



VERISIGN™



WHITE PAPER

AN EXCERPT FROM 2012 iDEFENSE CYBER THREATS AND TRENDS

VERISIGN® iDEFENSE SECURITY
INTELLIGENCE SERVICES

CONTENTS

PURPOSE AND SCOPE	3
KEY FINDINGS	3
MALICIOUS CODE TRENDS	4
New Breeds of Malware: Zeus Source Code as an Enabler	4
Examples	4
Looking Ahead	6
Use of Free Domain Providers for Malicious Activity Spikes in 2011	6
Advancements in Web-Malware Evasion	7
VULNERABILITY TRENDS	9
Vulnerability Analysis	9
New Vulnerability (v1) Trends	10
Overall Vulnerability Trends	11
Top-10 Vendors in 2011	12
Increasing Sophistication of Exploits	12
Reducing Exploits through Sandboxing Technology	13
Chrome Browser Adoption to Surpass Firefox in 2012	15
Vendor Bounty Programs in 2011	16
CONCLUSION	19
ABOUT VERISIGN iDEFENSE SECURITY INTELLIGENCE SERVICES	19
LEARN MORE	19
ABOUT VERISIGN	19

THE END OF THE YEAR IS AN OPPORTUNE TIME TO TAKE A STRATEGIC LOOK AT THE CYBER SECURITY LANDSCAPE AND CONSIDER WHAT ADJUSTMENTS ENTERPRISES NEED TO MAKE TO BETTER ANTICIPATE AND MANAGE THREATS. REMOVED, IF ONLY TEMPORARILY, FROM THE TACTICAL DAILY MANAGEMENT OF CYBER SECURITY ISSUES, IN THIS REPORT, IDEFENSE PAUSES TO SURVEY THE PAST 12 MONTHS AND TO REFLECT ON HOW THE FIELD OF CYBER SECURITY IS TAKING SHAPE.

In this tradition, each year, with its cyber threats and trends report, iDefense attempts to shed light on the salient issues of the previous year. In 2008, iDefense assessed the emergence of cyber espionage and cyber criminal cartels. In 2009, governments were becoming the most influential participants in the global cyber threat environment, which iDefense measured by the urgency of threats that governments perceived, increased cyber security budgets and the designations of cyber infrastructures as national security assets. The year 2010 was the year of Aurora and Stuxnet, which signified that every enterprise was a potential target.

PURPOSE AND SCOPE

The following is an excerpt of the Verisign iDefense 2012 Cyber Threats and Trends report. The full report is sent to Verisign iDefense customers annually, providing a valuable overview of key cyber security trends during 2011 and how those trends and others might unfold in 2012. iDefense intends for this report to serve as a reference and a strategic complement to daily tactical intelligence reports for the purpose of providing IT security and business operations with actionable and relevant decision support. The objective of this report is to effectively inform IT security and business operations teams of potential threats; to allow those teams to anticipate key cyber security developments for the coming year; and to provide, where appropriate, solutions to help reduce organizational risk related to cyber security. This report uses iDefense intelligence-collection research and analysis, and research using both primary and secondary sources.

KEY FINDINGS

This year's Verisign iDefense Security Intelligence Services report, "2012 Cyber Threats and Trends" produced the following key findings as part of the team's research into significant cyber security trends during 2011. These key findings relate to trends in cyber crime, malware, vulnerabilities and exploits.



It is no surprise then that with the release of the Zeus source code comes a litany of Zeus-based variants. Given the sophisticated nature of Zeus and its source code, this is a trend that will continue into 2012 and beyond.

Zeus Banking Trojan Becomes an Open-Source Crime Kit

The release of the Zeus source code effectively converted the Zeus banking Trojan from a proprietary, pay-per-use crime kit into an open-source crime kit. The source code quickly spread across the Internet via underground websites and file-sharing sites, giving malware authors across the globe access to a powerful and well-written malware platform.

New Criminal Business Model Emerges: Malware-as-a-Service (MaaS)

Cyber criminals are starting to shift to a business model known as malware-as-a-service (MaaS), where authors of exploit kits offer extra services to customers in addition to the exploit kit itself. This trend will probably continue as other developers adopt the same business model.

Use of Sandboxes Significantly Increases Cost and Complexity of Exploit Development

The application of sandboxes has made exploiting vulnerabilities significantly more difficult. Currently, only two public demonstrations of bypassing sandboxes exist in environments that use and support defense-in-depth strategies such as address layout randomization (ASLR) and data execution prevention (DEP). None of the public demonstrations included any public exploit code. Until corporate enterprises widely adopt newer client-side applications that have implemented sandboxes, however, attackers will have an easier time developing exploits.

MALICIOUS CODE TRENDS

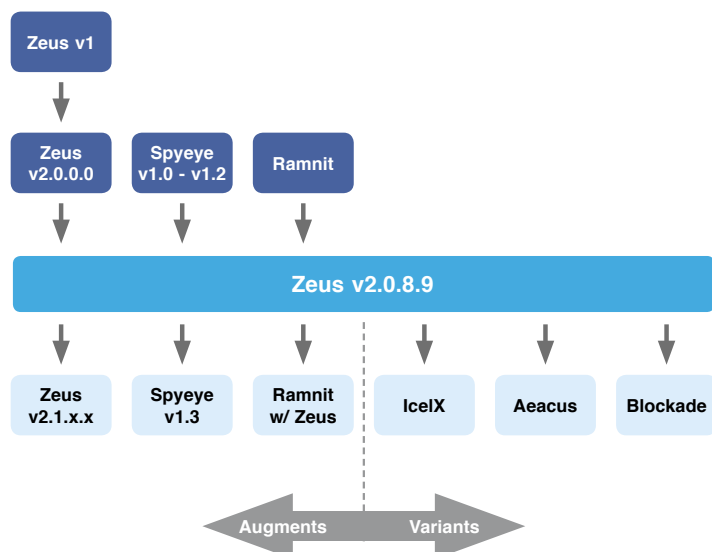
New Breeds of Malware: Zeus Source Code as an Enabler

In April 2011, the source code for Zeus version 2.0.8.9 became available online. The release of the Zeus source code effectively converted the Zeus banking Trojan from a proprietary, pay-per-use crime kit into an open-source crime kit. The source code quickly spread across the Internet via underground websites and file-sharing sites, giving malware authors across the globe access to the powerful and well-written malware platform. Fortunately, the Zeus source code is incomplete and does not compile without modification and additions. This means that those wishing to use the Zeus source must have the necessary programming skills to overcome the deficiencies in the code base. This prevents less-experienced from attempting to use the source code but at the same time forces malicious actors to modify the source, thereby branching the source code into variants.

Examples

There are several known variants currently in the wild that are in part or entirely based on the Zeus source code. Some of the variants augment the Zeus framework while others steal pieces of the Zeus infrastructure for use in completely different code bases. In either case, the fact remains that these variants are leveraging some aspect of the Zeus system to proliferate in an already congested malware environment. Exhibit 1 provides a graphic summary of the Zeus code's influence on malware.

EXHIBIT 1: VARIANTS OF ZEUS



- **Spyeye:** Originally, Spyeye was a direct Zeus competitor. The initial versions of Spyeye, when infecting a new victim, would locate and uninstall any existing Zeus infections.¹ In late 2010, the author of Zeus, who uses the handles “Slavik” and “Monstr,” announced that he or she would be retiring and that he or she would transfer the source code of Zeus to the author of Spyeye, who uses the handles “Gribodemon” and “Harderman.” In January 2011, the first Spyeye and Zeus hybrid appeared in the wild.² The Spyeye and Zeus hybrid increased Spyeye’s original capabilities by enhancing Spyeye with features such as Zeus’s HTML injection functionality that allows man-in-the-browser (MITB) attacks.
- **Ramnit:** Ramnit is a worm that first appeared in 2010. The Ramnit worm began life as a basic file infector. Its unique feature was that not only did it infect .exe and .dll files, but it also infected .html files to propagate.³ The worm consists of multiple components that provided various additional features. After the release of the Zeus source code, one of the additional features that Ramnit began including in infections was an HTML injection engine.⁴ This engine was a direct derivative of the Zeus HTML injection engine. The configuration file for the Ramnit HTML injection engine was a direct knockoff of Zeus’s Web inject configuration format.
- **Ice IX:** Ice IX (known as Ice 9) is a direct Zeus variant. Unlike Ramnit and Spyeye, which their authors augmented using pieces of the Zeus source code, the authors of Ice IX merely modified and completed the existing source code to produce a Zeus clone. Ice IX does not offer any new features related to data theft but instead focuses on attempting to thwart trackers, such as the Zeus Tracker website, abuse.ch. To do this, Ice IX uses a weak encryption system that a tracker must implement to access the configuration file from the command-and-control (C&C) server. This protection scheme has already failed, as abuse.ch reported on Aug. 25, 2011.⁵

Attackers typically use popular free domains for all kinds of malicious activities, including malware hosting, C&C servers, exploit kits, spamming, phishing and even selling fake anti-virus products.

- **Aeacus:** Like Ice IX, Aeacus is a clone of Zeus that the authors based directly on the Zeus source code. The data theft functionality of Aeacus is identical to that within the Zeus source code. What makes Aeacus notable is the fact that the authors of Aeacus implemented a novel peer-to-peer (P2P) communication infrastructure for updating both the configuration files and the executable. In addition to the P2P communication network, Aeacus' authors modified the underlying encryption subsystem of Zeus to allow the possibility of encryption systems other than the standard RC4 algorithm.

Looking Ahead

With the Zeus source code freely available and nearly complete, it is a safe bet that many more variants will appear. As 2011 has demonstrated, with the de facto banking Trojan's source code in an open-source format, many new malicious actors will capitalize on such a robust system to elicit financial gains (either through the use of the modified Trojans or the sale of the modified Trojans). As Ramnit and Spyeye demonstrate, there will be more minor Trojans that include the functionality of Zeus into their arsenals. This trend will be even more pronounced when new malware families emerge that not only augment themselves with components of Zeus but also augment Zeus with new functionality specific to each new variant family. The release of the Zeus source code is going to have a dramatic impact on the production of new, dangerous banking Trojans in 2012. Fortunately, anti-virus programs may actually detect as Zeus the malware variants that malware authors have based on Zeus' source code—a detection that will decrease the effects of these variants.

Use of Free Domain Providers for Malicious Activity Spikes in 2011

While registering a domain with one of the generic country code top-level domains (TLD) generally has an annual fee associated with it, some TLDs and second-level domains offer free domain registrations. For instance, the domain “.nr” is the country code TLD for the Republic of Nauru. However, the company CO.NR has registered the domain co.nr, which allows the company to offer sub-domains. In this case, the company also offers URL redirection, URL cloaking and masking. Given that these sub-domains have no cost, attackers are drawn to them for hosting malicious content. According to research data⁶ from M86 Security Labs, there has been a 250 percent increase in the malicious use of free domain services in 2011. In many instances, hackers prefer to take advantage of free domains instead of compromising existing websites. Even though some domain-hosting services are not completely free, their low costs still attract malicious actors. These DNS providers also make it possible for attackers to add countless numbers of domain names cheaply. The “co.cc” “registry” can register 15,000 addresses at a time for \$1,000 US, which equates to about \$0.07 per sub-domain name. Exploit toolkits and malware C&C servers often must remain operational for only a few days to make an impact and are frequent targets of takedown by the security community. As such, purchasing a domain to host these malicious resources provides little benefit over the free alternatives.

6 “Security Labs Report January – June 2011 Recap.” Accessed on Oct. 10, 2011. M86 Security. http://www.m86security.com/documents/pdfs/security_labs/m86_security_labs_report_1h2011.pdf.

Advancements in Web-Malware Evasion

As the browser became the platform of choice for most applications, attackers followed the trend and began deploying malware through their targets' browsers rather than their inboxes, as was popular in the early 2000s. Through Google's safe-browsing program that informs the protections that Mozilla's Firefox and its own Chrome browser provide, Google displays around 3 million malicious website warnings each day.⁷ The efforts of Google, Microsoft Corp. and others in the security community to detect these malicious websites and prevent them from harming users have resulted in an arms race. As with traditional anti-virus programs, attackers prefer that their creations go undetected to maximize their infection potential. To extend their longevity, these exploit toolkits use complex obfuscation techniques to evade detection.

One common way to analyze and detect Web-based malware is by using Web honeypots, which use virtual-machine-based systems running full operating systems. These systems visit potentially malicious pages and then scrutinize the result of these visits for suspicious activity. This activity may be the creation of new processes, encryption routines or specific indications of known vulnerabilities. Another common tactic is to use browser emulators that act like a browser but do not actually execute the potentially malicious payload.

Most advances in Web-malware evasion focus on evading emulators. Emulators aim to simulate execution of the malicious Web code to discover what vulnerabilities the page is attempting to exploit. Attackers combat emulators using obfuscation systems that they have designed specifically to detect or confuse these systems. While toolkits have used many obfuscation tactics have in previous years, in 2011, the vast majority of malicious Web pages that iDefense analyzed began to deploy two or more obfuscation techniques, greatly improving their chances of evading detection. To be completely effective, browser emulators must emulate the targeted browser (e.g., Internet Explorer 7) perfectly. Perfect emulation means even minor deviations from the standard and bugs in code parsing must also function as the attacker expects. These types of browser idiosyncrasies often play in the attacker's favor when trying to evade an emulator.

Internet Explorer has the ability to conditionally execute code within HTML comments, depending on what version of the browser parses HTML. The following code demonstrates how a conditional comment can check which version of Internet Explorer the victim is using and only execute JavaScript code if the version is greater than or equal to seven:

```
<!--[if gte IE 7]>
<script>
document.write("Hello Malware");
</script>
<![endif]-->
```

⁷ Ballard, Lucas. "Four Years of Web Malware." Aug. 17, 2011. Google. <http://googleonlinesecurity.blogspot.com/2011/08/four-years-of-web-malware.html>.



Many often refer to both Web honeypots and browser emulators as honeyclients or client honeypots. These two techniques are the main methods researchers use to detect malicious websites and are therefore the most important for attackers to evade.

Internet Explorer also allows for conditional compilation of JavaScript code using the following syntax:

```
/*@cc_on  
document.write("Hello Malware");  
@*/
```

Attackers use these simple tactics to evade detection in emulators that do not incorporate these deviations from the standard. Other evasion techniques involve hiding code within features of the browser that an emulator may not include. As an example, cascading style sheets (CSS) is a typically benign technology that developers commonly use to alter how a browser displays a Web page. The BlackHole exploit toolkit stores data within CSS files, which it then accesses using JavaScript and decodes before injecting into Web pages.

Asynchronous JavaScript and XML (AJAX) calls that Web-based malware uses can also cause problems for emulators. iDefense analyzed a variant of the Phoenix exploit kit in March 2011 that contacted the Twitter application programming interface (API) in the course of its decoding routine. In the case of this attack, the exploit kit executed a function that requested data from the Twitter "Trends" API and used the returned data to determine if it should continue decoding its true payload. If a browser emulator did not make the request to Twitter and incorporate the returned data into the execution process, it would fail to detect the malicious payload.

One place browser emulators may fail to accurately portray a browser is in the handling of document object model (DOM) APIs, such as when missing resources cause errors. Multiple malicious websites that iDefense analyzed in 2011 forced the browser into error conditions to trigger the execution of "onerror" events that decoded the malicious payload. In one case, the toolkit added an "img" tag that attempted to load "about:blank." This tag caused no additional network requests but forced the browser to decode the additional payload; comparatively, an emulator may have simply ignored the fact that about:blank is not a valid image and continue execution.

When combined, these small tactics make it very difficult for an emulator to accurately portray a legitimate browser. As the malware arms race continues, HTML5 features may represent a treasure trove of new locations for attackers to store data and detect browser emulators. Malicious actors could use new multimedia tags, such as <video> and <audio>, to store obfuscated JavaScript code. Attackers could use the Geolocation API to help target their exploits at individuals in more-specific locations or to avoid decoding payloads when a browser does not properly report its location. There are a tremendous number of possibilities available for attackers, and iDefense expects that exploit toolkits will abuse new browser features in 2012.



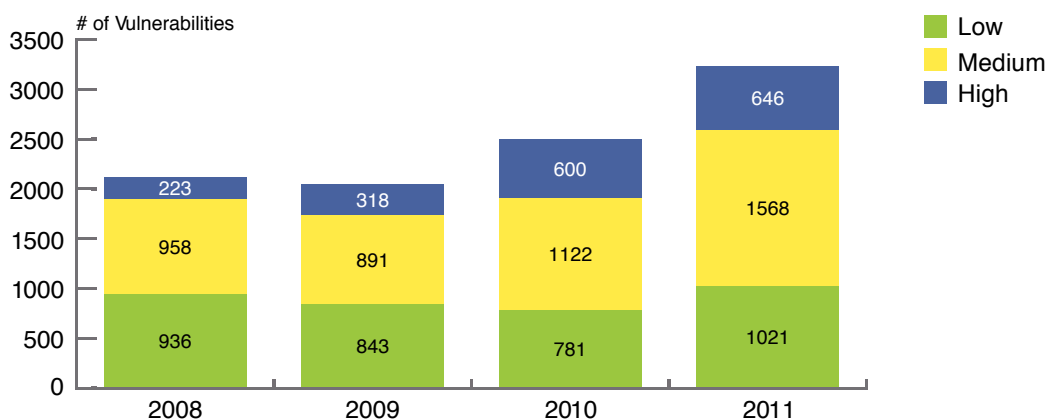
Even though some domain-hosting services are not completely free, their low costs still attract malicious actors.

VULNERABILITY TRENDS

Vulnerability Analysis

The number of new vulnerabilities iDefense reported for the months of January-October 2011 increased by 732 vulnerabilities, or 30 percent, in comparison to the same period in 2010. The increase in the number of vulnerabilities is across all three severity levels (HIGH, MEDIUM and LOW) that iDefense assigns to its vulnerability reports (see Exhibit 2).

EXHIBIT 2: VULNERABILITY COUNT BY SEVERITY FOR THE MONTHS JANUARY THROUGH OCTOBER 2011



The increase in the vulnerability reports across all the three severity levels is a reflection of the increased scrutiny vulnerabilities have been receiving from security analysts. iDefense did not notice any significant change in any vendor's security posture in 2011, nor did any vendor publicly announce a change in its quality assurance policy. Thus, the spike in this year's vulnerability counts appears to be an overall reflection of increasing security awareness.

iDefense monitors all the vulnerabilities within the products of major vendors and a selected set of additional customer-requested vendors. The total number of vendors iDefense covers is roughly 450. iDefense vulnerability trends are based on this set of important vendors and so tend to represent relevant vulnerability trends better than more generic vendor lists used by other security vendors.

iDefense broadly categorizes its vulnerability reports as the following:

- Version 1 vulnerability reports (v1)
- Updated vulnerability reports



There are a tremendous number of possibilities available for attackers, and iDefense expects that exploit toolkits will abuse new browser features in 2012.

iDefense based this classification on whether the vulnerability report is new (in its first version) or is an old vulnerability (any report with a version number greater than one). Both of these types of vulnerability reports have their own unique significance in vulnerability trending charts. A v1 vulnerability implies that this vulnerability was not publically known, and the date of the v1 report is when the vulnerability became publicly known. An update (non-v1 report) implies that the vulnerability is already publically known and perhaps one vendor has already patched it.

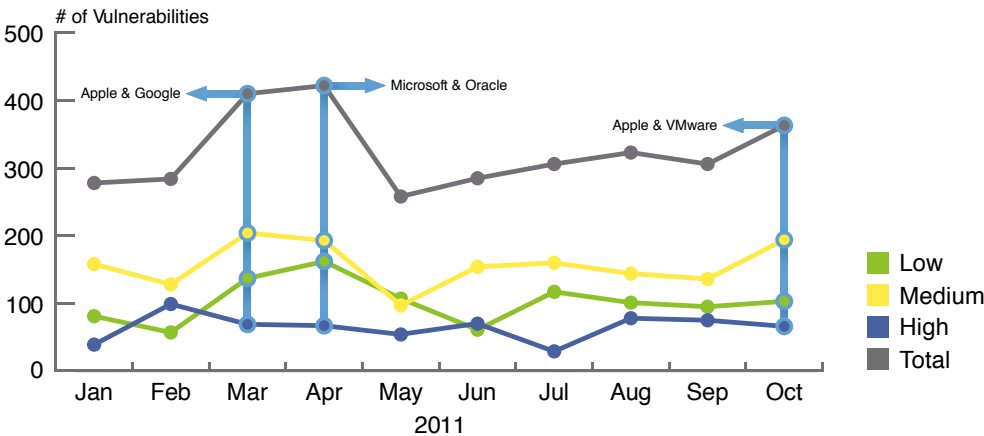
New Vulnerability (v1) Trends

By aggregating the v1 vulnerability counts by months, as Exhibit 3 shows, it is evident that the largest spike in new vulnerability counts were during the months of March and April.

The two major vendors for whom iDefense wrote the greatest number of vulnerability reports in March were Apple Inc. and Google. Comparing this information with the totals for the first half of 2010, Apple's big release in March comes as no surprise. Apple also released patches for an unusually large number of vulnerabilities in March 2010. Although Apple does not have a fixed patching cycle, this release of a large number of patches in March seems to be a trend. Microsoft and Oracle Corp. released an abnormally large number of patches in April. Thus, this is the reason for the spike in the trend lines that Exhibit 3 displays. Google, of course, is a relatively new addition to the list of vendors that release a large number of patches.

There is no significance to the release of an unusually higher number of vulnerability patches by vendors in March and April. The month of March generally sees a large number of vulnerability patches due to the confluence of patching cycles of major vendors.

EXHIBIT 3: MONTHLY AGGREGATES



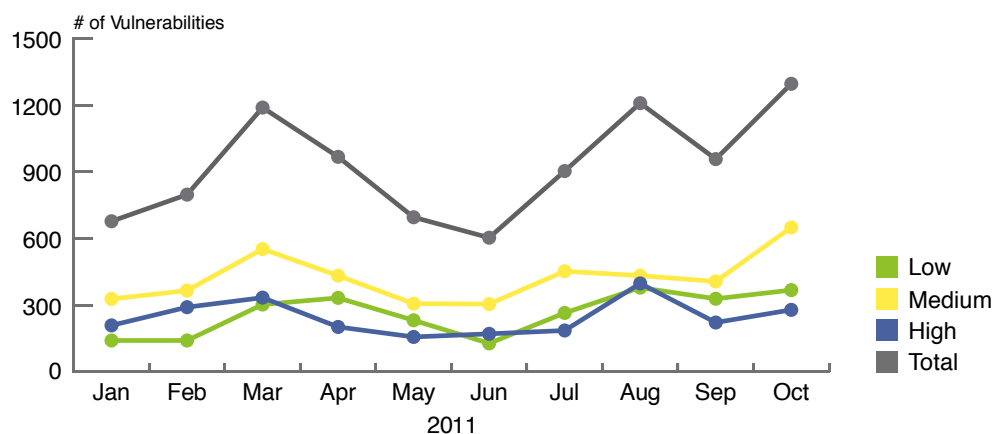
In October, the v1 report count spiked again but did not reach the same volumes as that of March and April. Apple, VMware and Oracle were the vendors who released large numbers of patches in October. The v1 vulnerability count for October was not the highest for this year, as Apple and VMware also released patches for existing known vulnerabilities.

Overall Vulnerability Trends

Reviewing all of the vulnerability reports that iDefense has written in 2011 (v1 and updates) offers additional insights into vulnerability trends. The month of October had the greatest total number of vulnerabilities, as Exhibit 4 shows:

Apple, VMware, Microsoft and Oracle released patches in October. The combined total of patches from just these vendors was approximately 300 vulnerabilities.

EXHIBIT 4: TOTAL VULNERABILITY REPORT COUNTS FOR 2011



In its report “Vulnerability Events and Trends of H1 2011,” published earlier this year,⁸ iDefense mentioned that there are a few months of the year wherein more than one scheduled vendor patch release coincides. Thus, it should not be surprising that the number of patches for such months is high. October is one such month.

Apple and VMware do not follow regular patching cycles, and the high number of patches from these two vendors caught everyone by surprise. There are no explanations for the release of this large volume of patches from Apple and VMware. Both companies use many open-source products and thus have to patch a large number of vulnerabilities within third-party products. Since neither of these companies maintain scheduled patch releases, it is not possible to predict when they will release patches.

⁸ “iDefense Vulnerability Events and Trends of H1 2011” (Aug. 24, 2011).

Although Apple does not follow a patch release schedule, for the past few years, Apple has displayed a trend of releasing a large number of patches in March, June and October. Given this trend, it is safe to assume that Apple will also continue to release a larger number of patches in March, June and October in the next few years.

Top-10 Vendors in 2011

Exhibit 5 displays the top-10 proprietary source vendors in terms of vulnerability count covering the period January through October 2011.

EXHIBIT 5: TOP-10 PROPRIETARY SOURCE VENDORS

2007	2008	2009	2010	2011
Novell	Novell	Oracle	Apple	Oracle
Avaya	Apple	Apple	Oracle	Apple
Microsoft	Sun	Novell	VMware	Novell
Sun	Avaya	Avaya	Novell	Microsoft
Apple	Hewlett-Packard	Microsoft	Hewlett-Packard	Google
Oracle	IBM	Hewlett-Packard	Microsoft	VMware
Silicon Graphics	Microsoft	Nortel Networks	Adobe	Adobe
Hewlett-Packard	Oracle	Mozilla	Google	Hewlett-Packard
IBM	Cisco	IBM	Cisco	Cisco
PHP Group	Vmware	Cisco	Mozilla	IBM

Apple relinquished its top position to Oracle for 2011. Since Oracle has taken over Sun Microsystems, the number of vulnerabilities it has to patch increased exponentially after the acquisition. This effectively ensures Oracle will remain in the top-5 vendor list for more years to come. Google and Adobe, which both entered the top-10 list for the first time in 2010, continued to remain on the list. Conspicuous in its absence is the Mozilla Foundation, which dropped out of the top-10 list for this year.

Looking ahead to 2012, Oracle, Microsoft and Apple will most likely continue to remain within the top-5 proprietary source vendors with regard to the number of released patches. The months of March and October will continue to remain the highest patch release months due to the confluence of vendors' scheduled patch releases.

Increasing Sophistication of Exploits

As the "Vulnerability Events and Trends of H1 2011" report references,⁹ the cat-and-mouse game for software vulnerabilities has not changed in the sense that software vendors continue to make security improvements for their products while security researchers, benign and malicious, continue to find ways to defeat or bypass the security improvements that vendors have implemented; however, software vulnerabilities continue to grow in complexity, which often leads to complex exploits.

⁹ "iDefense Vulnerability Events and Trends of H1 2011" (Aug. 24, 2011).

The typical vulnerability that attackers exploit today is much more complicated than exploited vulnerabilities from only 5 years ago. Because of this increased complexity, vulnerability and exploit discovery has proven more difficult when using modern static-analysis tools and manual analysis of the source or binary code. Observers attribute this increased difficulty to software vendors' improvement of their coding practices and their use of complex coding methods.

Defense-in-depth strategies such as ASLR and DEP evolved more than 4 years ago; however, the broad adoption of defense-in-depth strategies by corporate enterprises has forced attackers to employ more sophisticated methods to bypass the available protections. Consider for instance the Adobe Flash Integer Overflow Vulnerability,¹⁰ which malicious actors used in targeted attacks against the defense industrial base. The exploit that attackers used for this vulnerability

was the first exploit that bypassed both ASLR and DEP but did not rely on the location of a poorly configured dynamic-link library (DLL) at a fixed address. Instead, the exploit used one vulnerability to leak information and trigger memory corruption. A more common practice for exploiting software that utilizes these exploit protections requires an attacker to use multiple vulnerabilities to achieve arbitrary code execution on a vulnerable system.

Looking ahead, many of the popular client-side applications (e.g., Adobe Reader X and Office 2010) have implemented sandboxes, which will pose similar challenges for attackers to exploit, as ASLR and DEP did when corporate enterprises initially introduced them. iDefense is aware of one instance of an exploit bypassing ASLR, DEP and a sandbox, which occurred during the 2011 Pwn2Own contest; however, until corporate enterprises widely adopt these newer client-side applications, attackers need not be overly concerned with circumventing this exploitation protection.

In short, modern exploits require greater sophistication and thus greater resources (i.e., time, money, creativity) to be successful. This increased resource requirement on the part of attackers relates directly to both the complexity of vulnerabilities and the adoption of exploit protections.

Reducing Exploits through Sandboxing Technology

The use of sandbox technologies has significantly hindered the ability of malicious actors to exploit vulnerabilities. Consequently, software vendors will continue to use sandbox technologies to help protect their products and customers. Sandbox technology is a mitigating security mechanism that limits the environment in which a program can execute. Companies typically use sandboxes to process untrusted content while keeping a host system protected from persistent changes. Sandboxes do not eliminate vulnerabilities but rather make exploiting vulnerabilities much more difficult. Oftentimes, an attacker must exploit multiple vulnerabilities together to exploit a vulnerability in software that uses sandbox technology.

This focus of software vendors to increase the complexity and cost of exploit development through the application of sandboxed environments has been an effective approach, as no public exploit code currently exists for popular sandboxed applications.

¹⁰ "CVE 2011-2110. See Adobe Security Advisory APSB11-18: "Security update available for Adobe Flash Player." June 14, 2011. <http://www.adobe.com/support/security/bulletins/apsb11-18.html>."

The concept of sandboxes is not new, but the application of sandboxing by many software vendors is relatively new. In 2007, Microsoft first introduced the concept of sandboxing for the modern browser in Internet Explorer 7 with Protected Mode. Google reacted the following year with a sandboxed Web browser, Chrome. In 2010, Adobe, with the help of Google and Microsoft, soon after released Protected Mode for Adobe Reader X. Microsoft continued to grow its sandbox technology through its product line by introducing a sandbox technology for its Office products in Protected View Mode for Office 2010. Similarly, in 2011, Adobe used its experience and knowledge from Protected Mode for Adobe Reader X to introduce a sandbox for Adobe Acrobat, which Adobe dubbed Protected View Mode.¹¹

The application of sandboxes has made exploiting vulnerabilities much more difficult. Currently, only two public demonstrations of bypassing sandboxes exist in environments that use and support DEP and ASLR. The two sandboxed applications that people were able to exploit were Internet Explorer 8, which

security researcher Stephen Fewer exploited during the 2011 Pwn2Own contest, and Chrome, which VUPEN Security, a French security research organization, was able to exploit. None of the public demonstrations included any public exploit code. Exploiting vulnerabilities in sandboxed environments significantly drives up the complexity and cost of exploit development. This focus of software vendors to increase the complexity and cost of exploit development through the application of sandboxed environments seems to have been an effective approach, as no public exploit code currently exists for popular sandboxed applications.

One noteworthy side-effect of sandboxing is that, at least in one case with Adobe, on multiple occasions the vendor delayed the release of patches for Adobe Reader X because the vendor's sandbox kept the vulnerability from being exploitable. Ideally, all vendors would employ sandboxing, but the reality is that introducing a sandbox to an existing application is not simple. Sandboxing requires an architectural change, which means additional resources, whether those resources are internal or external. Adobe, for example, collaborated with both Microsoft and Google to bring sandboxing to Adobe Reader X and Acrobat X.

Additionally, as useful as sandboxing is, attackers are migrating away from exploiting vulnerabilities and focusing more on exploiting the human element by convincing users to download and execute malicious content. This is a testament to the fact that attackers always take the easiest route. It is easier for attackers to use social engineering to trick their victims than it is for those attackers to find ways to bypass the current mitigating technologies, such as sandboxing. This is not to say that vulnerabilities are not important to watch for, but rather reflects how attackers are adapting to an evolving security landscape.

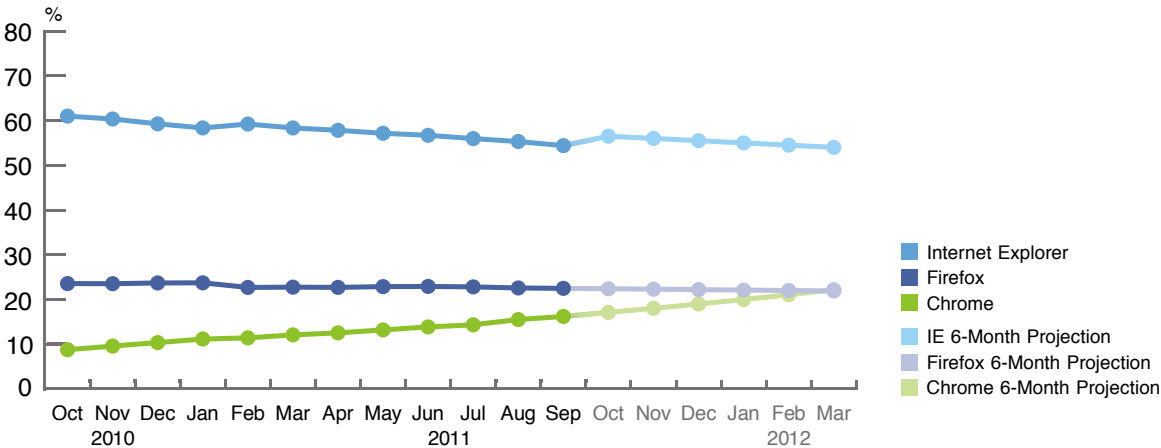
¹¹ Randolph, Kyle. "Inside Adobe Acrobat Protected View." June 14, 2011. Adobe. <http://blogs.adobe.com/asset/2011/06/inside-adobe-acrobat-protected-view.html>.

For the time being, vendors continue to push sandboxing technology throughout their products, as the application of sandboxing technologies greatly increases the complexity of exploiting vulnerabilities. Vendors realize that vulnerabilities will always exist and that they will not be able to find all of them, but vendors can use sandboxing technologies to keep malicious actors from exploiting those vulnerabilities and thus far have been successful.

Chrome Browser Adoption to Surpass Firefox in 2012

As Exhibit 6 shows, the adoption of Google's Web browser, Chrome, has grown from just more than 10 percent of the market share at the end of 2010 to well more than 17 percent of the market share as of October 2011, according to Net Applications.com.¹² This growth places Chrome as the third-most-popular Web browser, behind Firefox and Internet Explorer. As of October 2011, Chrome held a respectable 17.6 percent of the market share while its competitors Firefox and Internet Explorer represented 22.5 percent and 52.6 percent of the market, respectively. Chrome is the only browser to enjoy robust growth in 2011. While Internet Explorer maintains a healthy lead over its competitors, it experienced the largest drop in market share since the beginning of the year, declining from 59.3 percent of the market share at the end of 2010 to the aforementioned 52.6 percent as of October 2011. Firefox's slow decline in market share and Chrome's quick ascent will enable Chrome to become the second-most-popular Web browser in the near future. At current adoption rates, Chrome will surpass Firefox in the Web browser market sometime around March 2012. Internet Explorer will continue to maintain its top spot for some time primarily due to its commanding lead.

EXHIBIT 6: INTERNET BROWSER ADOPTION RATES



¹² NetMarketShare. "Desktop Top Browser Share Trend." Accessed on Oct. 17, 2011. Net Applications.com. <http://www.netmarket-share.com/browser-market-share.aspx?qprid=1&qpcustomb=0&qptimeframe=M&qpsp=141&qnp=13#>.

One possible explanation for Chrome's strong adoption rate is that home users and enterprise users alike are more willing to allow Google to update their browsers whenever a new version is available. Relinquishing control over software patching stems from benefits associated with cost, convenience and security. From a patching standpoint, Google patches its Web browser more frequently and quicker than any other Web browser. In some instances, Google patches Chrome's built-in Flash player and PDF reader quicker than Adobe can release a fix for the same vulnerability in its other products; however, the quick patching scheme comes at a price because IT administrators no longer have control over patching for compatibility issues.

Chrome's recent popularity can also be attributed to Chrome's built-in security mechanisms, particularly its sandbox and group policy compatibility. Group policy is a mechanism IT administrators can use to enforce certain non-security and security-related features on the supporting product. Sandboxing technology has thus far proven to be an effective mechanism to keep malicious users from exploiting security holes in software.¹³ Although not perfect, sandboxing technology has made exploiting the browser much more difficult. As this report discussed earlier, Stephen Fewer had to use three vulnerabilities to circumvent the sandbox of Internet Explorer 8 on Windows 7 during this year's Pwn2Own contest.¹⁴ VUPEN Security was able to bypass Chrome's flash sandbox in addition to circumventing DEP and ASLR to execute arbitrary code by using two exploits earlier this year.¹⁵ No public exploit code currently exists that can bypass Chrome's sandbox.

Vendor Bounty Programs in 2011

Mozilla was one of the first vendors to roll out a bug bounty program in 2004. Google followed a few years later, launching its bug bounty program in early 2010, and new players soon followed to establish their own programs. Barracuda Networks, a company that specializes in security and networking devices, established its bug bounty program in November 2010¹⁶ for vulnerabilities that anyone discovered in its security product line. In 2011 Rapid7, a vulnerability management and penetration-testing organization, followed with its own Metasploit bounty program for which contributors would receive cash payments for modules they developed from their top-5 or top-25 exploit lists.¹⁷ This new bounty program was a shift from the traditional bug programs that vendors had normally used to discover new bugs in existing products.

13 Keizer, Gregg. "Google's Chrome Untouched at Pwn2Own hack match." March 10, 2011. ComputerWorld. http://www.computer-world.com/s/article/9214022/Google_s_Chrome_untouched_at_Pwn2Own_hack_match.

14 Naraine, Ryan. "Pwn2Own 2011: IE8 on Windows 7 hijacked with 3 vulnerabilities." March 9, 2011. ZDNet. <http://www.zdnet.com/blog/security/pwn2own-2011-ie8-on-windows-7-hijacked-with-3-vulnerabilities/8367>.

15 Higgins, Kelly Jackson. "Google, VUPEN Spar Over Hack Chrome Hack." May 11, 2011. InformationWeek. <http://www.information-week.com/news/security/attacks/229500086>.

16 "Barracuda Networks Launches Security Bug Bounty Program." Nov. 9, 2010. Barracuda Networks. http://www.barracudanetworks.com/ns/news_and_events/index.php?nid=423.

17 "Bounty: 30 Exploits, \$5,000.00, in 5 weeks." June 14, 2011. Metasploit Blog. <https://community.rapid7.com/community/metasploit/blog/2011/06/14/metasploit-exploit-bounty-30-exploits-500000-in-5-weeks>.

The trend continued into 2011 when ExploitHub, a marketplace for buying and selling exploits used in penetration tests, rolled out its Requests and Bounty system, in which the company pays security researchers who develop exploits from the company's current list of 12 vulnerabilities.¹⁸ What makes this bounty program unique is that customers making requests for exploit development from ExploitHub's list of vulnerabilities pay the bounty. The bounty amounts range from \$200 US to \$500 US, and the researcher who first develops an exploit for a specific vulnerability receives the bounty. Researchers can earn extra income if different customers make requests for the same vulnerability as long as the exploit stays on ExploitHub. Finally, one of the biggest players to enter the bounty program in 2011 is Facebook.¹⁹ Facebook provides bounties for security bugs that individuals discover in its Web-based social site.

iDefense believes that more software vendors wishing to improve the security and safety of their products will reach out to the security community for assistance in coming years. Barracuda Networks and Facebook took this approach to yield results similar to Google's and Mozilla's. Three weeks after launching its program, Facebook awarded \$40,000 US to its contributors. It is currently unknown how many security bugs Facebook has fixed, but Facebook believes the program is a success due to the high number of quality submissions that security researchers have reported.

Barracuda Networks did not produce similar results during the first 90 days of its program. During this period, Barracuda Networks received only 32 submissions, most of which were of low quality. Additionally, this low submission rate is far lower than expected. This can be attributed to various factors, including the following:²⁰

- To identify bugs, security researchers had to purchase products for which Barracuda Networks was paying bounties, though to address this issue, Barracuda Networks set up a Hacking Lab, which provides researchers the appropriate resources, such as virtualized versions of products to identify bugs.
- Security researchers did not follow bounty program rules and guidelines, which forced Barracuda Networks to reject submissions.

From these experiences, Barracuda Networks revamped the program to ensure that the organization and security researchers produce better results.

Rapid7 and ExploitHub continued the trend in implementing bounty programs; however, their programs steered away from traditional security bug discovery. Both programs offered bounties for the development of exploit code for non-zero-day vulnerabilities.

18 "Development Requests." Oct. 12, 2011. ExploitHub. <https://www.exploitHub.com/request/index/developmentrequests>.

19 "Security Bug Bounty." Oct. 12, 2011. Facebook. <http://www.facebook.com/whitehat/bounty/>.

20 "Barracuda Networks: Bug bounty program not without bumps." Feb. 8, 2011. CSO Online. <http://www.csoonline.com/article/662975/barracuda-networks-bug-bounty-program-not-without-bumps>.

The Rapid7 bounty program offered incentives to security researchers who develop Metasploit modules from a list of 25 high-severity vulnerabilities in both client and server applications. The result was only five modules during the 5-week program, but Rapid7 saw the experience as a success due to participation from both experienced and non-experienced exploit developers. Readers must note that security researchers who sought to develop exploits for the Metasploit Bounty program were limited to just 1 week.²¹

What makes ExploitHub’s exploit bounty program unique is that customers, instead of the vendor, pay the bounties. The program is still in its infancy, so results are not yet available. Readers must note that the exploits that researchers in this program developed are for existing vulnerabilities.

Since late 2010, iDefense saw the emergence of four new bounty programs. Success from the Mozilla, Google and Facebook bounty programs demonstrates that engaging the security researcher community through financial compensation has played a key role in improving the security of existing products. The small number of out-of-band patches that vendors released this year compared to last year, as CVE-IDs in Exhibit 7 demonstrate, may also indicate that security researchers are holding off their findings for financial compensation.

EXHIBIT 7: DEMONSTRATING A DECREASE IN OUT-OF-BAND PATCHES

2010	2011
CVE-2010-2862	CVE-2010-4476
CVE-2010-2883	CVE-2011-0609
CVE-2010-2884	CVE-2011-0611
CVE-2010-3654	CVE-2011-0610
CVE-2010-4091	
CVE-2010-2568	
CVE-2010-3332	
CVE-2010-0188	
CVE-2010-1297	
CVE-2010-0806	

In 2012, iDefense analysts predict that organizations with a substantial online presence, such as Twitter and Amazon, will adopt bug bounty programs. Implementation of such programs will reflect a broader trend whereby organizations that have not typically concerned themselves with vulnerabilities or exploits related to their own products will embrace bug bounty programs in their efforts to leverage the security researcher community to improve product security.

21 "Metasploit Exploit Bounty - Exploit List." June 13, 2011. Rapid7 Community. <https://community.rapid7.com/docs/DOC-1467>.

CONCLUSION

2011 presented plenty of evidence that malicious actors are as determined and persistent as ever in their pursuit of financial or strategic gain, and the tools at their disposal continue to develop as attackers weave elements from more-complex code into their attacks. A perfect example of this is the evolution of the Zeus Trojan into an open-source crime kit with the public release of the Zeus Trojan's source code. The growing use of free domain providers as conduits for malicious activity and the increased complexity of exploits is an example of the innovation, perseverance and tenacity of malicious actors. Unfortunately, success breeds success even for criminals, and security organizations must remain vigilant in the face of these persistent and adaptive threats.

ABOUT VERISIGN IDEFENSE SECURITY INTELLIGENCE SERVICES

Verisign iDefense Security Intelligence Services gives information security executives access to accurate and actionable cyber intelligence related to vulnerabilities, malicious code and global threats 24 hours a day, 7 days a week. Verisign iDefense in-depth analysis, insight and response recommendations help keep businesses and government organizations ahead of new and evolving threats and vulnerabilities.

LEARN MORE

For more information about Verisign iDefense Security Intelligence Services, please e-mail learnmore@verisign.com or visit us at http://verisigninc.com/en_US/products-and-services/network-intelligence-availability/idefense/index.xhtml.

ABOUT VERISIGN

Verisign is the trusted provider of Internet infrastructure services for the networked world. Billions of times each day, our services allow public and private sector organizations, along with consumers all over the world, to engage in trusted communications and commerce.



VERISIGN™



VerisignInc.com

© 2012 VeriSign, Inc. All rights reserved. VERISIGN and other trademarks, service marks, and designs are registered or unregistered trademarks of VeriSign, Inc. and its subsidiaries in the United States and in foreign countries. All other trademarks are property of their respective owners.