Microsoft

Microsoft Security Intelligence Report

Volume 16 | July through December, 2013

Featured Intelligence

Authors

Dennis Batchelder
Microsoft Malware Protection Center

Joe Blackbird
Microsoft Malware Protection Center

David Felstead
Bing

Paul Henry
Wadeware LLC

Jeff Jones
Microsoft Trustworthy Computing

Aneesh Kulkarni
Windows Services Safety Platform

John Lambert
Microsoft Trustworthy Computing

Marc Lauricella
Microsoft Trustworthy Computing

Ken Malcolmson
Microsoft Trustworthy Computing

Matt Miller
Microsoft Trustworthy Computing

Nam Ng
Microsoft Trustworthy Computing

Daryl Pecelj
Microsoft IT Information Security and Risk Management

Tim Rains
Microsoft Trustworthy Computing

Vidya Sekhar
Microsoft Malware Protection Center

Holly Stewart
Microsoft Malware Protection Center

Todd Thompson
Microsoft IT Information Security and Risk Management

David Weston
Microsoft Operating Systems Group

Terry Zink
Exchange Online Protection

Contributors

Hyun Choi
Joe Faulhaber
Tanmay Ganacharya
Ben Hope
Aaron Hulett
Hong Jia
Marianne Mallen
Geoff McDonald
Scott Molenkamp
Dolcita Montemayor
Hamish O'Dea
Bill Pfeifer
Dmitriy Pletnev
Hilda Larina Ragragio
Shawn Wang
Iaan Wiltshire
Dan Wolff
Microsoft Malware Protection Center

Joe Gura
Microsoft Trustworthy Computing

Chris Hale
Microsoft Trustworthy Computing

Satomi Hayakawa
CSS Japan Security Response Team

Yurika Kakiuchi
CSS Japan Security Response Team

Jimmy Kuo
Wadeware LLC

Greg Lenti
Microsoft Trustworthy Computing

Chad Mills
Windows Services Safety Platform

Daric Morton
Microsoft Services

Takumi Onodera
Microsoft Premier Field Engineering, Japan

Anthony Penta
Windows Services Safety Platform

Cynthia Sandvick
Microsoft Trustworthy Computing

Richard Saunders
Microsoft Trustworthy Computing

Frank Simorjay
Microsoft Trustworthy Computing

Norie Tamura
CSS Japan Security Response Team

Henk van Roest
CSS Security EMEA

Steve Wacker
Wadeware LLC

Table of contents

About this report

The *Microsoft Security Intelligence Report (SIR)* focuses on software vulnerabilities, software vulnerability exploits, and malicious 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 2013, 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, 1H13 represents the first half of 2013 (January 1 through June 30), and 4Q12 represents the fourth quarter of 2012 (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" in the full report. 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 family or variant that is detected by the Microsoft Malware Protection Engine.

Trustworthy Computing:
Security engineering at Microsoft

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

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

Exploitation trends

From potential risk to actual risk

Effective risk management requires having enough information about potential threats to accurately assess both their likelihood and consequences. Microsoft is committed to helping customers assess the risk they face from vulnerabilities.

The Microsoft Security Bulletins and Microsoft Security Advisories that are issued each month give IT professionals the latest information about vulnerabilities, the products they affect, and any security updates or actions they can implement to mitigate related risks. For the past several years, Microsoft Security Bulletins have also included Exploitability Index ratings designed to help customers assess not only the severity of vulnerabilities, but the likelihood that a given vulnerability will be exploited in the wild within the first 30 days of a bulletin's release. For example, a critical vulnerability that would be difficult and costly for an attacker to exploit may be less likely to be exploited than a less severe vulnerability that is easier to exploit. Microsoft believes that providing customers with comprehensive and relevant information about vulnerabilities can help make the entire computing ecosystem safer, by reducing the return on investment that attackers expect to gain from exploiting vulnerabilities.

Although forward-looking mechanisms such as Security Bulletins and the Exploitability Index can help customers assess the potential risk they face from software vulnerabilities, reviewing past vulnerabilities that have actually been exploited can help put that risk into perspective. To that end, Microsoft researchers have studied some of the exploits that have been discovered over the past several years and the vulnerabilities they targeted. Understanding which vulnerabilities get exploited, who exploits them, how they do it, and when vulnerabilities are exploited is key to accurately assessing the risk that they pose.

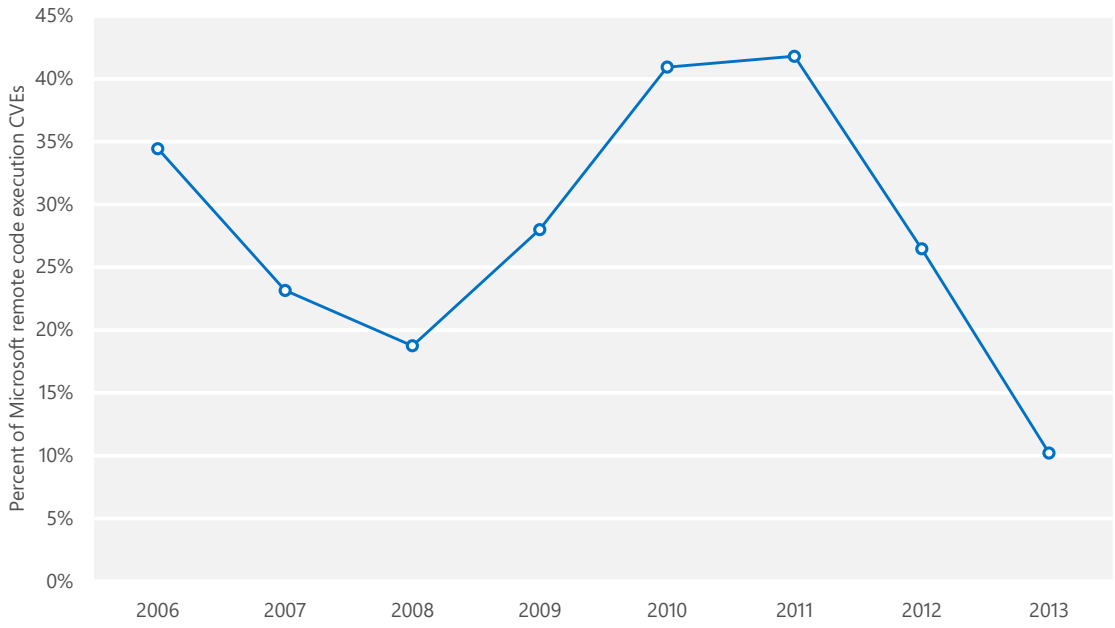
> In the modern era, the profit motive underlies most malicious exploitation activity.

Putting exploits into perspective

In the modern era, the profit motive underlies most malicious exploitation activity. "Black hat" researchers and exploit developers sell access to vulnerability

information and exploit code, and attackers use exploits to deliver malware to victims' computers for use in illegitimate endeavors such as sending spam, credential theft, and many other profit-making schemes. For this reason, vulnerabilities often go unexploited if they would cost more to successfully exploit than an attacker is likely to make from doing it. For example, some vulnerabilities can only be exploited under very limited and uncommon conditions; others do not provide an attacker with access to enough of the computer's functionality to be worthwhile. As Figure 1 shows, even some of the most dangerous vulnerabilities—those that allow an attacker to remotely execute arbitrary code on the victim's computer—only get exploited in a minority of cases.

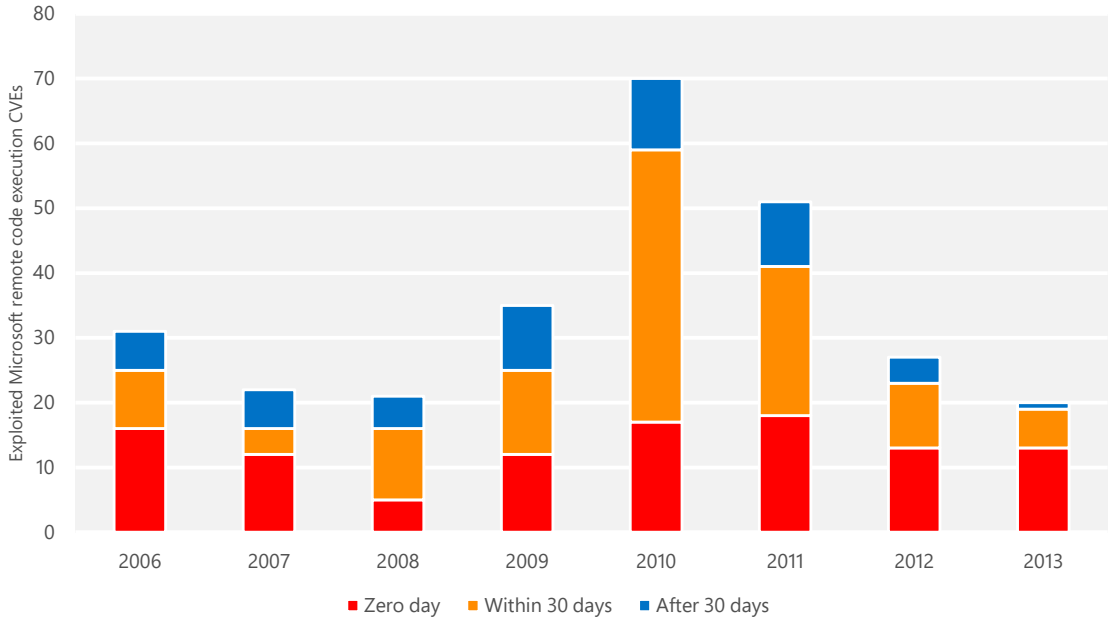
Figure 1. Percent of Microsoft remote code execution CVEs with known exploits, by year of security bulletin



When vulnerabilities are exploited

Of those vulnerabilities that do get exploited, the greatest potential risk comes from *zero-day* exploits, which are discovered in the wild before the publisher of the affected software is able to release a security update to address the vulnerability.

Figure 2. Microsoft remote code execution CVEs, 2006–2013, by timing of first known exploit



As Figure 2 shows, the number of zero-day exploits detected each year has decreased since 2011 in absolute terms; subsequently, zero-day exploits have accounted for a larger share of the total in each of the last three years, and now account for the bulk of all exploited Microsoft remote code execution CVEs. With new remote code execution vulnerabilities becoming harder to find and exploit as secure coding practices improve across the software industry, the value of previously undisclosed exploits in the underground economy has increased, and developing new exploits has become more expensive. This reality provides "black hat" security researchers and exploit developers with a powerful incentive to maximize their own profits by selling exclusive access to a vulnerability and exploit to an attacker before the affected publisher can issue a security update, and before security software vendors can update their detection signatures. Such a scenario could explain the relative rise in zero-day vulnerabilities seen in recent years.

> Exploits that first appear more than 30 days after a security update are rare.

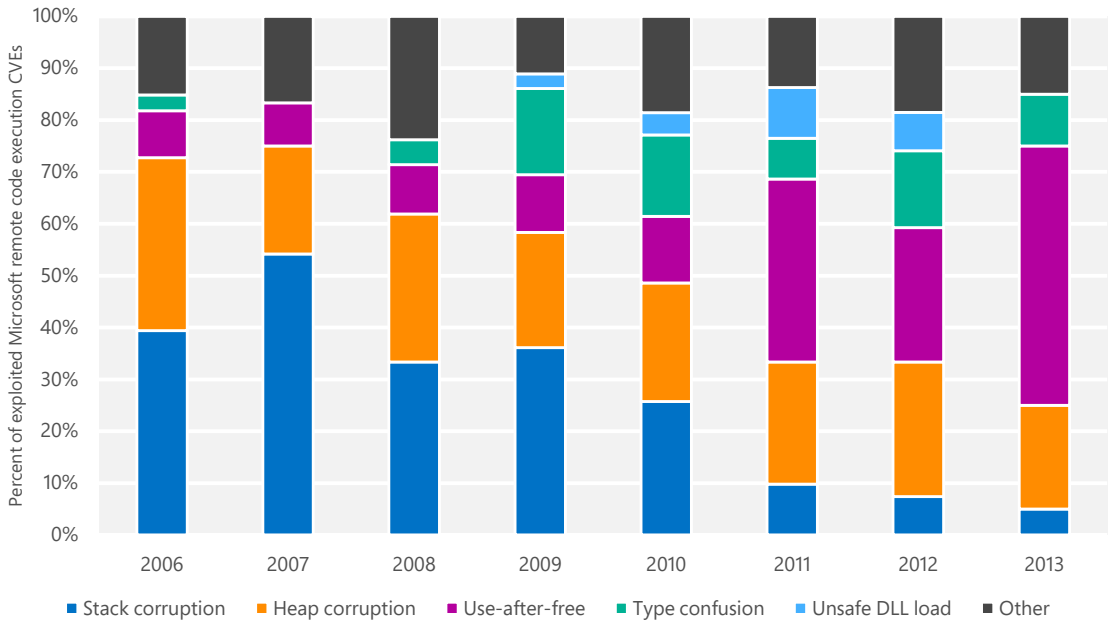
By contrast, exploits that first appear more than 30 days after security update publication have become rare, with only one such instance in 2013. Microsoft has worked with customers to make it easier for them to test and deploy updates quickly after release, even in large organizations. As the share of

computers receiving updates with the first month of release continues to increase, exploiting older vulnerabilities becomes less profitable for attackers, which provides an incentive for them to focus their attentions elsewhere.

How vulnerabilities are exploited

The root cause of a vulnerability plays a key role in defining the set of exploitation techniques that an attacker can use when developing an exploit. As a result, the level of difficulty in developing an exploit is heavily dependent on the type of vulnerability that is being exploited. In terms of risk management, the root cause of a vulnerability can be an important factor in influencing the likelihood that an exploit will be developed. As Figure 3 illustrates, there have been some noteworthy shifts in the classes of vulnerabilities that are known to have been exploited.

Figure 3. The root causes of exploited Microsoft remote code execution CVEs, by year of security bulletin



The first clear shift can be seen in the declining percentage of exploits for stack corruption vulnerabilities, such as stack-based buffer overflows, which accounted for 54.2 percent of known exploited Microsoft remote code execution CVEs in 2007 but accounted for just 5.0 percent in 2013. This vulnerability class has historically been the most likely to be exploited, but has declined considerably since its 2007 peak. Two factors that could be contributing to this decline are the increasing prevalence of exploit mitigations

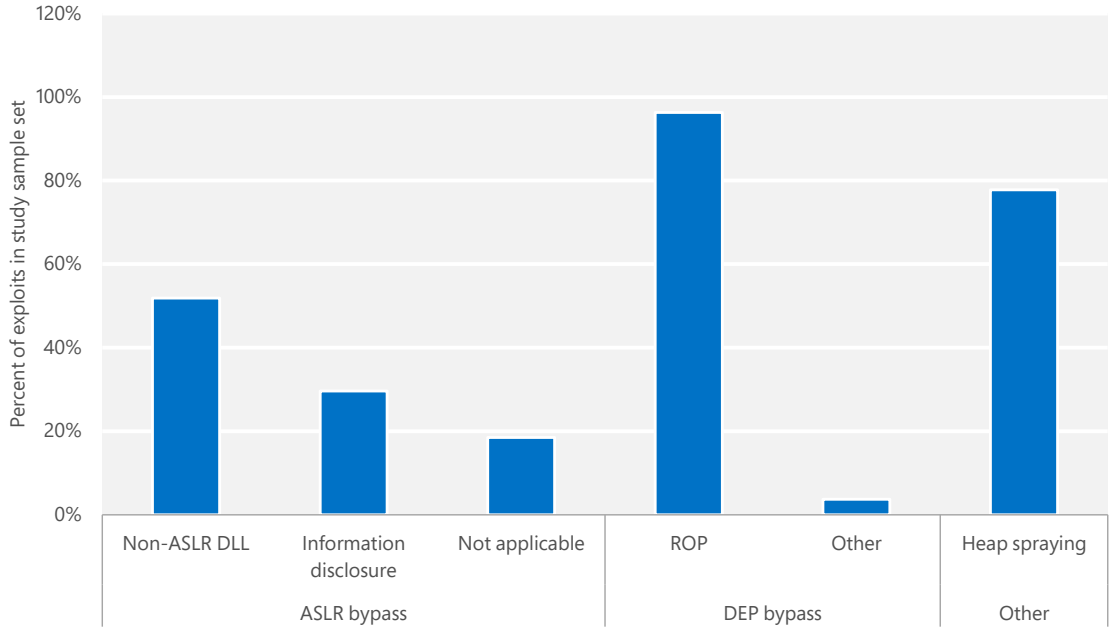
for stack corruption issues (such as /GS and SafeSEH) and the increasing effectiveness of static analysis tools designed to detect such vulnerabilities.[1]

A second shift can be seen in the increasing number of use-after-free vulnerabilities that have been exploited. This vulnerability class includes issues that arise because of incorrect management of object lifetimes. One reason for this increase is that client-side vulnerabilities have become a prime focus for attackers, and object lifetime issues are a common vulnerability class encountered in applications. Exploits that involve unsafe dynamic-link libraries (DLLs) were seen in a small percentage of cases from 2009 to 2012, but not in 2013.

> Stack corruption exploits have declined, and use-after-free exploits have increased.

The introduction of technologies such as Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR) has also affected the way attackers attempt to exploit vulnerabilities. Figure 4 shows the techniques used in exploits targeting vulnerabilities in Microsoft products that were discovered over the past two years.

Figure 4. Techniques used by exploits targeting Microsoft products, January 2012–February 2014



[1] See www.microsoft.com/sdl for information and guidance about using the Security Development Lifecycle to develop secure software.

As this data suggests, the increasing prevalence of DEP and ASLR has forced attackers to identify new techniques that can be used to exploit vulnerabilities even when these features are enabled. An increasing number of exploits attempt to bypass ASLR by relying on images that have not opted into ASLR or by taking advantage of a vulnerability to disclose information about the layout of an application's address space. (Customers can reduce the risk they face from these bypass techniques by deploying the latest version of the Enhanced Mitigation Experience Toolkit (EMET), which can be used to block exploits that use the ROP technique.)

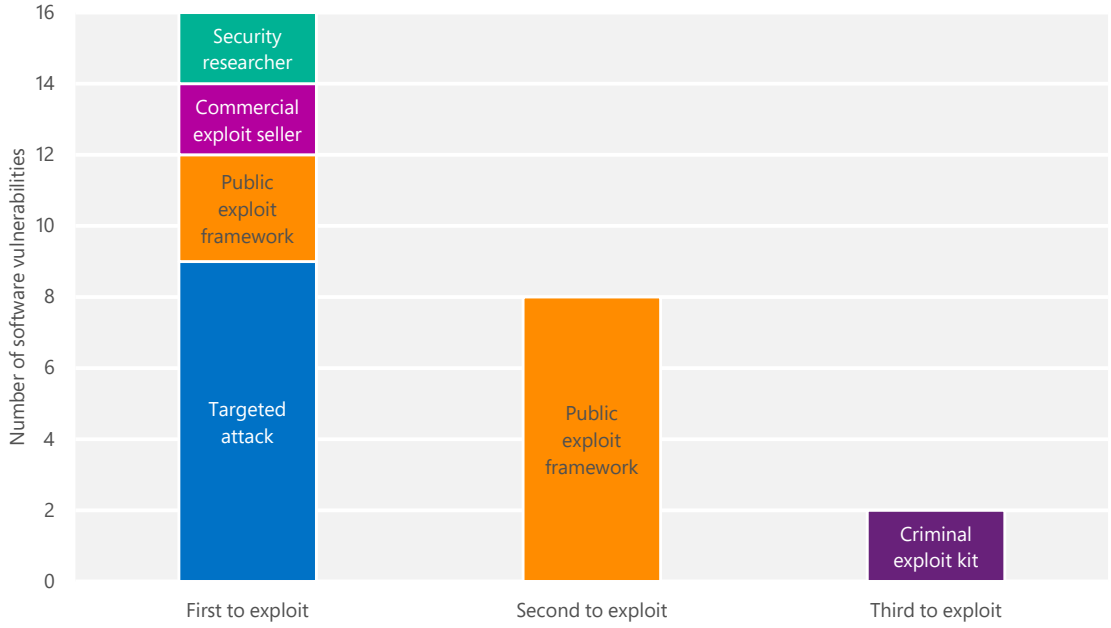
DEP and ASLR have forced attackers to find new techniques.

Having to bypass DEP and ASLR makes developing exploits more difficult and expensive, which has likely been a major factor in the declining trend of new exploits discovered over the past several years. Increased adoption of recent versions of Internet Explorer and EMET should help contribute to this trend, as developing effective exploits becomes even more difficult.

Who exploits vulnerabilities

The parties that initially disclose vulnerabilities are not always the same parties that go on to develop and use exploits that take advantage of them. Vulnerability disclosures originate from a variety of sources, from the dangerous (such as malicious exploit developers and vulnerability sellers) to the beneficial (such as the affected software vendors themselves and security researchers who are committed to coordinated vulnerability disclosure). To explore how exploits make their way into criminal hands, Microsoft analyzed exploits targeting the 16 vulnerabilities in various software products that had known exploits discovered between January 2012 and February 2014.

Figure 5. The first, second, and third parties responsible for known exploits of the 16 software vulnerabilities studied, discovered between January 2012 and February 2014



Of these 16 vulnerabilities, nine were initially exploited in *targeted attacks* against specific targets. In these attacks, often called *advanced persistent threats* or *targeted attacks by determined adversaries*, the attacker concentrates on compromising a single designated target by using a variety of technical and social engineering techniques as necessary. Such attackers are often able to draw upon considerable technological and financial resources, which can include obtaining exclusive access to information about previously undisclosed vulnerabilities that the target is unlikely to have mitigated.[2] Of the remaining exploits, three were first released via public exploit framework, two were released through commercial sellers, and two were released by security researchers.

> Most exploits were first used in targeted attacks that affected relatively few people.

Eight of the exploits subsequently showed up in public exploit frameworks. A public exploit framework is a tool designed to help test computer systems for vulnerability to a variety of exploits. Two of these exploits then appeared in criminal exploit kits.

[2] For more information about targeted attacks, see the paper "Determined Adversaries and Targeted Attacks," available from the Microsoft Download Center, and the post "Targeted Attacks Video Series" (June 13, 2013) on the Microsoft Security Blog at blogs.technet.com/security.

Although the small sample size makes generalization difficult, these findings may be considered to lend additional support to the proposition that installing security updates quickly is one of the best ways to mitigate the risk from exploits. Most of the analyzed exploits were first used in targeted attacks that affected relatively few people. Criminal exploit kits affect a much larger number of people, but the only two exploits to be used in exploit kits were added to the kits several months after security updates that addressed the vulnerabilities were published and widely distributed.

The rise of exploit kits

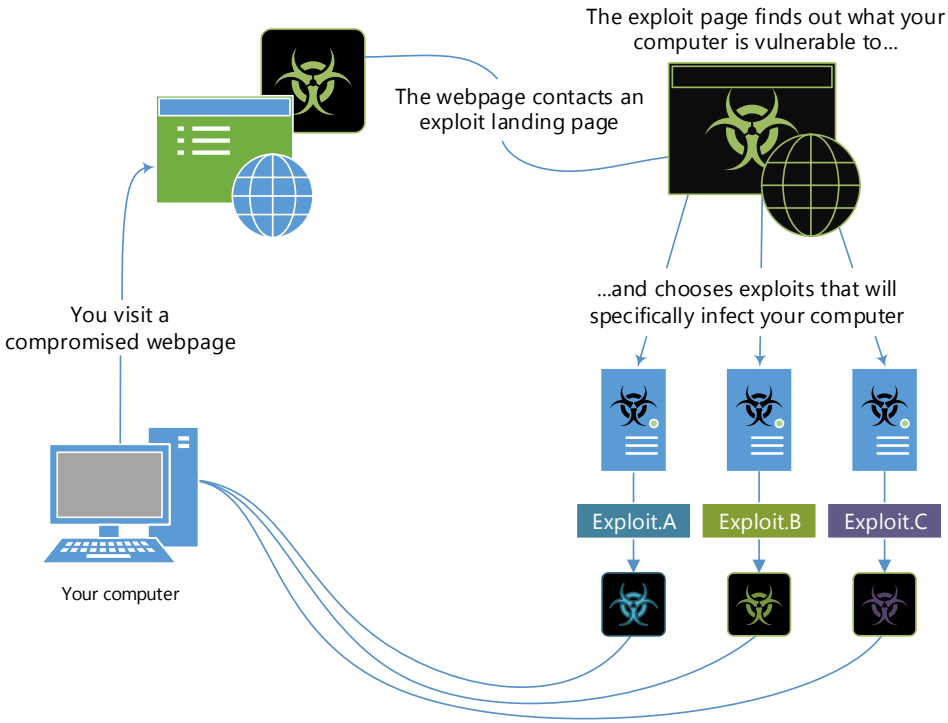
In addition to one-on-one transactions in which buyers purchase exclusive access to exploits, exploits are also monetized through *exploit kits*—collections of exploits bundled together and sold as commercial software or as a service.

Prospective attackers buy or rent exploit kits on malicious hacker forums and through other illegitimate outlets. A typical kit contains a collection of web pages that contain exploits for several vulnerabilities in popular web browsers and browser add-ons, as shown in Figure 6. 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 infection through drive-by download attacks. (See page 98 in the full report for more information about drive-by download attacks.) Commercial exploit kits have existed since at least 2006 in various forms, but early versions required a considerable amount of technical expertise to use, which limited their appeal among prospective attackers. This requirement changed in 2010 with the initial release of the Blackhole exploit kit, which was designed to be usable by novice attackers with limited technical skills—in short, anyone who wanted to be a cybercriminal and could afford to pay for the kit. The potential profits that can be gained by using exploit kits to distribute malware can be considerable: the criminal group behind the malware family Win32/Reveton was reportedly making $50,000 USD per day in 2012 through Reveton installations delivered by exploit kits.[3] (See the "Ransomware" section in the full report for more information about Reveton and similar threats.)

> The potential for illegitimate profit from exploit kits can be considerable.

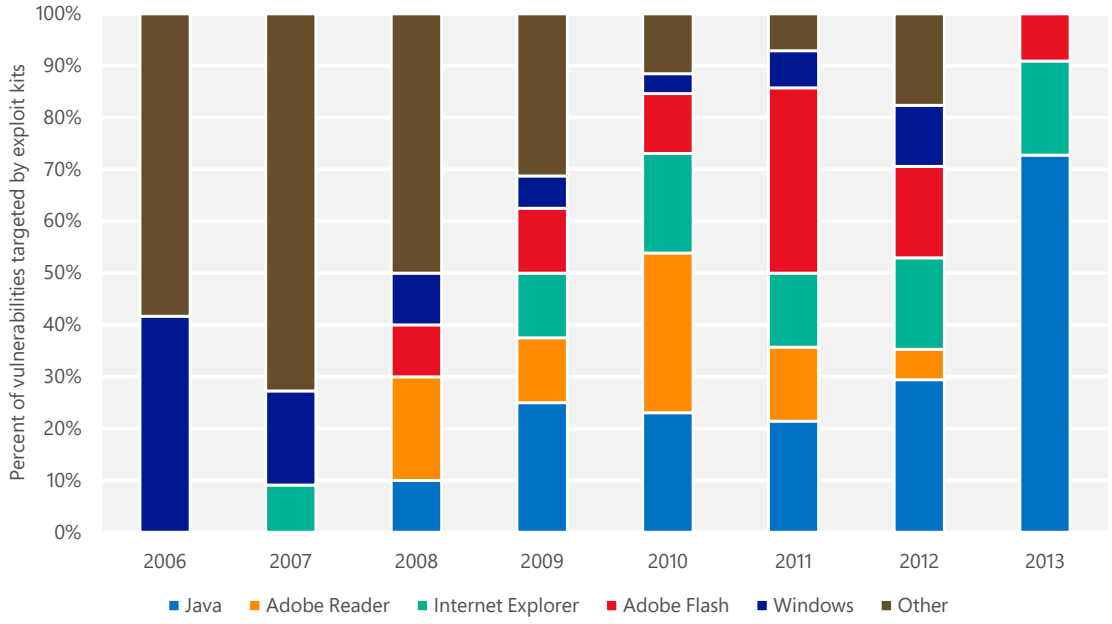
[3] Brian Krebs, "Inside a 'Reveton' Ransomware Operation," *Krebs on Security*, August 13, 2012, http://krebsonsecurity.com/2012/08/inside-a-reveton-ransomware-operation/.

Figure 6. How the Blackhole exploit kit works



Exploit kits are commercial products, if illegitimate ones, and many kits offer highly polished user interfaces and advanced feature sets. Several well-known kits provide attackers with in-depth analytics that can help them plan more effective attacks. The administration screen for the Blackhole kit is similar to a web analytics package, showing where the kit's victims came from, the browsers and operating systems they were using, how many were successfully infected, and how they were infected. Like legitimate commercial software, exploit kits often include license agreements and may come with support contracts.

Figure 7. Exploit kit exploits targeting vulnerabilities in different products, 2006–2013



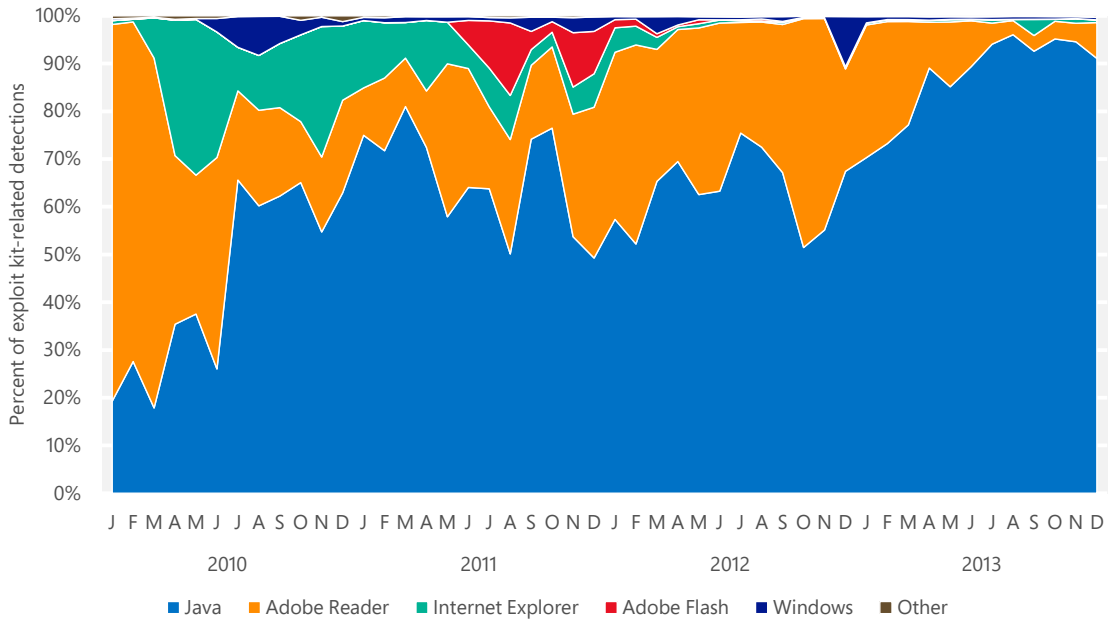
Data taken from the Contagio Exploit Pack Table, http://contagiodump.blogspot.com/2010/06/overview-of-exploit-packs-update.html.

Exploit kit makers continually update the set of exploits included in their kits, adding new exploits as they are discovered and discarding old exploits that are no longer effective or are considered too likely to be detected by security software. Early exploit kits targeted vulnerabilities in a diverse set of products from several different vendors. Over the years, kit makers have gradually narrowed down the list of products they target to a handful of widely deployed products and components, notably Adobe Flash and Reader, Microsoft Windows and Internet Explorer, and Oracle Java. Recently, kit makers have increasingly focused on vulnerabilities in out-of-date versions of the Java Runtime Environment (JRE), which is often installed on desktop and laptop computers as a browser add-on. In 2013, nearly three-quarters of the exploits used by exploit kits targeted JRE vulnerabilities.

As Figure 8 shows, the trend toward JRE vulnerabilities becomes even more pronounced when actual exploit detections are considered.[4]

[4] Figure 8 and Figure 9 examine computers with detections of exploits that are known to be targeted by exploit kits. Detections for CVEs that are not known to be exploited by exploit kits are not included in these charts, nor are detections that cannot be associated with a specific CVE. Computer totals are expressed as percentages of

Figure 8. Exploit kit-related malware detections, 2010–2013, by product or component targeted



Although exploit kit makers continue to include exploits for a variety of programs and components, not all of the exploits get exposed to every computer that visits a malicious web page. To reduce their chances of detection by security software, many exploit kits include code that allows them to expose only a subset of the vulnerabilities in the kit based on the characteristics of the visiting computer, or on which exploits have been the most successful in the past. Over the past few years, exploit kit-related detections have become increasingly dominated by JRE exploits. In 2013, JRE exploits accounted for between 84.6 and 98.5 percent of exploit kit-related detections each month, with Adobe Reader exploits a distant second. Exploits targeting all other products, including Internet Explorer, accounted for just 1.1 percent of detections each month in 2013 on average.

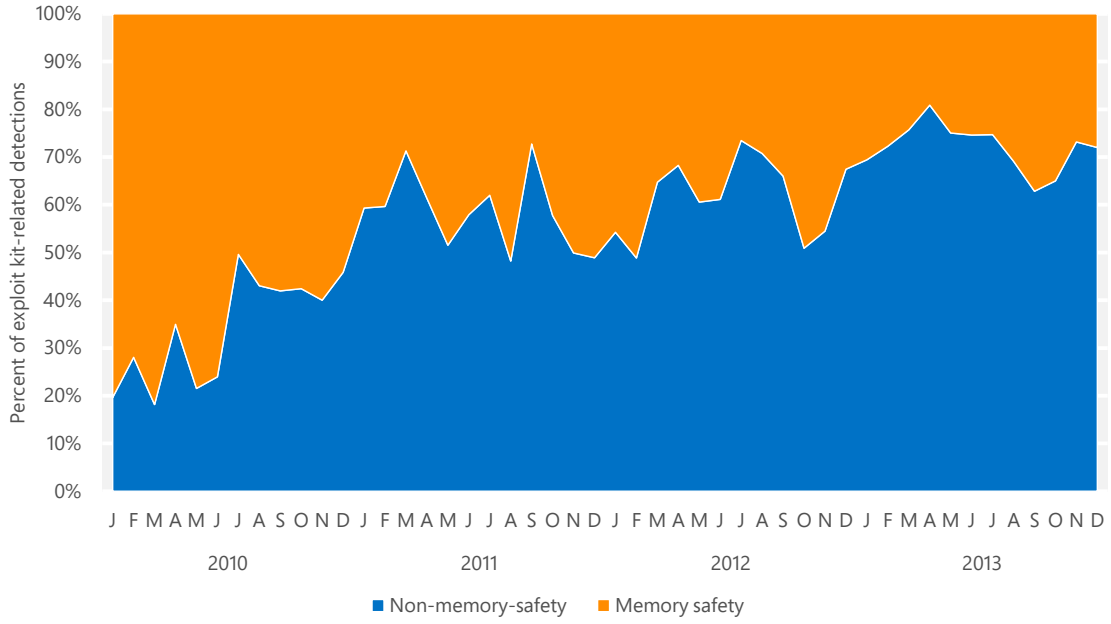
> Memory safety issues have become harder to exploit because of mitigations like ASLR and DEP.

Technologies such as DEP and ASLR are a likely factor in exploit kit authors' increasing preference for exploits that don't involve memory safety, as shown in Figure 9.

---

computers that encountered the aforementioned exploits, not as percentages of all reporting computers. See "Exploits" in the full report for a more comprehensive look at exploits and related threats.

Figure 9. Exploit kit-related malware detections, 2010–2013, by type of vulnerability



Memory safety issues, which as recently as 2010 accounted for a clear majority of malware detections from exploit kits, have become harder to reliably exploit because of mitigations such as ASLR and DEP. Consequently, memory safety exploits have become less popular among kit authors than other exploit techniques.

Guidance: Staying ahead of exploits

The likelihood that a vulnerability will be successfully exploited depends on many factors, including the type of vulnerability being exploited, the product versions being targeted, an attacker's ability to make use of the necessary exploitation techniques, and the amount of time required to build a reliable exploit. The following actions can help organizations and individuals significantly reduce the risk they face from exploits.

Stay current on security updates

Most of the examined vulnerabilities only showed signs of being exploited after a security update had been made available. Exploit kits, in particular, tended to target vulnerabilities for which security updates had already been available for a significant amount of time. Installing security updates as soon as they are available can help minimize risk.

Use the newest versions of applications

Windows 8.1, Internet Explorer 11, and Office 2013 all take advantage of improved security features that more effectively mitigate techniques that are currently being used to exploit vulnerabilities. Deploying these product versions widely can mitigate the risk an organization faces from several of the most commonly detected exploits. Using the 64-bit edition of Internet Explorer 11 with Enhanced Protected Mode enabled can also help protect users from a range of Internet-borne threats.

Use the Enhanced Mitigation Experience Toolkit (EMET)

EMET can be used to protect applications that run on all supported versions of Windows. The features included in EMET are specifically designed to break exploitation techniques that are currently used by attackers. See "Enhanced Mitigation Experience Toolkit" in the full report for more information about EMET and how it can be used to reduce risk.

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