

# THE SECURITY EXPOSURE OF SOFTWARE PORTFOLIOS

*An empirical analysis of the patching challenge faced  
by the average private user*

In this paper, we examine the software portfolio of the average user based on empirical data from over two million users frequently scanning their systems with Secunias Personal Software Inspector (PSI). Our analysis demonstrates, that the complexity and frequency of the actions required to keep a typical end-user system secure, most likely exceeds what users are willing, and able to invest. We classify typical software portfolios, and provide a tool for the community to estimate their individual security exposure and patch complexity. Our data demonstrates the need for coordinated patch solutions capable to update programs from diverse vendors, in order to protect end-users at global scale.

Stefan Frei, Research Analyst Director, Secunia

Thomas Kristensen, Chief Security Officer, Secunia

# THE SECURITY EXPOSURE OF SOFTWARE PORTFOLIOS

*An empirical analysis of the patching challenge faced by the average private user*

## I. INTRODUCTION

One of the most significant changes over the past few decades, has been the rise of information technology and security as important, integral parts of everyday economic and social life (1). World Internet usage has grown 362% to an estimated 1.6 billion users since 2000 (2), and networking has evolved from dedicated point to point connections to ubiquitous communication between people, platforms, and applications. This also means that the security of information technology is affected by a wide variety of actors and processes, which together make up a security ecosystem (3). In the last decade the number of targets, revenue per target, and type of exploitation has also evolved rapidly. At the same time the complexity and variety of software found on a typical host has generally increased, which implies more vulnerabilities are present and continued patching is needed. Classic service-centric vulnerability exploitation required attackers to scan for, and remotely connect to vulnerable hosts (typically servers) in order to exploit them. Unlike these, vulnerabilities on end-user systems are commonly exploited when the user of the vulnerable host visits a malicious Web site, or opens any type of content, or documents, with one of the numerous programs and plug-in technologies installed on his/her host. Profit motivated cyber-criminals have rapidly adopted end-user host exploitation, as a key vector for malware installation and distribution. The variety and prevalence of program portfolios found on typical end user systems, paired with the unpredictable usage patterns of users, makes them a formidable attack vector.

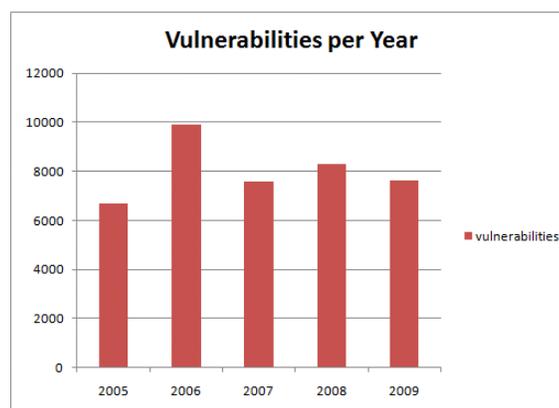


Figure 1: Number of vulnerabilities per year based on Secunia Advisories and vulnerability counts. Source: (4)

The continued identification, communication, and remediation of vulnerabilities in a user's host is a critical path to minimize the users' exposure to successful attacks. Whenever a new vulnerability is discovered, various parties with different, and often conflicting motives and incentives, become

engaged in complex ways. The processes and interactions driven by the continuous discovery of new vulnerabilities drive the constant need of the software users (private and businesses alike) for vulnerability intelligence and patches. It is important to note that the number of vulnerability discoveries has not declined over the last decade, as shown in Figure 1.

Despite massive efforts of the security and software industry, vulnerabilities in software are persistent across all types of software and vendors. Thus, with the continued fixing of vulnerabilities of vendors' software, patching has become a routine and necessity.

In this paper, we attempt to estimate the end-users' security exposure and the complexity of the task of keeping their systems secure. Our analysis is based on empirical data gathered by over two million users of Secunia's Personal Software Inspector (PSI) (5), which is the only highly accurate vulnerability scanner and updating tool for home users.

In a recent study ETH Zurich and Google (6) examined the global security exposure of Web browsers and the users' performance to apply available patches. In this paper we extend this research beyond Web browsers and analyze the end users' software portfolio, based on Secunia PSI and Secunia's vulnerability intelligence database.

## II. SECUNIA PSI

The main data sources for this analysis are scan results from Secunia PSI paired with Secunia's extensive vulnerability intelligence database. Secunia PSI works by examining files on the user's computer (primarily .exe, .dll, and .ocx files). These files only contain non-user specific meta information provided by the software vendor. After examining all the files on local hard drive(s), the collected data is matched against Secunia's file signatures engine to determine the exact version of the programs installed. This provides accurate information on the installation base of the users' host - that is, the user's software portfolio. Figure 2 shows a screenshot of Secunia PSI after a full scan of one of the authors test machines.



Figure 2: Secunia PSI screenshot of one of the authors test machines

Secunia PSI was first released in 2007, and supports all recent Windows operating systems. As of January 2010 more than two million users have installed Secunia PSI. Figure 3 shows the prevalence of the host operating systems as seen by all Secunia PSI scans of the last week of January 2010. Note the high share of Windows 7, released in September 2009, compared to the share of its predecessor, Windows Vista, released in January 2007.

A first full scan of the host is run after the initial installation of Secunia PSI, thereafter it is configured to run once a week (or every time it is found not to have run for more than six days), or the user can start a full scan when needed.

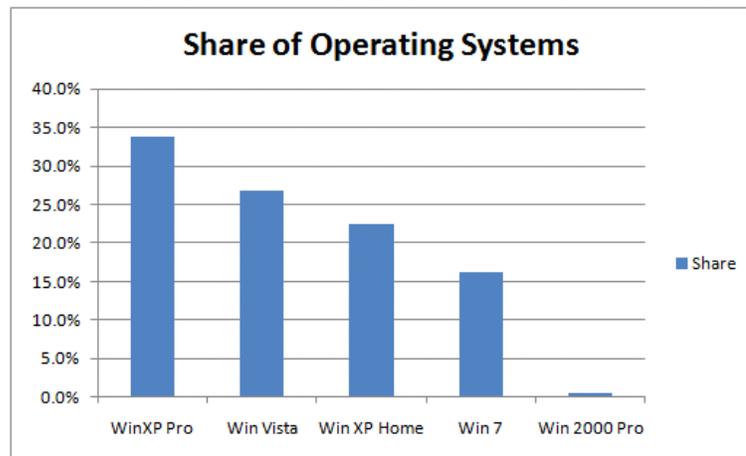


Figure 3: Share of operating systems as seen by Secunia PSI scans in the last week of January 2010. Note the high share of Windows 7 (released September 2009) compared to Windows Vista (released January 2007)

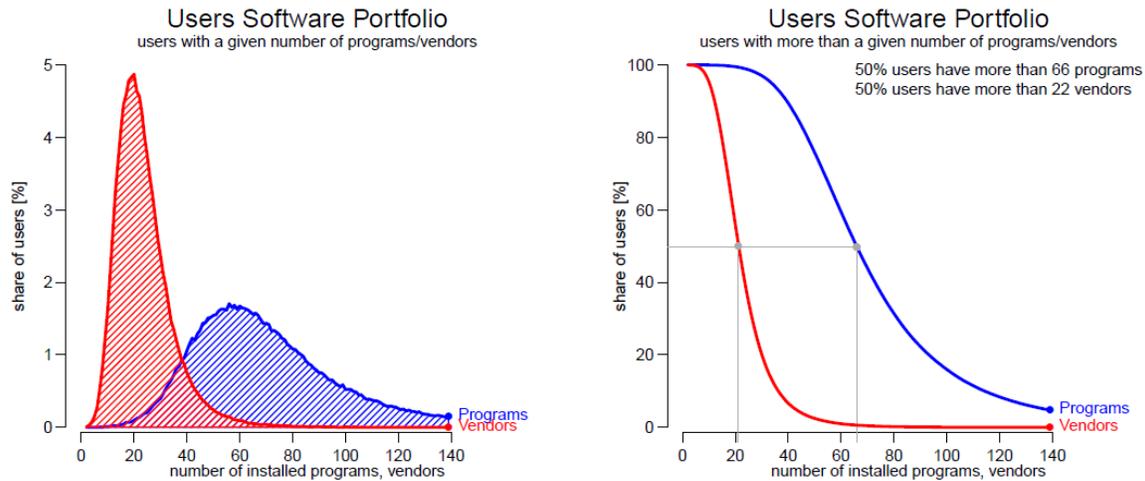
The security of an end-user host is a function of the number and kinds of software installed on it. With the empirical data from Secunia PSI, we determine the software portfolios of all users scanned in the last week of January 2010. This data provides formidable insight on the composition and distribution of programs installed, and their exact patch level. We apply statistical analysis to capture the main properties of this extensive set of data. The correlation of software portfolio information with vulnerability intelligence covering the last 12 month provides insight into the security exposure of a typical user.

### III. TYPICAL SOFTWARE PORTFOLIOS

#### 1. Vendors and Programs

We first examine the users' software portfolio; that is, the number of programs, or vendors, PSI typically finds on a scanned host. As a first order approximation, we consider the number of different vendors in the software portfolio of the end-user to be a minimum estimator for the number of different patch mechanisms needed to keep a system up-to-date. In Figure 4 we plot the distribution of the percentage of users found to have a given number of programs, or vendors, in their software portfolio. The left pane in Figure 4 shows a peak in the share of users at 56 programs, and at 20 vendors. To determine the share of users *with more than* a given number of programs, or

vendors, in their portfolio, we plot the cumulative distribution function (CDF)<sup>1</sup> in the right pane of Figure 5. Here we readily see that 50% of the users have *more than* 66 programs from *more than* 22 vendors in their software portfolio. These numbers give a first insight into the complexity of the task to keep a typical software portfolio up to date. As vendors do not typically share their update mechanisms, 50% of the end-users have to familiarize themselves with at least 22 different update mechanisms to keep their systems secure.

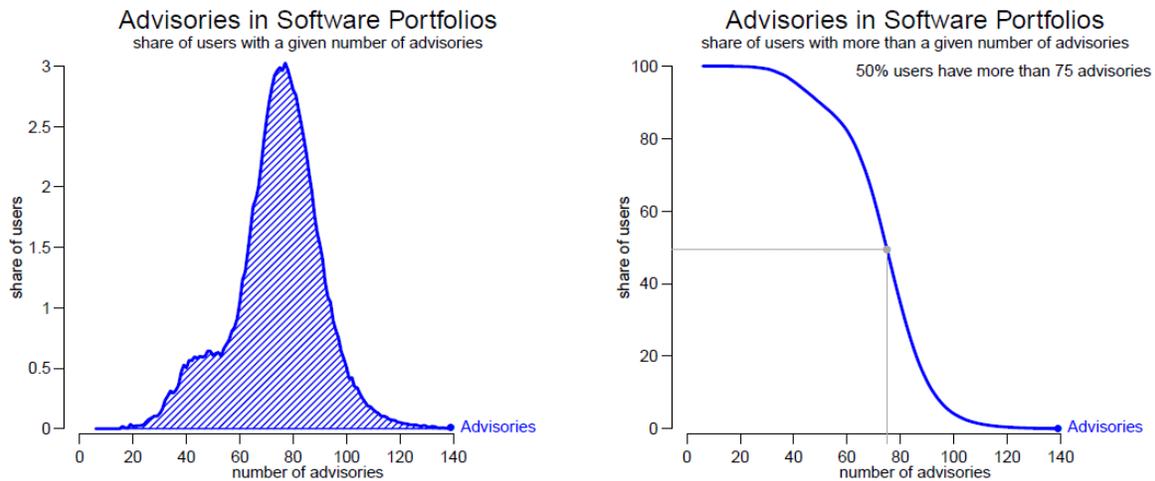


**Figure 4:** Distribution of the number of programs, or vendors, found in the software portfolio of end-users. The left pane depicts the percentage of users with a given number of programs, or vendors, in their software portfolio. The right pane shows the percentage of users *with more than* a given number of programs, or vendors, in their software portfolio. Source: Secunia PSI

## 2. Security Advisories

The complexity of the end-user task to keep a system secure is a function of the number of different patch mechanisms needed, and the frequency of how often updates have to be installed. We estimate the number of patch actions required based on the number of Secunia Advisories released in a given period of time. Typically, a Secunia Advisory translates into a patch action needed to fix the vulnerability (or vulnerabilities) reported in that advisory. Given an end-user's software portfolio, we can now count all Secunia Advisories released in the last 12 month affecting any of the programs in this portfolio. In Figure 5 we plot the share of users whose software portfolios were affected by a given number of Secunia Advisories in the last 12 month. From Figure 5, in the right hand pane, we see that 50% of the end-user software portfolios are affected by *more than* 75 Secunia Advisories per year.

<sup>1</sup> Technically, we plot the survival function  $P(X > x)$



**Figure 5: Share of users whose software portfolio was affected by a given number of Secunia Advisories over the past 12 month.**

From the defenders and end-user ergonomics perspective, we are interested in estimating the number of different patch mechanisms, and the number of patch actions required in a given year, to keep the host secure. Our results from Figure 4 and Figure 5 show that 50% of the users have software portfolios with more than 22 different vendors that are affected by 75 or more Secunia Advisories per year. Thus, a typical end-user has the daunting task to administer his host approximately 75 times a year (or every 4.8 days), thereby handling approximately 22 different update mechanisms to keep his/her system secure.

## IV. PORTFOLIO SECURITY EXPOSURE

### 1. Vulnerabilities in Software Portfolios

From an attacker's perspective we are interested in estimating the security exposure of a typical end-user, given his/her software portfolio. Thus, for all software portfolios found by Secunia PSI we identified and counted all relevant vulnerabilities disclosed in the last 12 month. In Figure 6 we plot the user's exposure to vulnerabilities, given his/her software portfolio. We find that 50% of the users are typically exposed to more than 297 security vulnerabilities over a year.

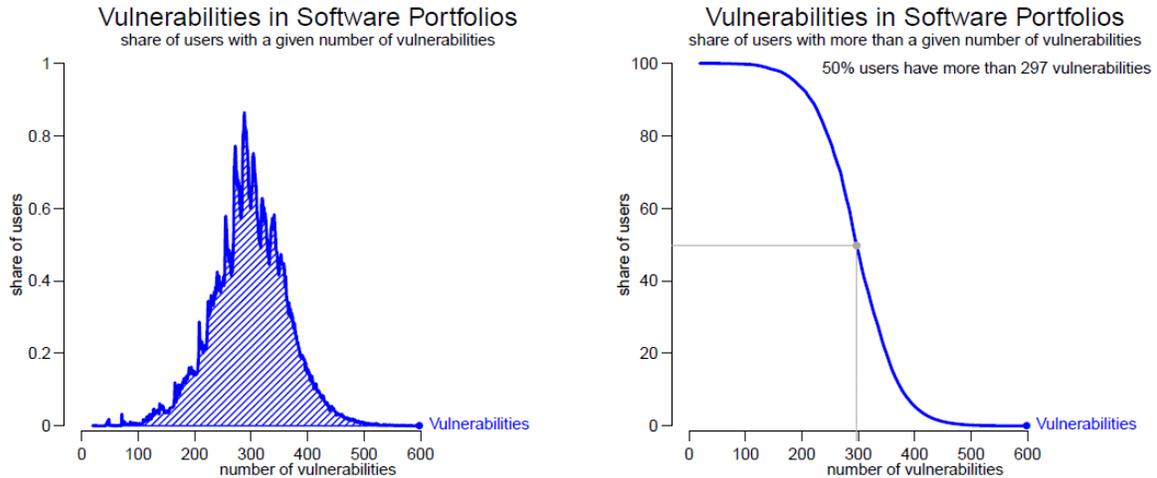


Figure 6: Share of users whose software portfolio was affected by a given number of vulnerabilities over the course of a year.

## 2. Estimate your security exposure

The previous plots provided insight into the distribution of typical software portfolios, based on empirical data of a large user base. To provide a tool for the security community to assess the security exposure of a given portfolio, we now study how the average number of vulnerabilities and Secunia Advisories evolve, given the number of programs in the software portfolio. This helps to shed light on questions such as “How many vulnerabilities do I expect to have if my software portfolio consists of 50 products, or 80 products?”

To provide such a tool we take again a snapshot of all scans done in the last week of January 2010, and group the identified software portfolios according to the number of programs they contain:

- Group  $G[x]$ : { set of all software portfolios with exactly  $x$  products }

For each such group we then calculate the average number of vendors; and for the last 12 months we calculate the average number Secunia Advisories, and vulnerabilities:

- $vendors[x]$ : average number of vendors in  $G[x]$
- $advvs[x]$ : average number of Secunia Advisories in the last 12 months in  $G[x]$
- $vuln[x]$ : average number of vulnerabilities in the last 12 month in  $G[x]$

The result is shown in Figure 7, where the x-axis depicts the number of programs. From this plot we can read the expected average number of vulnerabilities given a software portfolio with  $x$  programs. The average number of *vendors*, *Secunia Advisories*, and *vulnerabilities* can be read from the y-axis on the left of Figure 7. We also plot the curves of one standard deviation from the average of the vulnerability count. For example, as marked in Figure 8, we see that a software portfolio of 89 products would on average lead to 79 patch actions and 321 vulnerabilities in a year.

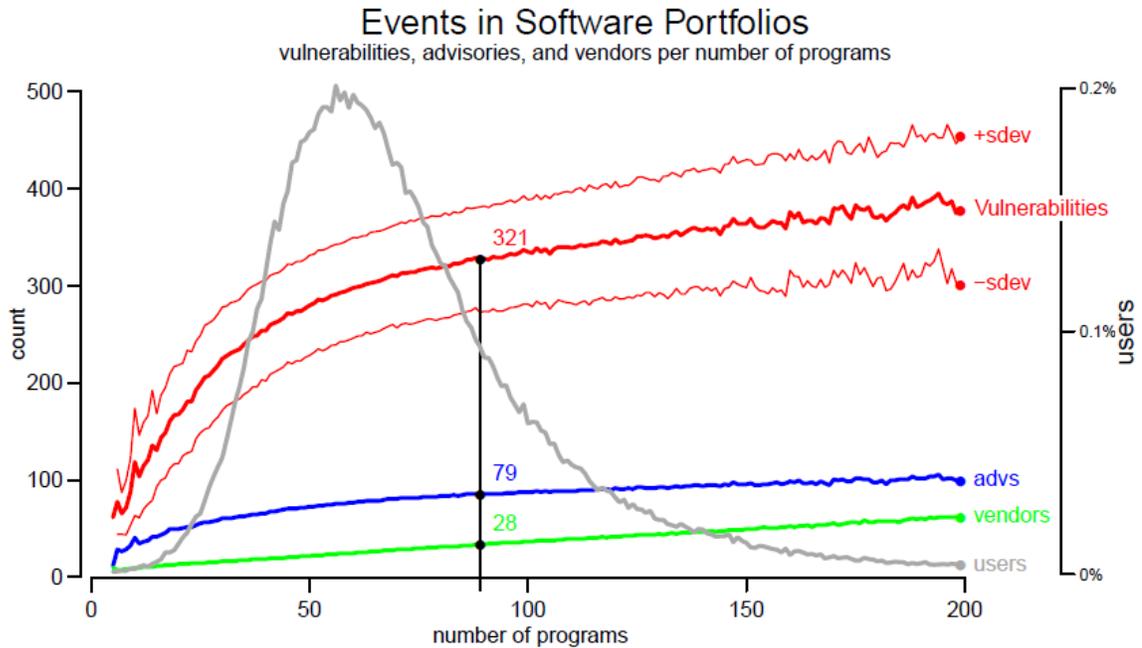


Figure 7: Typical evolution of the number of vendors, Secunia Advisories, and vulnerabilities given a set number of programs in the software portfolio

To indicate the prevalence of software portfolios with a given number of products, we also include the share of users of such portfolios. This share refers to the y-axis on the right in Figure 8. As already seen in Figure 4, we find the largest share of users at around 56 programs. To find representative values that demonstrate the patching challenge of end-users, we look at a set of software portfolios covering a given percentage of all users. In Figure 8 we plot typical ranges of software portfolios representing 30%, 50%, 70%, and 90% of the end-users. We determine the values representing the patching challenge for these user shares from Figure 7, and summarize them in Table 1.

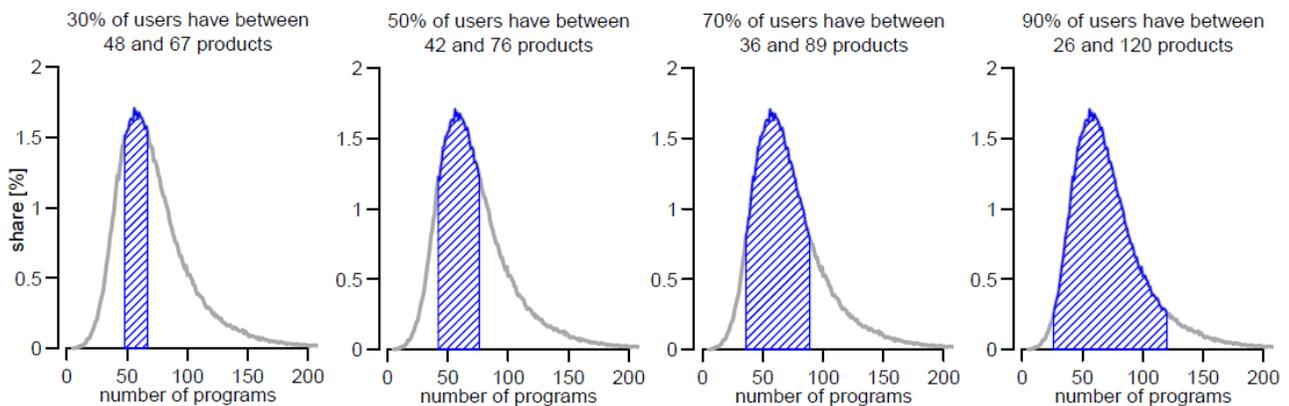


Figure 8: Typical number of programs in the software portfolios covering 30%, 50%, 70%, and 90% of the users. Details are listed in Table 1.

Users	Programs	Vendors	For the last 12 month	
			Patch Actions	Vulnerabilities
30%	48 – 67	16 – 22	66 – 74	268 – 301
50%	42 – 76	14 – 24	62 – 77	256 – 311
70%	36 – 89	12 – 28	56 – 79	238 – 321
90%	26 – 120	9 – 36	51 – 86	200 – 342

**Table 1: Expected number of programs, vendors, patch actions, and vulnerabilities for a given percentage of users as of Figure 7 and Figure 8.**

Our analysis reveals that 90% of the users have to handle on average between 51 and 86 patch actions per year in order to address between 200 and 342 vulnerabilities affecting the programs of 9 to 36 vendors in their software portfolios.

## V. CONCLUSION & IMPLICATION

The complexity of the task to simply keep a home system up-to-date clearly shows the need for accurate vulnerability intelligence; and for tools to help identifying and patching all these programs. Our analysis demonstrates, that the total effort, and the frequency of actions, required to keep an end-user system secure most likely exceeds what the typical user is able, or willing, to invest into the security. As with back-ups, if the process is not fully automated and monitored, it is almost certain to fail when most needed – with dire consequences. Unlike back-ups, we still lack the technology, processes, or a common standard to facilitate the automated patching of diverse programs, across different vendors, at global scale. Major software vendors could afford the development and continued operation of state-of-the-art update processes. However, the increasing number of 3<sup>rd</sup> party programs, plug-ins, and technologies creates new challenges.

Given the increase in e-crime and the subsequent quest to find new ways to compromise end-user systems, it is no surprise that criminals have changed their primary focus from Microsoft programs to 3<sup>rd</sup> party programs; and Adobe programs in particular due to the prevalence of these on end-user systems. Microsoft Security Intelligence Report Vol. 7 (7) claims that 85% of all compromises of Vista systems exploited vulnerabilities in 3<sup>rd</sup> party programs, and a recent report from ScanSafe (8) claims that 80% of all web-based exploits in Q4 2009 were "utilizing" PDF documents.

The MS SIR Vol. 7 numbers are likely to be "best case" scenarios due to a lack of signatures for detecting exploitation of 3<sup>rd</sup> party programs. Microsoft is the only vendor to provide signature guidance for vulnerabilities in their products via Microsoft Active Protections Program (MAPP) (9), a new program for security software providers. Members of MAPP receive security vulnerability information from the Microsoft Security Response Center (MSRC) in advance of Microsoft’s monthly security update. No other software vendors provide this sort of technical details for creating signatures.

As a result, the security vendors providing reactive signature based protection like anti-virus, IDS, and IPS only focus on creating signatures for vulnerabilities in Microsoft programs and a few other high profile products.

## BIBLIOGRAPHY

1. **Oborne, Michael W.** *The Security Economy*. Paris : OECD, Paris , 2004. ISBN 92-64-10772-X.
2. **Internet World Statistics.** [Online] <http://www.internetworldstats.com>.
3. **Frei, Stefan.** *Security Econometrics - The Dynamics of (In)Security*. ETH Zurich. s.l. : Amazon, ISBN 1-4392-5409-5, 2009. PhD Thesis. ETH Zurich, Dissertation 18197.
4. **Secunia.** Vulnerability Intelligence Provider. [Online] <http://www.secunia.com>.
5. **Secunia (PSI).** Personal Software Inspector. [Online] [http://secunia.com/vulnerability\\_scanning/personal/](http://secunia.com/vulnerability_scanning/personal/).
6. **Frei, Stefan, et al.** *Understanding the Web browser threat*. s.l. : ETH Zurich, 2008. <http://www.techzoom.net/insecurity-iceberg>.
7. **Microsoft Security Intelligence Report.** Vol. 7. [Online] 2009. <http://www.microsoft.com/security/about/sir.aspx>.
8. **ScanSafe.** Annual Global Threat Report. [Online] 2009. [http://scansafe.com/agtr\\_2009](http://scansafe.com/agtr_2009).
9. **Microsoft Active Protections Program.** [Online] <http://www.microsoft.com/security/msrc/collaboration/mapp.aspx>.

**Copyright 2010 Secunia. All rights reserved.**

This whitepaper may only be redistributed unedited and unaltered. This whitepaper may be cited and referenced only if clearly crediting Secunia and this whitepaper as the source. Any other reproduction and redistribution in print or electronically is strictly prohibited without explicit permission.