC) to improve detection of adware and other unwanted software. (See “Potentially unwanted applications in the enterprise” on page 114 for details about this project and how enterprise customers can block PUA in their networks.)

Figure 102 shows the top 10 file types among threat detections at Microsoft in 2H15.

Figure 102. Top ten file types used by threats detected at Microsoft in 2H15



Executable program files with the .exe extension were the most commonly detected type of malicious file at Microsoft in 2H15. Malicious .dll files were the next most common type of threats, followed by the .tmp and .temp extensions, typically used for temporary files.

Transmission vectors

Examining the processes targeted by malware can help illustrate the methods that attackers use to propagate it. Figure 103 lists the top five transmission vectors used by the malware encountered at Microsoft in 2H15.

Figure 103. The top five transmission vectors used by malware encountered at Microsoft in 2H15

Rank	Description
1	File transfers in the operating system
2	Cloud backup/storage
3	Non–operating-system tasks
4	Web browsing
5	Compiling tools

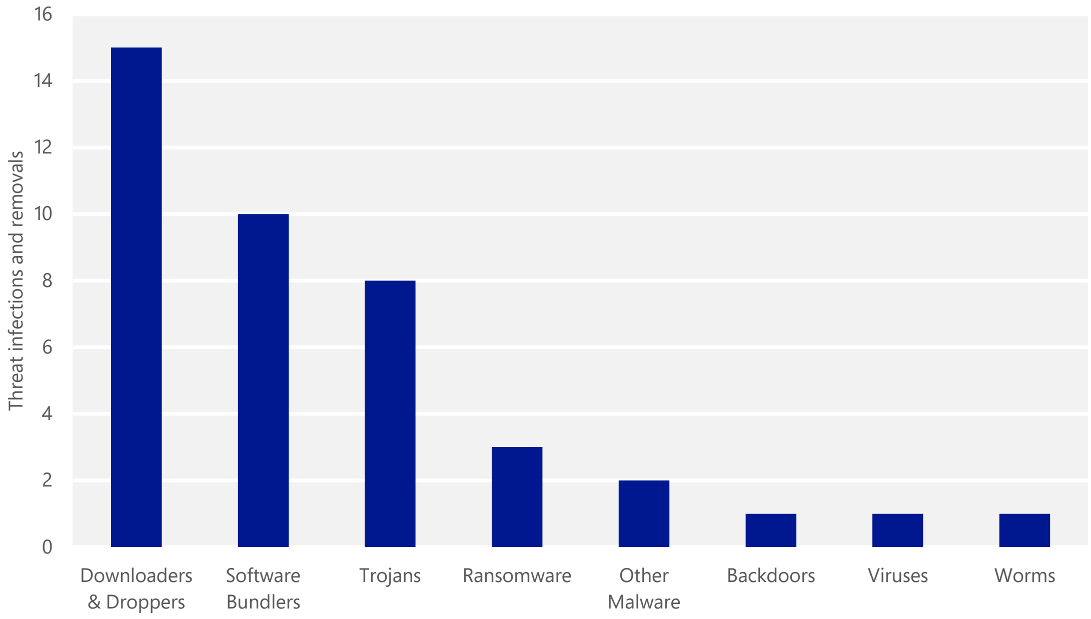
The transmission vector most commonly used by infection attempts detected on Microsoft computers in 2H15 involved file transfers made through Windows Explorer, followed by cloud backup, storage services, and non-operating-system tasks. Web browsing was fourth, followed by compiling tools.

Malware infections

Because almost all of the computers at Microsoft run real-time security software at all times, most infection attempts are detected and blocked before they are able to infect the target computer. When Defender or SCEP do disinfect a computer, it is usually because the software's signature database has been updated to enable it to detect a threat that it did not recognize when the computer first encountered the threat. This lack of recognition may be because the threat is a new malware family, a new variant of a known family, a known variant that has been encrypted or otherwise repackaged to avoid detection, or because of some other reason. The MMPC constantly analyzes malware samples submitted to it, develops appropriate detection signatures, and deploys them to customers who use SCEP, Microsoft Security Essentials, and Windows Defender.

Figure 104 shows the most commonly detected categories of malware and unwanted software that SCEP and Defender removed from computers at Microsoft between July and December of 2015.

Figure 104. Infections and removals at Microsoft in 2H15, by category



As this chart shows, detection and infection statistics were significantly different in 2H15. Adware, which accounted for more than a million detections at Microsoft in 2H15, was not discovered infecting a single computer internally during the period. Most of the other categories also show clear differences between Figure 101 and Figure 104, although the ordering in the latter chart is significantly influenced by the low volumes involved.

Figure 105 shows the top 10 file types used by malware to infect computers at Microsoft in 2H15.

Figure 105. Infections and removals at Microsoft in 2H15, by file type

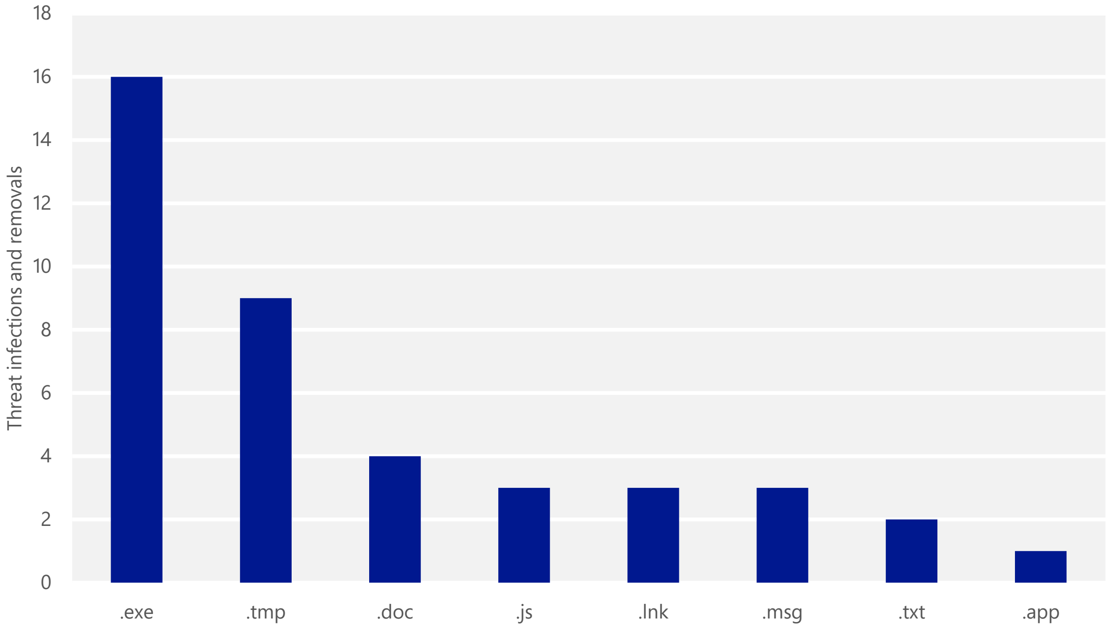
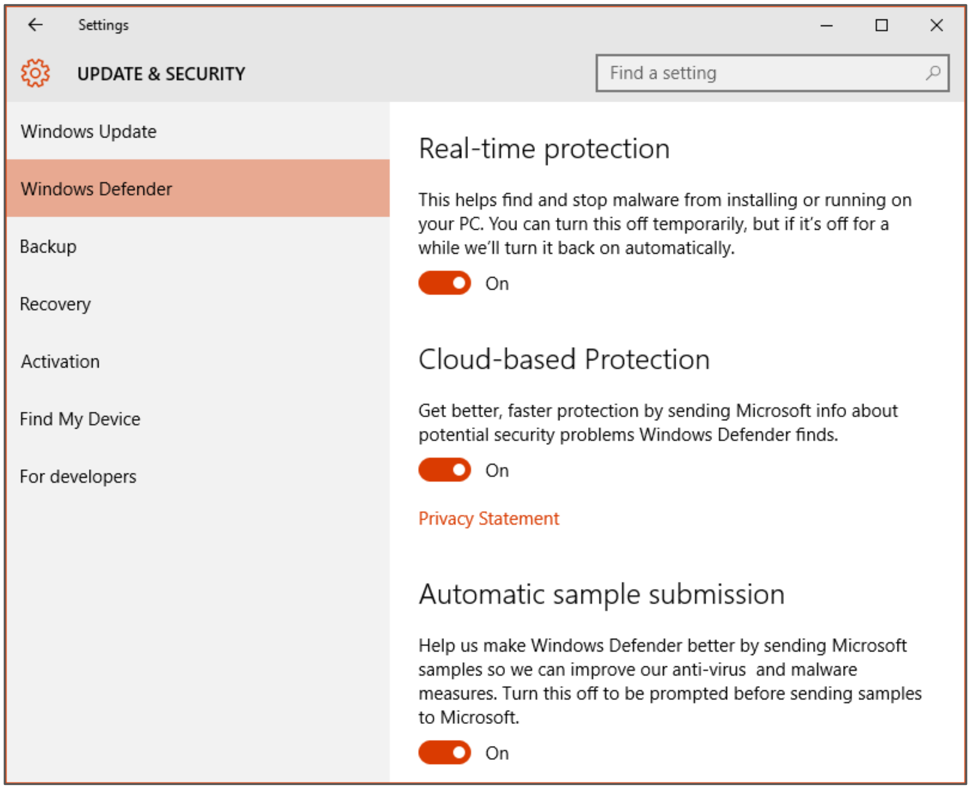


Figure 105 is important because it provides information about threats that Defender and SCEP did not detect when they were first encountered—and therefore provides a clue about the areas in which malware authors have been focusing their efforts in recent months. Almost half of the malicious files removed from computers at Microsoft by Defender and SCEP in 2H15 had the extension .exe, used by executable program files, with seven extensions accounting for the remaining files. The .tmp extension often used for temporary files was next, followed by .doc, used for Microsoft Word binary files. The .js, .lnk, and .msg extensions were each responsible for three removals.

What IT departments can do to protect their users

- Evaluate commercially available management tools, develop a plan, and implement a third-party update mechanism to disseminate non-Microsoft updates.
- Ensure that all software deployed on computers in the environment is updated regularly. If the software provider offers an automatic update utility similar to Microsoft Update, ensure that it is enabled by default. See "Turn automatic updating on or off" at windows.microsoft.com for instructions on enabling automatic updates of Microsoft software.
- Ensure that SmartScreen Filter is enabled in Microsoft Edge and Internet Explorer. See "SmartScreen Filter: FAQ" at windows.microsoft.com for more information.
- Use Group Policy to enforce configurations for Windows Update, Windows Firewall, and SmartScreen Filter. See Knowledge Base article KB328010 at support.microsoft.com, and "Windows Firewall with Advanced Security Deployment Guide" and "Manage Privacy: SmartScreen Filter and Resulting Internet Communication" at technet.microsoft.com for instructions.
- Set the default configuration for antimalware to enable real-time protection across all drives, including removable devices.
- Enable Windows Defender Cloud Protection in Windows 10 to automatically send information about suspicious files and behaviors to the Windows Defender Cloud, which can help identify and block threats during the first critical hours of an attack. For information about using Group Policy to enable MAPS throughout your organization, see Configure Windows Defender in Windows 10 at Microsoft TechNet.

Figure 106. Enabling cloud-based protection for Windows Defender in Windows 10



Real-time protection

This helps find and stop malware from installing or running on your PC. You can turn this off temporarily, but if it's off for a while we'll turn it back on automatically.

On

Cloud-based Protection

Get better, faster protection by sending Microsoft info about potential security problems Windows Defender finds.

On

Privacy Statement

Automatic sample submission

Help us make Windows Defender better by sending Microsoft samples so we can improve our anti-virus and malware measures. Turn this off to be prompted before sending samples to Microsoft.

On

- Identify business dependencies on Java and develop a plan to minimize its use where it is not needed.
- Use AppLocker to block the installation and use of unwanted software such as Java or peer-to-peer (P2P) applications. See "AppLocker" at technet.microsoft.com for more information.
- Implement the Enhanced Mitigation Experience Toolkit (EMET), if possible, to minimize exploitation of vulnerabilities in all software in your environment. See technet.microsoft.com/security/jj653751 for more information.
- Implement strong password policies, and require employees to change their passwords periodically.
- Strengthen authentication by using smart cards. See "Smart Cards" at technet.microsoft.com for more information.
- Use Network Access Protection (NAP) and DirectAccess (DA) to enforce compliance policies for firewall, antimalware, and patch management on remote computers that connect to a corporate network. See "Network

Access Protection" at msdn.microsoft.com and "Windows 7 DirectAccess Explained" at technet.microsoft.com for more information.

- Enable Windows PowerShell v5 security features via Windows Management Framework 5.0:
    - Script block logging
    - System-wide transcripts
    - Constrained PowerShell
    - Antimalware integration (AMSI) in Windows 10

Securing privileged access roadmap

Mark Simos and Nir Ben Zvi

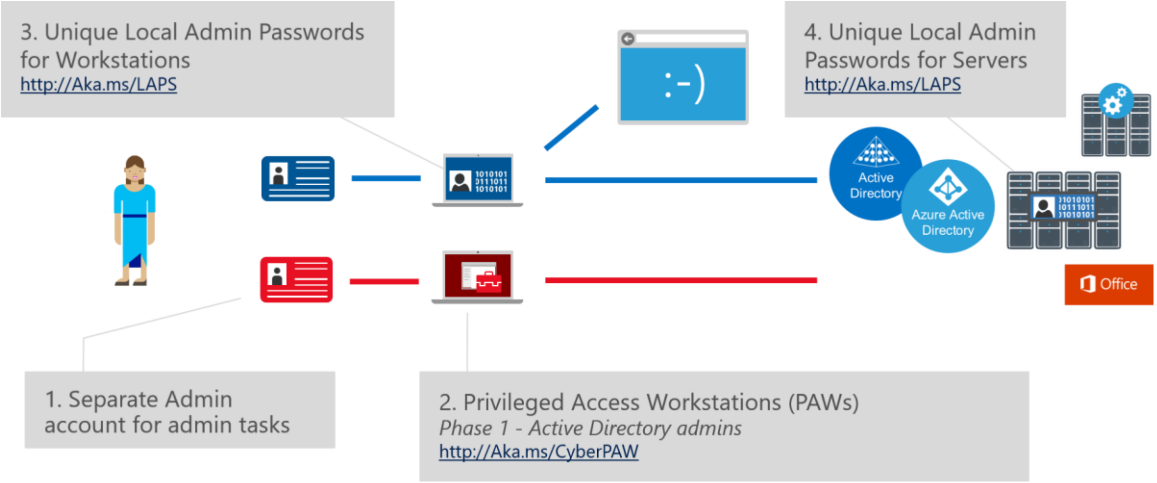
Finding the best way to secure privileged access is a daunting task. What should I do first? How do I make sure it's effective? How can I make incremental progress to get a good return on investment without embarking on a long and expensive journey? These are all valid questions on your journey to secure privileged access.

To address these and related questions, Microsoft recently released actionable step-by-step guidance in the "Securing Privileged Access Roadmap" (available at http://aka.ms/privsec). This guidance provides prioritized steps to help you achieve the best ROI when securing your environment against common attacks. This roadmap is based on field experience observing attacks on corporate and government environments. (See "PLATINUM: Targeted attacks in South and Southeast Asia" beginning on page 3 for an example.)

The guidance format is designed to provide you with the insight that Microsoft senior security architects provide when helping enterprise customers secure their networks against advanced adversaries, so that you can better protect your environment.

The plan presented in the guidance provides guidance about what you should concentrate on immediately in the first 4 weeks, then the priorities for the following 3-month and 6-month periods. Each milestone includes specific steps with links to instructions that will help you complete them. Each milestone helps you achieve a better level of security, which also allows you to measure progress and provide transparency to management on the state of security and what efforts are being made.

Figure 107. Administration establishes a separate channel to isolate privileged access tasks from high risk standard user tasks like web browsing and accessing email


3. Unique Local Admin Passwords for Workstations
http://Aka.ms/LAPS
:-)
4. Unique Local Admin Passwords for Servers
http://Aka.ms/LAPS
Active Directory
Azure Active Directory
Office
1. Separate Admin account for admin tasks
2. Privileged Access Workstations (PAWs)
Phase 1 - Active Directory admins
http://Aka.ms/CyberPAW

This guidance shares what Microsoft has found to be most effective against advanced adversaries. Microsoft plans to continue updating and adapting the guidance as learning continues and as new technologies and solutions come online.

To be effective, the security mitigations include specific actionable elements to change architecture, technology, and operational processes versus just focusing on any single approach. This roadmap is designed to take an organization with a hybrid of both on-premises and cloud assets through the first basic mitigations (which may already be in place) all the way through to measures that will proactively increase adversary costs. Each measure in the roadmap is designed to cut off an access path in your environment that adversaries use today or will attempt if their proven methods are blocked or detected.

Microsoft is also integrating this roadmap approach into its professional services security assessments to help organizations understand where they are on the roadmap in addition to discovering other risks that are specific to their environment.

Microsoft has received a lot of positive feedback on this approach thus far and is planning to continue to develop additional security guidance in this format. Please provide any feedback you might have about how well this format works for you, how to improve it, and anything else you would look for on how to keep your environment secure. (email CyberDocFeedback@microsoft.com)

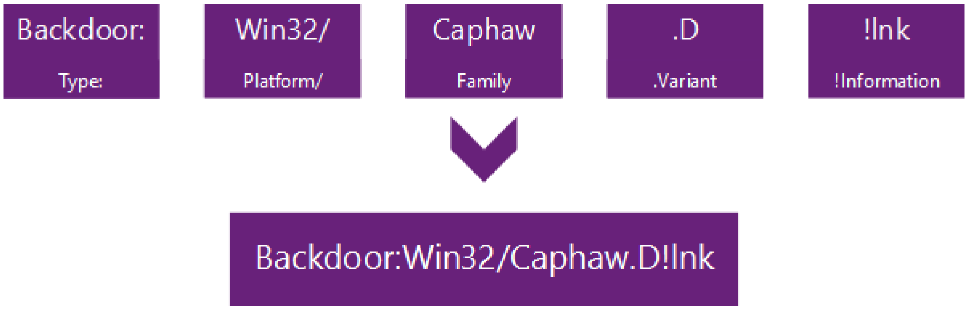
Appendixes

Appendix A: Threat naming conventions

Microsoft names the malware and unwanted software that it detects according to the Computer Antivirus Research Organization (CARO) Malware naming scheme.

This scheme uses the following format:

Figure 108. The Microsoft malware naming convention



When Microsoft analysts research a particular threat, they will determine what each of the components of the name will be.

Type

The type describes what the threat does on a computer. Worms, trojans, and viruses are some of the most common types of threats Microsoft detects.

Platform

The platform refers to the operating system (such as Windows, Mac OS X, and Android) that the threat is designed to work on. Platforms can also include programming languages and file formats.

Family

A group of threats with the same name is known as a family. Sometimes different security software companies use different names.

Variant letters

Variant letters are used sequentially for each different version or member of a family. For example, the detection for the variant ".AF" would have been created after the detection for the variant ".AE."

Additional information

Additional information is sometimes used to describe a specific file or component that is used by another threat in relation to the identified threat. In the preceding example, the !lnk indicates that the threat is a shortcut file used by the Backdoor:Win32/Caphaw.D variant, as shortcut files usually use the extension .lnk.

Appendix B: Data sources

Data included in the *Microsoft Security Intelligence Report* is gathered from a wide range of Microsoft products and services whose users have opted in to provide usage data. The scale and scope of this telemetry data allows the report to deliver the most comprehensive and detailed perspective on the threat landscape that is 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.
- Exchange Online is the Microsoft-hosted email service for business. Exchange Online antimalware and antispam services scan billions of messages every year to identify and block spam and malware.
- The Malicious Software Removal Tool (MSRT) is a free tool that Microsoft designed to help identify and remove specific 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 2H15. The MSRT is not a replacement for an up-to-date real-time antivirus solution.
- 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.
- Microsoft Security Essentials is a free, easy-to-download real-time protection product that provides basic, effective antivirus and antispyware protection for Windows Vista and Windows 7.
- Microsoft System Center Endpoint Protection (formerly Forefront Client Security and Forefront Endpoint Protection) is a unified product that provides protection from malware and 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.

- Office 365 is the Microsoft Office subscription service for business and home users. Select business plans include access to Office 365 Advanced Threat Protection.
- SmartScreen Filter, a feature in Internet Explorer and Microsoft Edge, 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, the browser displays a warning and blocks navigation to the page.
- Windows Defender in Windows 8, Windows 8.1, and Windows 10 provides real-time scanning and removal of malware and unwanted software.
- Windows Defender Offline is a downloadable tool that can be used to create a bootable CD, DVD, or USB flash drive to scan a computer for malware and other threats. It does not offer real-time protection and is not a substitute for an up-to-date antimalware solution.

Figure 109. 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/privacystatement/
Exchange Online, Office 365	www.microsoft.com/online/legal/v2/?docid=43
Internet Explorer 11	windows.microsoft.com/en-us/internet-explorer/ie11-preview-privacy-statement
Malicious Software Removal Tool	www.microsoft.com/security/pc-security/msrt-privacy.aspx
Microsoft Edge	privacy.microsoft.com/en-us/privacystatement/
Microsoft Security Essentials	windows.microsoft.com/en-us/windows/security-essentials-privacy
Microsoft Safety Scanner	www.microsoft.com/security/scanner/en-us/privacy.aspx
System Center Endpoint Protection	https://www.microsoft.com/privacystatement/en-us/SystemCenter2012R2/Default.aspx#tilepspSystemCenter2012R2EndpointProtectionModule
Windows Defender in Windows 10	privacy.microsoft.com/en-us/privacystatement/
Windows Defender Offline	windows.microsoft.com/en-us/windows/windows-defender-offline-privacy

Appendix C: Worldwide encounter and infection rates

"Malware and unwanted software" on page 79 explains how threat patterns differ significantly in different parts of the world. Figure 110 shows the infection and encounter rates for 3Q15 and 4Q15 for locations around the world.[20] See page 65 for information about how infection and encounter rates are calculated.

Figure 110. Encounter and infection rates for locations around the world, 3Q15–4Q15, by quarter (100,000 computers reporting minimum)

Country/region	CCM 3Q15	CCM 4Q15	ER 3Q15	ER 4Q15
Worldwide	*6.1*	*16.9*	*17.8%*	*20.8%*
Albania	26.9	57.8	34.6%	38.3%
Algeria	43.3	62.6	40.6%	52.6%
Angola	36.3	50.3	—	—
Argentina	8.8	36.2	25.6%	26.2%
Armenia	10.1	21.7	29.3%	37.0%
Australia	3.0	17.9	13.8%	15.1%
Austria	3.4	13.3	11.7%	13.3%
Azerbaijan	23.0	43.7	29.6%	37.2%
Bahamas, The	8.0	30.9	—	—
Bahrain	19.6	42.1	24.7%	0.0%
Bangladesh	25.0	40.3	42.5%	57.2%
Barbados	5.1	30.0	—	—
Belarus	7.2	10.1	25.2%	33.6%
Belgium	4.0	23.6	15.4%	16.6%
Bolivia	16.1	49.6	25.9%	36.7%
Bosnia and Herzegovina	13.5	55.3	28.4%	32.7%
Brazil	10.3	22.0	29.2%	34.4%

[20] Encounter rate and CCM are shown for locations with at least 100,000 computers running Microsoft real-time security products and the Malicious Software Removal Tool, respectively, during a quarter. Only computers whose users have opted in to provide data to Microsoft are considered when calculating encounter and infection rates.

Country/region	CCM 3Q15	CCM 4Q15	ER 3Q15	ER 4Q15
Bulgaria	8.0	27.1	27.2%	29.8%
Cambodia	18.2	31.1	39.2%	46.7%
Cameroon	28.7	44.4	—	—
Canada	3.2	17.4	13.1%	15.5%
Chile	10.0	42.1	23.0%	26.1%
China	3.7	4.7	14.9%	18.9%
Colombia	12.7	32.3	23.5%	28.7%
Costa Rica	4.8	30.8	18.6%	21.9%
Côte d'Ivoire	23.9	42.7	38.2%	0.0%
Croatia	6.1	35.2	21.5%	27.5%
Cyprus	7.4	30.2	20.9%	24.4%
Czech Republic	5.6	13.7	15.8%	19.4%
Denmark	2.4	13.0	10.8%	11.7%
Dominican Republic	22.3	54.0	30.6%	35.0%
Ecuador	12.4	44.5	26.0%	33.0%
Egypt	47.2	60.2	39.4%	52.9%
El Salvador	7.9	46.8	23.3%	28.7%
Estonia	4.6	17.3	17.1%	20.6%
Finland	2.3	8.3	7.1%	8.6%
France	5.2	21.2	18.8%	19.4%
Georgia	17.8	31.3	29.8%	33.2%
Germany	3.7	10.1	12.2%	13.8%
Ghana	28.2	49.0	40.6%	53.4%
Greece	6.2	29.5	21.6%	25.4%
Guadeloupe	7.3	22.0	—	—
Guatemala	12.0	44.0	23.0%	27.8%
Honduras	20.0	55.2	26.4%	32.9%
Hong Kong SAR	3.7	22.5	12.6%	15.5%
Hungary	4.5	22.2	21.2%	23.8%
Iceland	2.8	12.8	13.5%	13.9%
India	25.9	53.9	36.5%	44.2%
Indonesia	31.4	72.1	45.2%	60.6%

Country/region	CCM 3Q15	CCM 4Q15	ER 3Q15	ER 4Q15
Iraq	67.9	80.0	38.4%	47.9%
Ireland	3.4	21.1	13.8%	14.5%
Israel	7.9	24.0	21.2%	21.4%
Italy	6.1	21.1	19.8%	22.3%
Jamaica	9.2	41.6	27.2%	31.1%
Japan	3.2	6.6	6.3%	7.8%
Jordan	33.6	64.3	36.5%	45.3%
Kazakhstan	14.6	15.6	26.2%	37.0%
Kenya	21.2	37.0	33.1%	42.0%
Korea	8.9	14.1	12.0%	15.1%
Kuwait	20.0	45.4	26.8%	29.9%
Latvia	3.3	17.8	20.0%	22.3%
Lebanon	28.5	55.1	28.8%	36.5%
Libya	56.8	85.3	—	—
Lithuania	4.5	25.6	22.3%	24.3%
Luxembourg	3.4	10.0	12.4%	14.4%
Macao SAR	4.6	20.5	—	—
Macedonia, FYRO	14.7	52.1	33.5%	35.4%
Malaysia	16.6	51.1	27.0%	33.7%
Malta	4.9	29.3	20.4%	0.0%
Martinique	5.5	20.1	—	—
Mauritius	11.5	44.3	26.6%	27.5%
Mexico	11.8	40.9	23.9%	28.5%
Moldova	8.8	16.8	24.7%	32.3%
Mongolia	62.0	93.3	—	—
Morocco	46.1	69.9	36.4%	47.3%
Mozambique	27.5	44.4	—	—
Namibia	15.3	31.9	—	—
Nepal	39.7	59.2	45.4%	52.1%
Netherlands	3.6	15.3	14.1%	14.7%
New Zealand	3.1	17.2	13.3%	13.5%
Nicaragua	8.8	42.6	—	—

Country/region	CCM 3Q15	CCM 4Q15	ER 3Q15	ER 4Q15
Nigeria	27.1	40.8	31.4%	39.3%
Norway	2.3	12.5	10.1%	11.1%
Palestinian Authority	53.5	80.0	43.5%	57.3%
Oman	26.0	65.9	32.1%	43.0%
Pakistan	46.9	71.3	49.9%	63.0%
Panama	8.3	34.7	22.8%	27.4%
Paraguay	11.9	38.4	24.9%	0.0%
Peru	15.3	49.0	24.6%	32.4%
Philippines	30.9	71.7	37.8%	47.7%
Poland	7.9	18.9	20.8%	26.7%
Portugal	5.0	25.6	26.0%	25.6%
Puerto Rico	6.3	31.9	19.1%	19.1%
Qatar	15.7	41.8	28.0%	32.5%
Réunion	6.8	21.7	17.7%	19.6%
Romania	13.3	36.4	27.4%	31.3%
Russia	5.2	14.1	22.8%	28.7%
Saudi Arabia	20.0	45.6	28.9%	37.9%
Senegal	19.2	36.6	41.2%	51.1%
Serbia	11.0	48.5	29.3%	31.8%
Singapore	4.4	25.1	16.9%	19.8%
Slovakia	7.1	21.0	17.2%	20.6%
Slovenia	4.4	21.6	17.7%	19.2%
South Africa	10.5	27.8	23.1%	27.7%
Spain	7.6	34.0	21.0%	23.3%
Sri Lanka	16.5	38.2	31.3%	38.8%
Sweden	2.6	13.5	10.4%	11.4%
Switzerland	2.5	14.3	11.2%	12.5%
Taiwan	5.6	21.6	16.5%	19.3%
Tanzania	28.7	46.3	43.8%	0.0%
Thailand	22.2	46.3	29.8%	36.7%
Trinidad and Tobago	8.1	37.9	24.2%	25.1%
Tunisia	31.6	59.6	39.1%	47.1%

Country/region	CCM 3Q15	CCM 4Q15	ER 3Q15	ER 4Q15
Turkey	21.0	42.6	32.6%	40.3%
Ukraine	6.5	9.9	27.3%	35.3%
United Arab Emirates	17.9	49.5	29.1%	34.0%
United Kingdom	4.3	15.9	11.9%	13.9%
United States	3.2	12.3	10.8%	12.5%
Uruguay	6.0	39.6	21.6%	25.3%
Venezuela	18.6	43.2	29.5%	34.7%
Vietnam	26.5	39.5	41.2%	50.7%
Zimbabwe	17.8	40.3	—	—
Worldwide	*6.1*	*16.9*	*17.8%*	*20.8%*

Glossary

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

ActiveX control
A software component of Microsoft Windows that can be used to create and distribute small applications through Internet Explorer. ActiveX controls can be developed and used by software to perform functions that would otherwise not be available using typical Internet Explorer capabilities. Because ActiveX controls can be used to perform a wide variety of functions, including downloading and running programs, vulnerabilities discovered in them may be exploited by malware. In addition, cybercriminals may also develop their own ActiveX controls, which can do damage to a computer if a user visits a webpage that contains the malicious ActiveX control.

Address Space Layout Randomization (ASLR)
A security feature in recent versions of Windows that randomizes the memory locations used by system files and other programs, which makes it harder for an attacker to exploit the system by targeting specific memory locations.

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.

ASLR
See *Address Space Layout Randomization (ASLR)*.

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 bots, nodes, or zombies.

browser modifier
A program that changes browser settings, such as the home page, without adequate consent. This also includes browser hijackers.

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 the Microsoft Malicious Software Removal Tool (MSRT). For example, if the MSRT has 50,000 executions in a particular location in the first quarter of the year and removes infections from 200 computers, the CCM for that location in the first quarter of the year is 4.0 (200 ÷ 50,000 × 1,000). Also see *encounter rate*.

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

command and control
See *botnet*.

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.

Data Execution Prevention (DEP)
A security technique designed to prevent buffer overflow attacks. DEP enables the system to mark areas of memory as non-executable, which prevents code in those memory locations from running.

DEP
See *Data Execution Prevention (DEP)*.

detection
The discovery of malware or potentially unwanted software on a computer by antimalware software. Disinfections and blocked infection attempts are both considered detections.

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

detonation chamber
A sandbox environment in which potentially dangerous files can be automatically launched and monitored for possible malicious activity.

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*.

DNS
See *Domain Name System*.

Domain Name System
The infrastructure used for name resolution on the Internet. It comprises a hierarchical collection of name servers which translate alphanumeric domain names to numeric IP addresses, and vice versa.

downloader
See *downloader/dropper*.

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.

encounter
An instance of security software detecting a threat and blocking, quarantining, or removing it from the computer.

encounter rate
The percentage of computers running Microsoft real-time security software that report detecting malware or potentially unwanted software, or report detecting a specific threat or family, during a period. Also see *infection rate*.

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

exploit kit
A collection of exploits bundled together and sold as commercial software. A typical kit contains a collection of web pages that contain exploits for vulnerabilities in popular web browsers and add-ons, along with tools for managing and updating the kit.

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 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.

infection
The presence of malware on a computer, or the act of delivering or installing malware on a computer. Also see *encounter*.

infection rate
See *CCM*.

Internet Relay Chat (IRC)

A distributed real-time Internet chat protocol that is designed for group communication. Many botnets use the IRC protocol for C&C.

jailbreaking

See *rooting*.

keylogger

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

Malicious Software Removal Tool

A free tool that Microsoft designed to help identify and remove specific prevalent malware families from customer computers. An updated version of the tool is released each month through Windows Update and other updating services. The MSRT is not a replacement for an up-to-date real-time antivirus solution.

malware

Short for malicious software. The general name for programs that perform unwanted actions on a computer, such as stealing personal information. Some malware can steal banking details, lock a computer until the user pays a ransom, or use the computer to send spam. Viruses, worms and trojans are all types of malware.

malware impression

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

man-in-the-middle attack

A form of eavesdropping in which a malicious hacker gets in the middle of network communications. The malicious hacker can then manipulate messages or gather information without the people doing the communication knowing.

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)*.

MSRT

See *Malicious Software Removal Tool*.

multifactor authentication
Requiring a user to provide two or more forms of authentication, such as a username/password and a physical token, to access an account.

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

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 and being blocked by SmartScreen Filter in Microsoft Edge or Internet Explorer. Also see *malware impression*.

ransomware
A type of malware that prevents use of a computer or access to the data that it contains until the user pays a certain amount to a remote attacker (the "ransom"). Computers that have ransomware installed usually display a screen containing information on how to pay the "ransom." A user cannot usually access anything on the computer beyond the screen.

return-oriented programming (ROP)
An exploit technique that involves gaining control of a program's control flow and calling a chain of instructions that already exist in memory, each of which ends in a return command.

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.

rooting

Obtaining administrative user rights on a mobile device through the use of exploits. Device owners sometimes use such exploits intentionally to gain access to additional functionality, but these exploits can also be used by attackers to infect devices with malware that bypasses many typical security systems. The term "rooting" is typically used in the context of Android devices; the comparable process on iOS devices is more commonly referred to as jailbreaking.

ROP

See *return-oriented programming (ROP)*.

sandbox

A specially constructed portion of a computing environment in which potentially dangerous programs or processes may run without causing harm to resources outside the sandbox.

signature

See *detection signature*.

sinkhole

A server or set of servers designed to absorb and analyze malware traffic.

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.

software bundler

A program that installs unwanted software on a computer at the same time as the software the user is trying to install, without adequate consent.

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.

spear phishing
Phishing that targets a specific person, organization, or group, containing additional information associated with that person, organization, or group to lure the target further into a false sense of security to divulge more sensitive information.

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

targeted attack
A malware attack against a specific group of companies or individuals. This type of attack usually aims to get access to the computer or network, before trying to steal information or disrupt the infected machines.

tool
In the context of malware, a software program 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.

two-step verification
See *multifactor authentication*.

unwanted software
A program with potentially unwanted functionality that may affect the user’s privacy, security, or computing experience.

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.

XSS
See *cross-site scripting*.

zero-day exploit
An exploit that targets a zero-day vulnerability.

zero-day vulnerability
A vulnerability in a software product for which the vendor has not yet published a security update.

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 unwanted software families. See the encyclopedia for more in-depth information and guidance for the families listed here and throughout the report.

W97M/Adnel. A family of macro malware that can download other threats to the computer, including TrojanDownloader:Win32/Drixed.

Win32/Anogre. A detection for the Sweet Orange exploit kit, which exploits vulnerabilities in some versions of Windows, Adobe Flash Player, and Java to install malware.

INF/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.

JS/Axpergle. A detection for the Angler exploit kit, which exploits vulnerabilities in some versions of Internet Explorer, Silverlight, Adobe Flash Player, and Java to install malware.

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

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

Win32/Bayads. A program that displays ads as the user browses the web. It can be bundled with other software. It may call itself bdraw, delta, dlclient, Pay-By-Ads, or pricehorse in Programs and Features.

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

JS/Blacole. An exploit pack, also known as Blackhole, that is installed on a compromised web server by an attacker and includes a number of exploits that target browser software. If a vulnerable computer browses a compromised website that contains the exploit pack, various malware may be downloaded and run.

MSIL/Bladabindi. A family of backdoors created by a malicious hacker tool called NJ Rat. They can steal sensitive information, download other malware, and allow backdoor access to an infected computer.

JS/Bondat. A family of threats that collects information about the computer, infects removable drives, and tries to stop the user from accessing files. It spreads by infecting removable drives, such as USB thumb drives and flash drives.

ALisp/Bursted. A virus written in the AutoLISP scripting language used by the AutoCAD computer-aided design program. It infects other AutoLISP files with the extension .lsp.

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/CouponRuc. A browser modifier that changes browser settings and may also modify some computer and Internet settings.

Win32/CplLnk. A generic detection for specially-crafted malicious shortcut files that attempt to exploit the vulnerability addressed by Microsoft Security Bulletin MS10-046.

Win32/Crowti. A ransomware family that encrypts files on the computer and demands that the user pay a fee to decrypt them, using Bitcoins.

Win32/Diplugem. A browser modifier that installs browser add-ons without obtaining the user's consent. The add-ons show extra advertisements as the user browses the web, and can inject additional ads into web search results pages.

Win32/Dipsind. A threat that is often used in targeted attacks. It can give an attacker access to the computer to download and run files, steal domain credentials, and perform other malicious actions.

W97M/Donoff. A threat that uses an infected Microsoft Office file to download other malware onto the computer. It can arrive as a spam email attachment, usually as a Word file (.doc).

Win32/Dorkbot. A worm that spreads via instant messaging and removable drives. It also contains backdoor functionality that allows unauthorized access and control of the affected computer. Win32/Dorkbot may be distributed from compromised or malicious websites using PDF or browser exploits.

Win32/Dowadmin. A software bundler that does not provide the user with the option to decline installation of unwanted software.

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

JS/FakeCall. Rogue security software in the form of a webpage that claims the computer is infected with malware. It asks the user to phone a number to receive technical support to help remove the malware.

Win32/Fourthrem. A program that installs unwanted software without adequate consent on the computer at the same time as the software the user is trying to install.

Win32/Gamarue. A worm that is commonly distributed via exploit kits and social engineering. Variants have been observed stealing information from the local computer and communicating with command-and-control (C&C) servers managed by attackers.

AndroidOS/GingerMaster. 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/Ippedo. A worm that can send sensitive information to a malicious hacker. It spreads through removable drives, such as USB flash drives.

DOS/JackTheRipper. A virus that can stop some files from working correctly in Windows XP and earlier operating systems. It spreads by infecting the master boot record (MBR) on connected hard disks and floppy disks.

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.

VBS/Jenxcus. A worm that gives an attacker control of the computer. It is spread by infected removable drives, like USB flash drives. It can also be downloaded within a torrent file.

ALisp/Kenilfe. A worm written in AutoCAD Lisp that only runs if AutoCAD is installed on the computer or network. It renames and deletes certain AutoCAD files, and may download and execute arbitrary files from a remote host.

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

HTML/Meadgive. A detection for the RIG exploit kit, also known as Redkit, Infinity, and Goon. It attempts to exploit vulnerabilities in programs such as Java and Silverlight to install other malware.

Win32/Mytonel. A program that downloads and installs other programs onto the computer without the user's consent, including other malware.

JS/Neclu. A detection for the Nuclear exploit kit, which attempts to exploit vulnerabilities in programs such as Java and Adobe Reader to install other malware.

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/Ogimant. A threat that claims to help download items from the Internet, but actually downloads and runs files that are specified by a remote attacker.

Win32/OutBrowse. A software bundler that installs additional unwanted programs alongside software that the user wishes to install. It can remove or hide the installer's close button, leaving no way to decline the additional applications.

Win32/Peals. A generic detection for various threats that display trojan characteristics.

Win32/Peapoon. An adware program that shows users ads that they cannot control as they browse the web. It may identify itself as Coupon in Programs and Features.

Win32/Pokki. A browser add-on that formerly displayed behaviors of unwanted software. Recent versions of the add-on no longer meet Microsoft detection criteria, and are no longer considered unwanted software.

Win32/Putalol. An adware program that shows users ads that they cannot control as they browse the web. It may identify itself as Lolliscan in Programs and Features.

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/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.

PHP/SimpleShell. A backdoor that can give an attacker the ability to run shell commands on a compromised server.

Win32/Skeeyah. A generic detection for various threats that display trojan characteristics.

Win32/Sulunch. A generic detection for a group of trojans that perform a number of common malware behaviors.

Win32/SupTab. A browser modifier that installs itself and changes the browser's default search provider, without obtaining the user's consent for either action.

Win32/Tillail. A software bundler that installs unwanted software alongside the software the user is trying to install. It has been observed to install the browser modifier Win32/SupTab.

Win32/Tupym. A worm that copies itself to the system folder of the affected computer, and attempts to contact remote hosts.

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

Win32/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/Wecykler. A family of worms that spread via removable drives, such as USB drives; they may stop security processes and other processes on the computer, and log keystrokes which they later send to a remote attacker.

Win32/Xiazai. A program that installs unwanted software on the computer at the same time as the software the user is trying to install, without adequate consent.

Index

Microsoft

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