Secunia
Stay Secure

Secunia 2009 Report

Letter from the CEO: No standard for updating in 2009

Dear Customers and Partners,

Once again I would like to welcome you to the Secunia 2009 Report. This year, the focal point is the security threat posed by vulnerabilities in 3rd party programs on Microsoft Windows – a threat that has intensified over the last 12 months.

In 2009, the Secunia Research Team discovered 145 of the 7,605 vulnerabilities identified during the year, resulting in 65 Secunia Research Advisories and 4,620 Secunia Advisories. The majority of the vulnerabilities were in 3rd party programs, supporting the impact that they have had on the IT security scene in 2009. Who does not remember the case of Adobe that continued to deliver out-of-date versions of Adobe Reader for download, thereby, leaving the users vulnerable to exploits?

Considering the extent to which 3rd party programs are contributing to the overall threat picture, it is alarming that a large proportion of companies and private users continue to ignore 3rd party vulnerabilities. Even though they were more representative in 2009, the market pull and understanding of 3rd party program vulnerabilities was still some what limited.

While some are driven by mere ignorance, a substantial number of knowledgeable companies and end-users are not updating, patching, 3rd party programs merely due to the cumbersome process and immeasurable scope. Bottom-line is that software distribution and patch management of 3rd party programs in Windows environments by definition are tedious and dreaded tasks, due to the lack of standards and an old school approach by the existing solutions. It is therefore neglected, even though it constitutes a substantial security risk.

This causes an alarming amount of unpatched software on home and business systems that is freely available for criminals to attack and exploit. Especially with the sophisticated and effective tools for updating Microsoft products, Windows Update, and WSUS, the incentive for criminals to target the 3rd party programs is even higher, generating a greater impact.

Aware of this underlying construct, Secunia called out at the yearly RSA Conference 2009, encouraging the software industry to join forces and develop a unified standard software distribution and patching mechanism, benefiting IT administrators as well as end-points. This call-out was welcomed by many, however, little action was taken; it was clear, that if something had to materialise Secunia had to take the lead.

Being a market driven business, with the ultimate purpose of providing simple solutions to solve often complex vulnerability issues, Secunia saw the following as the best way to take action: rely on something widely used and acknowledged and make it even better by complementing it with Secunia's own solutions.

This has resulted in the ‘Secunia Corporate Software Inspector integrated with WSUS for 3rd party Patch Management’, offering a revolutionary approach to patching with automatic repackaging of most security patches. For the first time in the IT-security history, companies are able to manage and patch both Microsoft and 3rd party programs in the same simple way. This is the first step towards a qualified and user-friendly standard - headed by Secunia!

Even though Secunia has now taken another step into IT-security, core of the business remains the verified and accurate Vulnerability Intelligence, providing our customers with the best vulnerability coverage in the industry.

Finally, I would also like to add that Secunia continues to be a healthy company. Although 2009 was a challenging year business wise, impacted by the financial crisis, Secunia moved itself forward in terms of financial performance, almost doubling the headcount, sustaining strong organic growth with no bearing debt, and achieving the strategic targets. In addition, the prospects for 2010 look very promising.

I hope you enjoy reading our 2009 Report.

Patch and Stay Secure,

Niels Henrik Rasmussen

CEO and Founder

The 0-days of 2009

The number of 0-day vulnerabilities remain low – a positive trend, as they are the most difficult to manage from a risk management point of view

They are notoriously known for the headaches and concerns they cause many IT departments worldwide – the 0-days vulnerabilities. Not only are they patch-less, since no patches have been developed by the vendors, but they also circumvent the reactive security measures such as Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), and Anti-Virus (AV), as these cannot detect the exploitation of a 0-day without a signature update. Bottom line, mitigating the risk posed by 0-day vulnerabilities is almost impossible, and no pre-emptive actions can be taken.

A **0-day** is a vulnerability that is being exploited actively before public information describing the vulnerability has been released.

In fact, IT departments are left with merely four reactions upon receiving the information of a 0-day vulnerability:

- Un-install the software
- Reconfigure the software or other systems (if possible)
- Wait for signature updates to the reactive security measures
- Take a calculated risk and wait for the vendor to issue a patch

Key in managing 0-days is immediate access to information on the systems affected, the most likely attack vectors, and the proper rating of the vulnerability. To ensure this without having to dedicate substantial resources to monitoring and evaluating information, companies can rely on tools such as the Secunia Vulnerability Intelligence Feed (VIF) and the Secunia Enterprise Vulnerability Manager (EVM). Both of these alert companies of all new vulnerabilities, including 0-days. Further, the Secunia Corporate Software Inspector (CSI) pinpoints the exact systems affected, thereby enabling identification of the business units affected.

The optimal reaction to 0-day vulnerabilities:

(1) instantly get an overview of the systems affected, (2) the most likely attack vectors, (3) and a proper rating of the vulnerability itself.

Based on this information, it becomes easier to assess the different options for appropriate counter measures versus the associated costs and risks.

In 2009, a total of 12 0-day vulnerabilities were identified, of which close to 60 percent were in 3rd party programs. This is more than double as compared to 2008, where only three 0-days in non-Microsoft programs were identified. The five 0-days reported in Microsoft were primarily in the older versions of Microsoft Excel, Powerpoint, DirectShow, and Office Web components, supporting the following two trends:

0-day attacks from 2006-2009

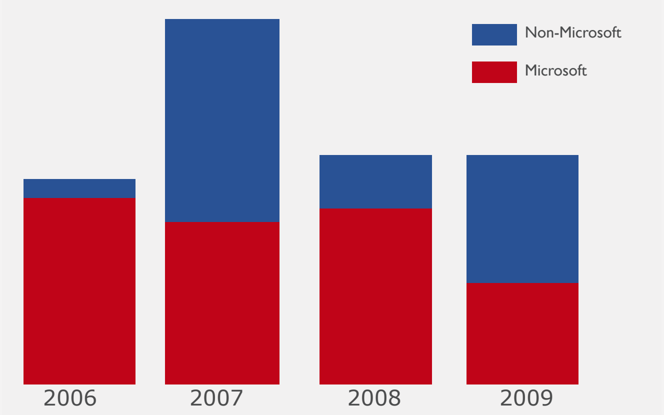


Figure 1: In 2006, all but one 0-day case concerned Microsoft vulnerabilities. In 2007, the number of Microsoft related cases declined to just 8 out of 20, and in 2008, 9 cases out of 12 were Microsoft-related. In 2009, 5 out of 12 were Microsoft related.

1.It is becoming increasingly difficult to find and reliably exploit vulnerabilities in newer Microsoft programs.

2.Criminals find attacking via widely used 3rd party programs a more viable approach, than the popular Microsoft targets of the past, such as Office.

Microsoft Office used to be the top target for 0-days, but better coding and the implementation of new security models in Windows Vista, Windows 7, and in newer versions of Microsoft Office has reduced the success of many exploits; unless the criminals made a customised exploit to explicitly attack newer Microsoft products. However, exploits written for 3rd party software, such as, Adobe Reader, usually "allow" the same exploits to run smoothly on all Windows versions without requiring specialised efforts by the criminals. This is supported by the ratio of 0-days in Adobe products, representing one third of the total 2009 0-days.

One simple explanation for the criminals' attraction to 3rd party programs is that many programs do not use the anti-exploitation mechanisms present in modern operating systems, and thus allow easier and more reliable exploitation.

The explanation for Adobe Reader is slightly different, as it does use these mechanisms. However, it supports JavaScript and includes Flash Player, which can be utilised via various techniques to often make exploitation quite reliable, regardless of the operating system.

Summarising the observations of this year's 0-days, with an increasing popularity of exploiting vulnerabilities in 3rd party programs, indicates that the criminals are aiming at improving the value of their "investments"; that is, they direct their attention and efforts towards the programs that will inflict as many as possible, creating a higher return on their investment. Vulnerabilities in 3rd party programs are generally easier to discover and exploit, as is the case with Adobe products, as compared to newer Microsoft products, and, further, users are often slower at patching 3rd party programs. The data from the Secunia Online Software Inspector (OSI) and the Secunia Personal Software Inspector (PSI) prove this point, with end-users being slower at patching Adobe products as compared to patching Microsoft software, and popular browsers, such as - Mozilla, Google, and Opera.

Secunia Advisories covering 0-days in 2009

SAID	Vulnerability	Date
SA36983	Adobe Reader/Acrobat Multiple Vulnerabilities	2009-10-09
SA36365	SPIP Database Backup Authentication Bypass Vulnerability	2009-08-20
SA35948	Adobe Flash Player Multiple Vulnerabilities	2009-07-23 *
SA35949	Adobe Reader/Acrobat SWF Content Arbitrary Code Execution	2009-07-23 *
SA35800	Microsoft Office Web Components Multiple Vulnerabilities	2009-07-13
SA35683	Microsoft DirectShow Streaming Video ActiveX Control Vulnerabilities	2009-07-06
SA35268	Microsoft DirectShow QuickTime Parsing Multiple Vulnerabilities	2009-05-29
SA34897	PJBlog3 “action.asp” SQL Injection Vulnerability	2009-04-24
SA34572	Microsoft PowerPoint OutlineTextRefAtom Parsing Vulnerability	2009-04-03
SA34405	JustSystems Ichitaro Products Unspecified Code Execution Vulnerability	2009-03-20
SA33954	Microsoft Excel Two Vulnerabilities	2009-02-24
SA33901	Adobe Reader/Acrobat Multiple Vulnerabilities	2009-02-20

* These two represent one 0-day vulnerability in the total count

Technical background on the Adobe 0-days

SA37690: Adobe Reader and Acrobat render PDF files and support the JavaScript for Acrobat API (described in a public specification written by Adobe) allowing a document to execute script in response to events. A use-after-free error when executing the Doc.media.newPlayer() API method can be leveraged for arbitrary code execution.

SA36983#1:A PDF document can contain stream objects, which may define data such as images. They can be encoded or compressed using a number of different filters. An integer overflow when parsing parameters to the "FlateDecode" filter when using a specific prediction algorithm will overflow a heap-based buffer and is exploitable for code execution.

SA35948 + SA35949: While Flash Player is specifically designed to parse SWF content, the same parser is included in other Adobe products such as Reader and Acrobat to parse embedded and stand-alone SWF objects. Insufficient validation of AVM2 (ActionScript Virtual Machine 2) bytecode can result in an attacker specifying the pointer to an object. This can be leveraged for arbitrary code execution.

SA33901: The PDF "JBIG2Decode" filter is used to provide image data via a JBIG2-encoded stream, which includes an arbitrary number of JBIG2 segments. An array indexing error when processing a JBIG2 segment having an overly large segment page association value can be exploited to corrupt arbitrary memory.

The Achilles heel of PCs - 3rd party programs – but why?

3rd party programs have received increased focus from criminals during the last three years. Ironically, the primary reason for this trend is the increased frequency of updating software programs

10 years ago the mantra for any IT-administrator was: *If it ain't broken, don't fix it!*

The problem is that any software with a known vulnerability, is "broken" and needs fixing, even though the user does not see it as "broken".

The increase in exploitation of Microsoft programs in the beginning of the decade made it absolutely necessary to apply Microsoft related updates fast and efficiently. Microsoft naturally realised this need, developing and improving "automatic updating" via Windows Update, and eventually promoting it widely. Fortunately, these initiatives by Microsoft have been very successful, and today the majority of home users apply Microsoft patches within a few days via automatic updates, and corporate IT-operations deploy Microsoft patches using WSUS.

Microsoft updating tools provide a very efficient and effective "patch management" process for Microsoft products on millions of PC. Within a few days, the value of an exploit for a Microsoft vulnerability has diminished significantly, and after just a few weeks, all the updated PCs with patched Microsoft programs are immune to the exploit. This means that the window of exploitation for Microsoft products is substantially reduced, and criminals have to search for other ways to attack PCs.

One way is to maintain a focus on Microsoft products, yet targeting the PCs belonging to users that neglect updating and, therefore, become vulnerable to very old exploits. Here the criminals can, in many cases, merely reuse old and well known exploits, which the un-updated PCs are not protected against. Naturally, this means that the more outdated the installed programs, the larger the window of exploitation for the criminals, having millions of exploits to use. However, due to the ease of updating Microsoft, the target potential is reduced over time, as is the return on the criminals efforts.

A more attractive and "cost efficient" way for criminals to attack systems is, therefore, to focus on the 3rd party programs, which the Microsoft patch management platforms do not support. And due to the low patching frequency and the large install base of the 3rd party programs, they represent a true Achilles heel to system security.

As 3rd party programs are not supported by Windows Update, substantial manual resources and time is needed to identify and update the approximately 50 different programs installed on every single PC. In large IT environments this becomes an immeasurable activity for IT administrators and users, meaning that the majority of PCs are not updated when security patches are released for media players, Office products, ActiveX controls, browser plug-ins, etc.. This leaves a window open for criminals to exploit – the Achilles heel of networks otherwise secured by advanced and expensive reactive measures, such as IDS/IPS, AV, and Firewalls.

Gartner

"Deployment of non-Microsoft patches is often significantly slower and less organized. All Internet-based applications, especially browsers and browser plug-ins (i.e.,Adobe and Apple QuickTime), should be a top patching priority."

Gartner, "Top10 Steps to avoid Malware infections" September 2009

It is of course not all 3rd party program vulnerabilities that are equally easy to reverse engineer and create "reliably" working exploits for, however, with an update pattern where a large proportion of users update as infrequently as once a year, or less, there are almost always millions of PCs vulnerable to one of the easier vulnerabilities. Some of the common 3rd party programs that were popular targets for criminals in 2009 include Flash Player, Adobe Reader, and Sun Java, due to both their installation base and, of course, the patching infrequency. During 2009, there have been three updates for Adobe Flash Player 10 fixing 23 vulnerabilities, six updates for Adobe Reader 9 fixing 56 vulnerabilities, and four updates for Sun Java JRE 1.6 fixing 41 vulnerabilities.

Statistics from the Secunia PSI show that less than 1% of all users are at a 100% patch level, when they run the Secunia PSI for the first time.

Adobe

"We know that getting people updated and keeping them updated is the number-one thing we can do in terms of keeping them protected against attacks"

Brad Arkin, Director of product security and privacy at Adobe January 2010

In addition to these more popular programs, which are frequently targeted by both vulnerability researchers and criminals, it is, however, also important to pay attention to other programs such as ActiveX controls and browser plug-ins. Some of the most exploited vulnerabilities in the beginning of 2009 were in fact in less known ActiveX controls (according to Microsoft SIR Vol 7.).

The vast majority of 3rd party programs are not updated very frequently, despite the presence of known vulnerabilities that are being exploited actively and even receive broad media attention. The only general exception with regards to 3rd party programs is browsers such as Firefox, Chrome, and Opera, because they follow an updating process similar to Windows Update.

Vulnerabilities in 3rd party programs are today a serious part of the overall threat picture, and it is essential to understand and acknowledge that even though expensive security measures are in place, it only takes one vulnerability in a 3rd party programs for all these defences to be circumvented and compromise the network.

One vulnerable program is all it takes to compromise a corporate network.

System administrators need to stay informed about the latest vulnerabilities, apply patches to eliminate the vulnerabilities as soon as possible, as well as assess the risk of the vulnerability in the applicable environment. All this to ensure that the patch is applied in a timely manner to limit the window of opportunity for attackers.

COMPUTERWORLD

“Future techies will be rightfully incredulous that there isn't a single software updating system for all the installed software. Imagine there were gas stations for General Motors, Toyota and Volvo cars and that owners of those cars could only be serviced at stations dedicated to them. That's the disgraceful system we all live with today.

Michael Horowitz, Columnist,
Computerworld.com, December 2009

Secunia Research Highlights 2009

An interesting year where particular products were scrutinised

Following a very solid effort in the 4th quarter 2009, the Secunia Research team discovered by own efforts a total of 145 vulnerabilities, reserving 65 Secunia Research Advisories. The outcome of our yearly vulnerability discovery day, where the entire Research Team spends a day finding vulnerabilities, was acceptable, and resulted in vulnerability discoveries in products from vendors like Adobe and Microsoft.

Customers and third parties sometimes ask us about the criteria that we use in deciding to research a given product, and if it is possible to "hire" the Research Team to audit specific products. While we are very happy that the skills of our Research Team are acknowledged, the answer to the latter question is: "No".

The answer to the first question is a bit longer. Each researcher is in general responsible for picking his/her own targets, but once in a while the Chief Security Specialist, managing the team, may ask the team or certain members of the team to focus on a specific product. Usually, we focus on enterprise software and widely used client programs – pretty much the software that is commonly used by our customers and community.

By finding vulnerabilities in these programs, and by working together with the vendors on getting these vulnerabilities fixed, we ensure that the programs are more secure and safe.

Sometimes, we may also decide to focus on particular programs, a particular vendor, or a particular functionality. In 2009, some of the team members focused on Microsoft software, resulting in 13 Secunia Research advisories, covering 15 vulnerabilities discovered in Microsoft products.

Another product that we specifically decided to look at in 2009 was Winamp. At the beginning of the year, a vulnerability was discovered in the parsing of CAF audio files. In addition, by the end of 2009, one of the team members discovered an additional four vulnerabilities in the Module Decoder Plug-in for Winamp.

For a brief period we examined specific functionality: The parsing of JBIG2 content. In February, some vulnerabilities were reported in Poppler when parsing JBIG2 content. By the end of the month, a 0-day vulnerability was reported in Adobe Acrobat/Reader, also related to the parsing of JBIG2 content.

It seemed likely that both of these and other products would have additional vulnerabilities when parsing this type of content. One of the team members was, therefore, tasked with scrutinising various products, implementing JBIG2 parsing functionality, and ended up finding vulnerabilities in Foxit Reader, Adobe Reader, Xpdf, CUPS, and Ghostscript.

To wrap this up we have collected the following statistics for the vulnerabilities discovered in products from some of the major software vendors:

Vendor	No. of Vulnerabilities
Adobe	10
Apple	2
Google	1
HP	2
Microsoft	15
Mozilla	3
Novell	3
Oracle	2
Sun	4
VMware	4

The majority of the vulnerabilities discovered (about 90%) were found via source code auditing and closed-source static analysis, not fuzzing, which is probably one of the most common approaches used in vulnerability research. However, to increase our efficiency and the output, we have this year also focused on creating some great in-house tools and fuzzers (all named: "Bruce"). We are, therefore, very excited to see what 2010 brings.

Stay Secure

Carsten Eiram

Chief Security Specialist

2009 - Vulnerabilities found by Secunia Research

Total Research Papers: 65,

Pending Disclosures: 17

http://secunia.com/secunia_research/

2009-65	Google Chrome Pop-Up Block Menu Handling Vulnerability
2009-64	PDF-XChange Viewer Content Parsing Memory Corruption Vulnerability
2009-63	Adobe Shockwave Player Four Integer Overflow Vulnerabilities
2009-62	Adobe Shockwave Player 3D Model Two Integer Overflows
2009-61	Adobe Shockwave Player 3D Model Buffer Overflow
2009-60	N/A - RESERVED - Pending Disclosure
2009-59	Microsoft - RESERVED - Pending Disclosure
2009-58	Adobe Illustrator Encapsulated Postscript Parsing Vulnerability
2009-57	Winamp Oktalyzer Parsing Integer Overflow Vulnerability
2009-56	Winamp Ultratracker File Parsing Buffer Overflow
2009-55	N/A - RESERVED - Pending Disclosure
2009-54	Microsoft - RESERVED - Pending Disclosure
2009-53	Winamp Impulse Tracker Sample Parsing Buffer Overflow
2009-52	Winamp Impulse Tracker Instrument Parsing Buffer Overflows
2009-51	DevIL DICOM "GetUID()" Buffer Overflow Vulnerability
2009-50	Sun Microsystems - RESERVED - Pending Disclosure
2009-49	Sun Microsystems - RESERVED - Pending Disclosure
2009-48	HP Power Manager "formExportDataLogs" Directory Traversal
2009-47	HP Power Manager "formExportDataLogs" Buffer Overflow
2009-46	RhinoSoft Serv-U TEA Decoding Buffer Overflow
2009-45	Mozilla - RESERVED - Pending Disclosure
2009-44	Novell iPrint Client Date/Time Parsing Buffer Overflow
2009-43	Gimp PSD Image Parsing Integer Overflow Vulnerability
2009-42	Gimp BMP Image Parsing Integer Overflow Vulnerability
2009-41	Lateral Arts Photobox uploader ActiveX Control Buffer Overflow
2009-40	Novell iPrint Client "target-frame" Parameter Buffer Overflow
2009-39	Microsoft - RESERVED - Pending Disclosure
2009-38	Roxio Creator Image Rendering Integer Overflow Vulnerability
2009-37	VMware - RESERVED - Pending Disclosure
2009-36	VMware - RESERVED - Pending Disclosure
2009-35	Mozilla Firefox Floating Point Memory Allocation Vulnerability
2009-34	Microsoft - RESERVED - Pending Disclosure
2009-33	Microsoft - RESERVED - Pending Disclosure
2009-32	Microsoft - RESERVED - Pending Disclosure
2009-31	Microsoft - RESERVED - Pending Disclosure
2009-30	Microsoft - RESERVED - Pending Disclosure
2009-29	Microsoft PowerPoint Freelance Layout Parsing Vulnerability
2009-28	Microsoft - RESERVED - Pending Disclosure
2009-27	OpenOffice.org Word Document Table Parsing Buffer Overflow
2009-26	OpenOffice.org Word Document Table Parsing Integer Underflow
2009-25	VMWare VMnc Codec Mismatched Dimensions Buffer Overflow
2009-24	Adobe Reader JBIG2 Text Region Segment Buffer Overflow

2009-23	Oracle BEA WebLogic Server Plug-ins Certificate Buffer Overflow
2009-22	Oracle BEA WebLogic Server Plug-ins Integer Overflow
2009-21	Ghostscript jbig2dec JBIG2 Processing Buffer Overflow
2009-20	IrfanView Formats Plug-in XPM Parsing Integer Overflow
2009-19	Mozilla Firefox Java Applet Loading Vulnerability
2009-18	CUPS pdftops JBIG2 Symbol Dictionary Buffer Overflow
2009-17	Xpdf JBIG2 Symbol Dictionary Buffer Overflow Vulnerability
2009-16	Garmin Communicator Plug-In Domain Locking Security Bypass
2009-15	Microsoft - RESERVED - Pending Disclosure
2009-14	Adobe Reader JBIG2 Symbol Dictionary Buffer Overflow
2009-13	Novell eDirectory iMonitor "Accept-Language" Buffer Overflow
2009-12	Microsoft Excel String Parsing Integer Overflow Vulnerability
2009-11	Foxit Reader JBIG2 Symbol Dictionary Processing Vulnerability
2009-10	QuickTime Sorenson Video 3 Content Parsing Vulnerability
2009-09	Orbit Downloader Long URL Parsing Buffer Overflow
2009-08	Winamp CAF Processing Integer Overflow Vulnerability
2009-07	libsndfile CAF Processing Integer Overflow Vulnerability
2009-06	Apple QuickTime MS ADPCM Encoding Buffer Overflow
2009-05	Free Download Manager Torrent Parsing Buffer Overflows
2009-04	OpenX Multiple Vulnerabilities
2009-03	Free Download Manager Remote Control Server Buffer Overflow
2009-02	AproxEngine Multiple Vulnerabilities
2009-01	Microsoft Excel Record Parsing Array Indexing Vulnerability

Software Inspection Results

Results from inspections conducted by Secunia's Online Software Inspector during 2009

The Secunia Online Software Inspector (OSI) is an online scanner that checks PCs for vulnerabilities in the 94 most common programs, including old versions that have been end-of-life (EoL) for a long time. In 2009, the OSI conducted 1,486,506 scans on an unknown number of individual PCs; that is, some have been scanned several times, whereas others have only been scanned once.

Reports based on the Secunia Personal Software Inspector statistics are currently being drafted, covering programs from thousands of different vendors. These reports will be more thorough, and will give a much more accurate picture of the security state of home PCs. These reports will be released later in 2010.

All figures are based on aggregate numbers of these scans. Thus a user, who first conducted a scan, then patched one or more programs, and scanned again later, is counted as two scans.

Top 10 most secure products 2009

The top 10 list for the most secure programs in 2009 clearly shows that programs, which are covered by Windows Update, are updated more frequently. However, it is also noteworthy that Apple iTunes, Firefox, and Thunderbird all make it to the top 10, as these programs also have auto updating mechanisms similar to Windows Update, reminding the user, persistently, until the update is installed.

Name	Unpatched
Microsoft Windows Media Player 11.x	1.48 %
Microsoft Windows Live Messenger 8.x	1.88 %
Microsoft Outlook Express 6	1.95 %
Apple iTunes 9.x	3.34 %
Microsoft Windows Media Player 9.x	3.46 %
Microsoft Internet Explorer 8.x	3.61 %
Microsoft Windows Media Player 10.x	4.48 %
Microsoft Internet Explorer 7.x	6.86 %
Mozilla Firefox 3.5.x	9.33 %
Mozilla Thunderbird 2.x	10.66 %

There is, however, a caveat with Apple iTunes, as the figures for iTunes 7.x and iTunes 8.x both are at 15%. The low number for iTunes 9.x may be caused by Apple having issued only 1 update for this version.

Two programs marked themselves by the high percentage of unpatched installations, namely RealPlayer 11 and Sun Java JRE 1.6 with 32.8% and 43.1% respectively. The primary reason for this is most likely to be found in tedious updating processes and a general lack of clear communication to the end-users of the updating need.

Comparison to 2008

Java and Flash

In 2009, Sun updated the installer for Sun Java JRE to automatically remove old versions of Sun Java, rather than leaving the old versions on the PC. This initiative was expected to improve the statistics for Sun Java, however, the insecurity rate has in stead increased from 38% in 2008 to 43.1% in 2009.

Adobe made a similar move for Adobe Flash Player 10 and it helped. Adobe Flash Player 10 is now at 24.2% insecure rate, as compared to 34% for Adobe Flash Player 9 in 2008.

Compared to the 10 "best" programs, which are within 10% insecure rate, it is clear that vendors like Sun and Adobe still need to improve and offer better updating mechanisms for their users.

It will be interesting to see if the new Adobe updater currently being tested will further improve the situation for Adobe users during 2010.

End-of-Life detections

Several of the programs detected by the Secunia OSI in 2009 have been EoL for a substantial period of time, counting 26 programs and accounting for a total of 313,000 installations. All 26 programs have known vulnerabilities for which there are no patches available from their vendors. This includes installations of old versions of Adobe Flash Player, Adobe Reader, Apple Quicktime, and Mozilla Firefox.

End-of-Life means that the program no longer is supported by the vendor. End-of-Life software should be considered vulnerable due to lack of reliable information and security updates from the vendor.

A program, which has reached EoL will no longer be patched by the vendor, and should, therefore, be uninstalled immediately. Even though a user may not intend to use them very frequently, it is likely that they are still associated with specific files and media types, and may be automatically launched when clicking on, downloading, or viewing these file types.

Browser security

In January 2010, almost everybody spoke about browser security, because of the Internet Explorer 0-day attack on Google, and the subsequent Google / China dispute.

0-day attacks are rare as described in the "The 0-days of 2009" section of this report. Usually, 0-days are exploited in targeted attacks, and less frequently in broad scale attacks on "normal" users.

The vast majority of broad scale attacks, which happen on a daily basis, exploit older vulnerabilities. In fact they do not very often target browsers, but rather plug-ins and programs that can be launched from a browser.

2009 average unpatched rate for browsers

Brower	Rate
Internet Explorer 8	3.6 %
Internet Explorer 7	6.9 %
Firefox 3.5	9.3 %
Opera 10	14.1 %
IE 6	14.3 %
Opera 9	16.1 %
Firefox 3.0	17 %
Safari	22 %
Google Chrome 3	24.7 %

As the above statistics indicate, some of the most popular browsers are also the ones, which users update most frequently, and thus have a low insecure rate. This rapid update rate makes it too costly for criminals to write exploits targeting browser vulnerabilities.

Please note that the real figure for Chrome could be less than those detected by the Secunia OSI. This misrepresentation is caused by the fact that Chrome leaves old versions behind on the system after an update.

Web Application Security

Web application vulnerabilities – a criminal's favourite

As illustrated below, web application vulnerabilities continue to represent a fairly large percentage of the regularly reported vulnerabilities, and that whether they are self managed programs, hosted solutions like Google and Yahoo, or social networking sites; web programs remain a favourite among criminals.

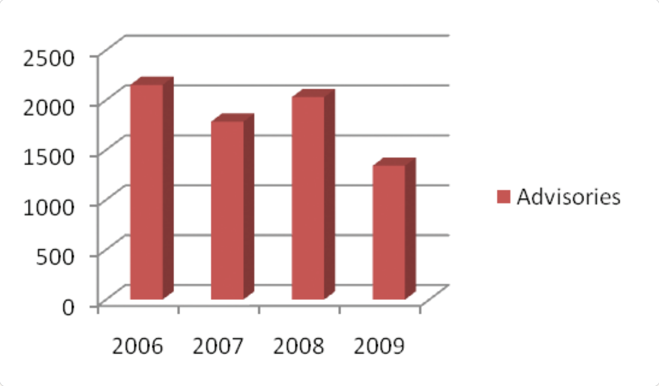


Figure 1: Number of web application advisories issued by Secunia yearly.

Most common web application attacks include cross-site scripting, cross-site request forgery, SQL injection, and file inclusion attacks.

Looking at the statistics in the figure below, it is clear that in 2006 the most common attacks were cross-site scripting, SQL injection, and file inclusions, whereas cross-site request forgery was not very prevalent. In 2009, while cross-site scripting and SQL injection are still very prevalent, file inclusion vulnerabilities have reduced and cross-site request forgery is on the rise.

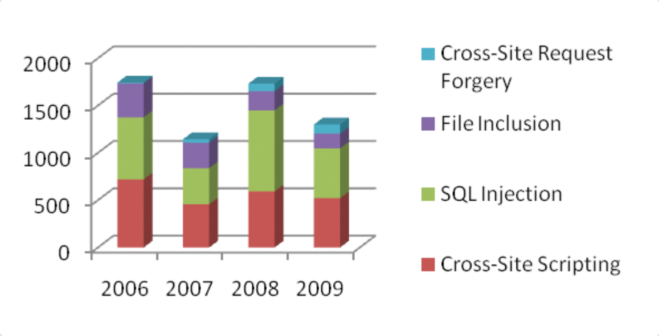


Figure 2: Number of web application vulnerabilities reported yearly

Secunia's analysis shows that most of the vulnerabilities reported in web application are written in PHP, including free and commercial scripts, as well as full fledged enterprise software. PHP has long been a favourite among web application programmers, due to its ease of use, which, however, has also meant that it has been preferred by the less experienced programmers, who tend to focus less on security and more on fast application development.

Next in line is ASP, but the gap between the two languages is very large. ASP is followed by programs written in Java.

Recent data shows a decline in vulnerabilities in PHP programs. Many tutorials have been written on writing secure code in PHP, but vulnerabilities are still being found. This indicates that although the trend is improving, progress still needs to be made.

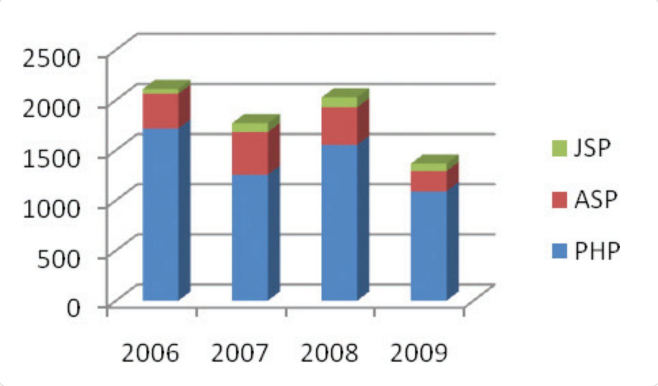


Figure 3: Vulnerabilities found in commonly used script languages

Vulnerabilities also affect popular open source projects like Drupal and Joomla. The good news is that large enterprises, individual programmers, and open source coders are becoming aware of the security issues that affect web programs, and improvements are being made to the frameworks for web application development. Both Drupal and Joomla projects have security teams, conducting security audits, and working towards improving the security of the projects.

Vendors like Microsoft and Sun have enhanced frameworks to integrate security features, and it is worth noting that the number of vulnerabilities reported in web programs is reducing. This is not because people are not conducting enough vulnerability research, but rather due to vendors becoming increasingly aware of the need to write secure code. Quick and dirty "hacks" to solve code issues, which frequently introduce vulnerabilities, is now being more carefully considered before implementation.

Fewer instances of vulnerabilities are also being found in the more recent written programs, as compared to the ones written a few years ago. But there is still a long way to go. Despite all the efforts, vulnerabilities are still being found, and the criminals continue to find new ways to attack. With companies increasingly adopting technologies like ASP.Net, EJB, and web services for web development, the number of traditional attacks are reducing, if not becoming rare. However, with more secure code being written, sophisticated attacks are becoming prevalent. Of the more recent ones, SA37649 & SA37642 leverage certain functionality within PHP to execute arbitrary PHP code.

It is not possible to predict the future of web application vulnerabilities, however, it is fair to claim that criminals are persistent, and will keep innovating new techniques to find and exploit vulnerabilities. The general observation is that security of web programs has improved a lot over the last few years, but there is still a gap that needs to be filled in terms of writing "hack proof" programs. Only time will tell what new vulnerabilities arise with new technologies in the future.

Stay Secure

Chaitanya Sharma, Security Specialist

Secunia Advisory Statistics

Method

The statistics shown in this section cover 1st January 2009 to 31st December 2009. When comparing these statistics to other sources, it is important to keep in mind that the quality and interpretation of data is different depending on who presents it.

Secunia validates and verifies the vulnerability information we gather. In this process, we often arrive at conclusions different from those originally reported because of reasons, such as - other versions or related products are affected, the alleged vulnerability is merely a bug, the software in question is in beta, the report is erroneous and irrelevant, the reported issue already has been described, etc. This naturally makes it difficult to compare our data with the many other sources, who do not conduct equally extensive verification and validation of the vulnerabilities reported.

Number of Advisories

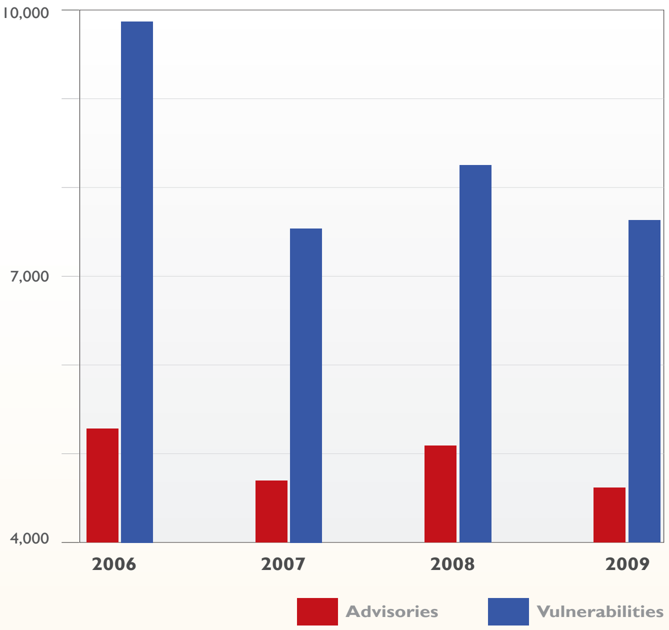
During the last four years, we have seen a fairly steady level of advisories and vulnerabilities. We do not see the drop to 4,620 advisories in 2009 as an indicator of a general trend, and we expect the volume of advisories and vulnerabilities to remain at this level in 2010.

Despite the strong focus on vulnerabilities, fuzzing, software testing, and writing secure code over the past years, it seems that we are not going to see a significant change in the number of vulnerabilities discovered any time soon. This may be caused by a number of factors, such as more programmers, more complex software code, among others. However, the most important is probably the imbalance between the amount of software being developed and the number of vulnerability researchers. Today, more and more software is being developed every day at a more rapid pace, however the number of vulnerability researchers who can identify vulnerabilities in such software is not increasing to the same extent.

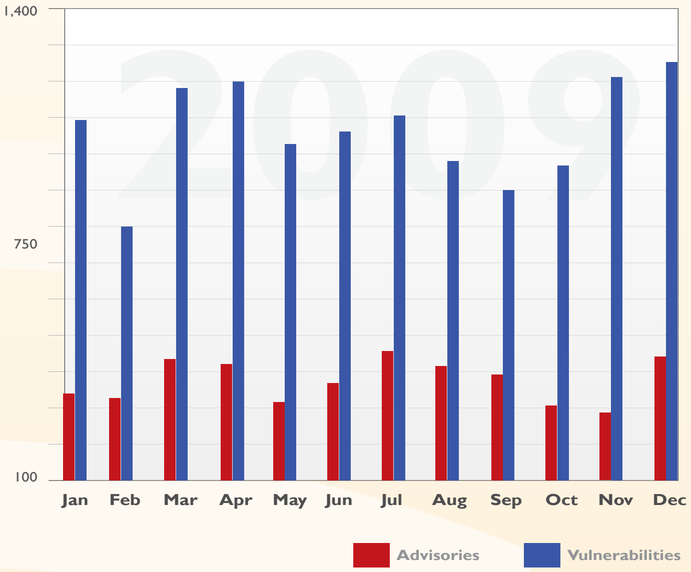
Further, many of the skilled researchers tend to focus their efforts on the most popular software, thereby constantly raising the bar for finding the next vulnerability in the latest versions of popular high-profile software. This naturally also limits the number of discoveries.

More and more vendors seem to be focusing on securing their software, and many vendors seem to conduct basic auditing of their software using fuzzing tools prior to release, thus reducing the number of vulnerabilities. While such practices do not eliminate typical coding errors, we continue to see and discover other 'hard to find' vulnerabilities, that are not easily caught by simple fuzzing, as well as vulnerabilities caused by design and implementation errors.

Secunia Advisories by year



Secunia Advisories month by month in 2009



Note: *The number of vulnerabilities exclude duplicates created by the "update for" advisories for Linux distributions.*

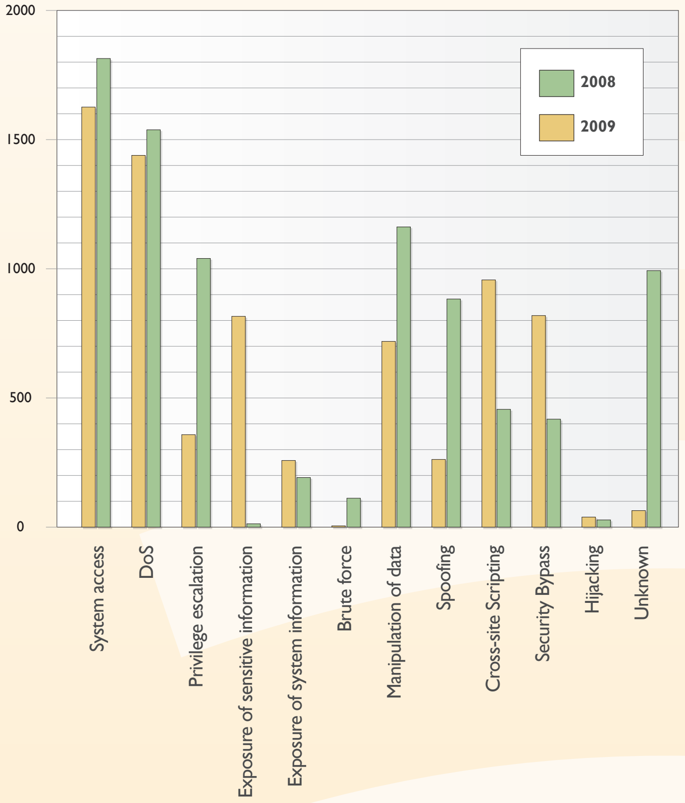
Impact

During the period 2008 - 2009, there has been a significant drop in the number of advisories with system access as the impact, partly due to a change in the type of web application vulnerabilities reported during 2009.

System impact year by year

Type	2008	2009
System access	1,814	1,626
DoS	1,538	1,439
Privilege escalation	1,040	358
Exposure of sensitive information	13	816
Exposure of system information	192	258
Brute force	112	5
Manipulation of data	1,162	719
Spoofing	883	262
Cross-site Scripting	456	957
Security Bypass	418	819
Hijacking	28	39
Unknown	993	64

System impact year by year



Criticality

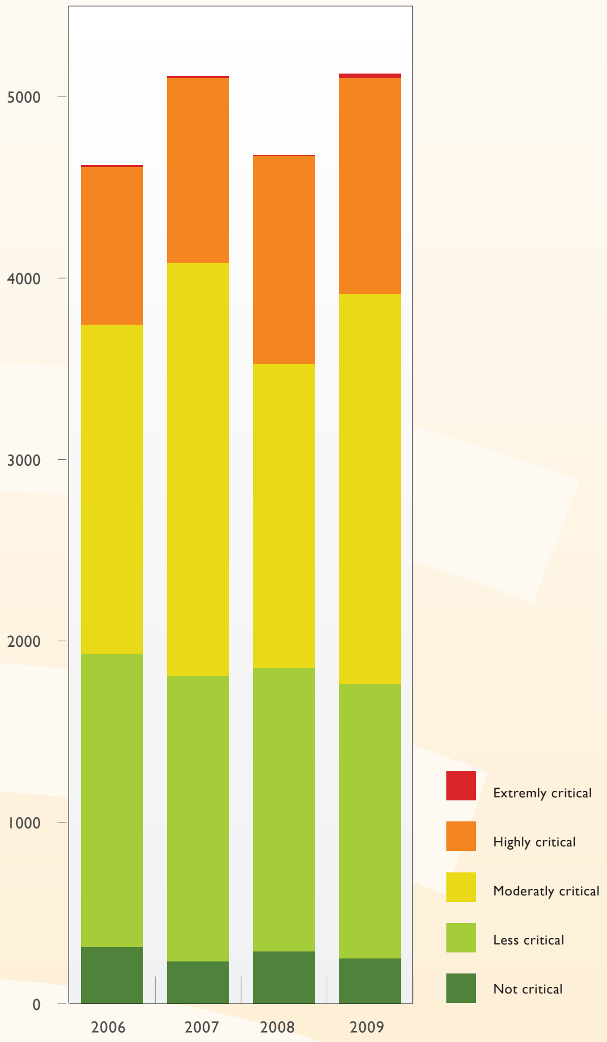
The most noteworthy change in the criticality ratings is the significantly fewer highly critical vulnerabilities in 2009 as compared to the period 2006-2008.

The change in "Highly Critical" advisories can also be seen in the change of system impact.

Criticality ratings year by year

Type	2006	2007	2008	2009
Extremely critical	24	2	11	11
Highly critical	1,191	1,149	1,019	869
Moderately critical	2,152	1,675	2,275	1,814
Less critical	1,511	1,562	1,576	1,615
Not critical	250	290	233	315

Criticality ratings year by year



Appendix A

Criticality rating system

Extremely Critical (5 of 5)

Typically used for remotely exploitable vulnerabilities that can lead to system compromise. Successful exploitation does not normally require any interaction and exploits are in the wild.

These vulnerabilities can exist in services like FTP, HTTP, and SMTP or in certain client systems like email programs or browsers.

Highly Critical (4 of 5)

Typically used for remotely exploitable vulnerabilities that can lead to system compromise. Successful exploitation does not normally require any interaction but there are no known exploits available at the time of disclosure.

Such vulnerabilities can exist in services like FTP, HTTP, and SMTP or in client systems like email programs or browsers.

Moderately Critical (3 of 5)

Typically used for remotely exploitable Denial of Service vulnerabilities against services like FTP, HTTP, and SMTP, and for vulnerabilities that allow system compromises but require user interaction.

This rating is also used for vulnerabilities allowing system compromise on LANs in services like SMB, RPC, NFS, LPD and similar services that are not intended for use over the Internet.

Less Critical (2 of 5)

Typically used for cross-site scripting vulnerabilities and privilege escalation vulnerabilities.

This rating is also used for vulnerabilities allowing exposure of sensitive data to local users.

Not Critical (1 of 5)

Typically used for very limited privilege escalation vulnerabilities and locally exploitable Denial of Service vulnerabilities.

This rating is also used for non-sensitive system information disclosure vulnerabilities (e.g. remote disclosure of installation path of programs).

Appendix B

Impact rating system

Brute force

Used in cases where an application or algorithm allows an attacker to guess passwords in an easy manner.

Cross-Site Scripting

Cross-Site Scripting vulnerabilities allow a third party to manipulate the content or behaviour of a web application in a user's browser, without compromising the underlying system.

Different Cross-Site Scripting related vulnerabilities are also classified under this category, including "script insertion" and "cross-site request forgery".

Cross-Site Scripting vulnerabilities are often used against specific users of a website to steal their credentials or to conduct spoofing attacks.

DoS (Denial of Service)

This includes vulnerabilities ranging from excessive resource consumption (e.g. causing a system to use a lot of memory) to crashing an application or an entire system.

Exposure of sensitive information

Vulnerabilities where documents or credentials are leaked or can be revealed either locally or from remote.

Exposure of system information

Vulnerabilities where excessive information about the system (e.g. version numbers, running services, installation paths, and similar) are exposed and can be revealed from remote and in some cases locally.

Hijacking

This covers vulnerabilities where a user session or a communication channel can be taken over by other users or remote attackers.

Manipulation of data

This includes vulnerabilities where a user or a remote attacker can manipulate local data on a system, but not necessarily be able to gain escalated privileges or system access.

The most frequent type of vulnerabilities with this impact are SQL-injection vulnerabilities, where a malicious user or person can manipulate SQL queries.

Privilege escalation

This covers vulnerabilities where a user is able to conduct certain tasks with the privileges of other users or administrative users.

This typically includes cases where a local user on a client or server system can gain access to the administrator or root account thus taking full control of the system.

Security Bypass

This covers vulnerabilities or security issues where malicious users or people can bypass certain security mechanisms of the application.

The actual impact varies significantly depending on the design and purpose of the affected application.

Spoofing

This covers various vulnerabilities where it is possible for malicious users or people to impersonate other users or systems.

System access

This covers vulnerabilities where malicious people are able to gain system access and execute arbitrary code with the privileges of a local user.

Unknown

Covers various weaknesses, security issues, and vulnerabilities not covered by the other impact types, or where the impact isn't known due to insufficient information from vendors and researchers.