Chapter 6
Modeling the Security Ecosystem - The Dynamics of (In)Security

Stefan Frei, Dominik Schatzmann, Bernhard Plattner, Brian Trammell

Abstract The security of information technology and computer networks is effected by a wide variety of actors and processes which together make up a security ecosystem; here we examine this ecosystem, consolidating many aspects of security that have hitherto been discussed only separately. First, we analyze the roles of the major actors within this ecosystem and the processes they participate in, and the the paths vulnerability data take through the ecosystem and the impact of each of these on security risk. Then, based on a quantitative examination of 27,000 vulnerabilities disclosed over the past decade and taken from publicly available data sources, we quantify the systematic gap between exploit and patch availability. We provide the first examination of the impact and the risks associated with this gap on the ecosystem as a whole. Our analysis provides a metric for the success of the "responsible disclosure" process. We measure the prevalence of the commercial markets for vulnerability information and highlight the role of security information providers (SIP), which function as the "free press" of the ecosystem.

6.1 Introduction

With the ongoing deployment of information technology in today's economy and society, comprehending the evolution of information security at large has become much more than the mere understanding of the underlying technologies. There is

Stefan Frei
Communication Systems Group, ETH Zurich, e-mail: frei@techzoom.net

Dominik Schatzmann
Communication Systems Group, ETH Zurich, e-mail: schatzmann@tik.ee.ethz.ch

Bernhard Plattner
Communication Systems Group, ETH Zurich, e-mail: plattner@tik.ee.ethz.ch

Brian Trammell
Hitachi Europe, ICTL Secure Systems Team, Zurich e-mail: trammell@tik.ee.ethz.ch

T. Moore et al. (eds.), *Economics of Information Security and Privacy,*
DOI 10.1007/978-1-4419-6967-5_6,

a growing realization that security failures are caused as often by bad incentives as by bad design or neglected implementation: Insecurity often results from what economists call an *externality*, a side-effect of using information technology, like environmental pollution [2]. E.g. vulnerabilities in software impose costs on the whole society of users, while software vendors get all the profits. Whenever a new vulnerability is discovered, various parties with different and often conflicting motives and incentives become engaged in a complex way. These players and their interactions form what we call the *Security Ecosystem*. The security impact resulting from the interplay of the actors of the security ecosystem cannot be understood and managed unless we can better measure these risks. The goal of this paper is to develop metrics that help to obtain a better understanding of the state and the evolution of today's security environment from a global perspective. Our method to give insight into the dynamics and the prevalence of important processes of the security ecosystem is the analysis of the *Lifecycle of a Vulnerability*, based entirely on publicly available data from various sources. In the following we define the lifecycle of a vulnerability and introduce a model to describe the main players and their interactions in the security ecosystem. The sequence of events in the vulnerability lifecycle measures the main processes governing the security ecosystem. To support the understanding of these complex processes we revisit the key elements of the "disclosure debate", look at "vulnerability markets", and analyze the motivations of vendors and cyber-criminals. Finally we show how the security ecosystem can be described and analyzed quantitatively using statistical analysis of the vulnerability lifecycle.

6.2 Related Work

After years of providing more and more security features, a realization emerged that a pure technical point of view is not sufficient to understand the ever evolving security landscape [2]. According to [34], the *security ecosystem* describes the activities of creating, preventing, dealing with, and mitigating insecurity in the use of information technology. The economics of information security is *cross-disciplinary* as much as *interdisciplinary* according to Pfleeger [39]. Quantitative measurements of the security ecosystem typically focused on partial analysis of individual events. In "The new school of information security" Shostack and Stewart observe that until today there exist no aggregated long-term indicators or indexes to better understand how the security ecosystem functions [47]. Research on the economic consequences of cyber attacks has been dealing primarily with microanalysis of specific events, technologies or targeted organizations [39]. In 2004, Cavusoglu and Arora examine how a disclosure policy affects the time for a vendor to release a patch [5, 16]. Kannan and Telang study whether market-based mechanism for vulnerability disclosure lead to a better social outcome [22]. The lure of money is changing the computer security playing field, and we must reexamine our assumptions in the face of financially motivated attackers. In 2004 Thomas et al. highlight that fraud is likely to

be as prevalent in the online environment as in the conventional environment [51] and Maillart et al demonstrated in 2008 that the largest possible ID losses per event grow faster-than-linearly [27]. The convergence of criminals and technically savvy crackers is on the way [25].

6.3 Methodology

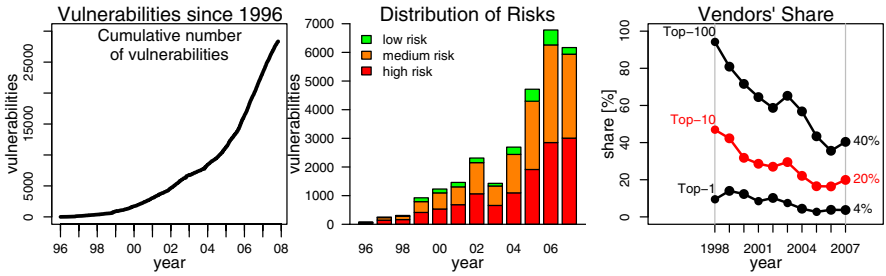


Fig. 6.1 Vulnerability disclosures 1996-2007 and share of the top-N vendors with the most vulnerabilities.

In this research, we analyze the state and the evolution of the security ecosystem over the last twelve years based on an empirical dataset of more than 27,000 vulnerabilities disclosed between 1996 and 2008. We examine the prevalence of different sequences of events in the vulnerability lifecycle for a large set of vulnerabilities, normalized to the time of vulnerability disclosure. Normalization with respect to the time of disclosure is an obvious approach as this is the first point in time the vulnerability becomes known to the public. To create a comprehensive vulnerability database we download, parse, and correlate the information of well over 200,000 individual security bulletins of various sources. Due to the inaccessibility, privacy or unavailability of data, only certain aspects of the security ecosystem can be measured from the outside. It is unlikely that cyber-criminals will ever share data about their operations, and software manufacturers are reluctant to publish data about their internal vulnerability handling processes. The data for this research is gathered exclusively from publicly available sources.

Phase 1 - Data Collection We do not attempt to take all possible information sources into consideration, rather than being exhaustive we choose a set of sources based on criteria such as independence, accessibility, and available history of information. Thus, we processed all security advisories from *US-CERT* [53], *SecurityFocus* [49], *IBM ISS X-Force* [19], *Secunia* [43], *Vupen* [15], *SecurityTracker* [44], *iDefense's* (VCP) [21], and *TippingPoints* (ZDI) [52]. For exploit information we analyzed *Milw0rm* [31], *Packetstorm* [1], *SecurityVulns* [45], and *Metasploit* [17]. Finally we imported the content of the National Vulnerability Database (NVD), the Open Source Vulnerability Database (OSVDB) [37], and the CVE database [33].

Phase 2 - Parsing We processed the data gathered in Phase 1 to extract the *date of publication*, all *CVE identifiers* and all *cross references (URLs)* to other security sources. From the NVD we derive the mapping of vulnerability to vendor/product name and risk rating (high, medium, low). This information is fed into our vulnerability database.

Phase 3 - Data Correlation In the database we correlate the raw data collected in the previous phases. CVE identifiers are used for the correlation of vulnerability information from different sources. To capture cases where the CVE identifier is missing in an advisory, we used cross references in NVD and CVE documents (where a CVE is always assigned by definition). The output of this step is a set of unique vulnerabilities identified by their CVE identifier and a set of related advisories from different sources providing the specific vulnerability lifecycle data.

Vulnerability Data Before we proceed with the analysis, we look at the total number of vulnerabilities in our database and their distribution among vendors and risk classes. In Fig. 6.1 left we plot the cumulative number of vulnerabilities disclosed since 1996 and in the center we plot the number of disclosures by year and risk rating. The information plotted is based on the content of our vulnerability database. Consistently, most vulnerabilities are classified as either "high" or "medium" risk, and up to 2006 we see a steady increase in the number of vulnerabilities disclosed per year. The distribution of these vulnerabilities among the affected vendors is depicted in Fig. 6.1 (right), and Fig. 6.2. Only a few vendors account for most vulnerabilities published in a given year and we observe a skewed distribution similar to a power law distribution. This fact is shown in Fig. 6.1 (right) where we plot the combined share of the *top-N* vendors (affected by vulnerabilities) per year since 1998 for $N \in \{1, 10, 100\}$. E.g. only $N = 10$ (or 0.04%) of the 2,491 vendors of vulnerable software in 2007 are responsible for 20% of the reported vulnerabilities in that year. Fig. 6.2 lists the names of the *top-10* vendors from 2002 to 2007. From this analysis we observe that most of the vulnerabilities published in any given year affect well known commercial and open-source software vendors. These vendors produce the majority of software products in daily use at home and within business. As a result most of the vulnerabilities disclosed are of relevance to the majority of users.

6.4 Vulnerability Lifecycle

Our method to give insight into the dynamics of the security ecosystem is the analysis of the vulnerability lifecycle shown in Fig. 6.3. The sequence of events in the vulnerability lifecycle is used to measure the main processes governing the security ecosystem. We first define what we consider to be a security vulnerability and introduce the events of the vulnerability lifecycle followed by the identification of specific risk exposure phases defined by the sequence of these events.

What is a Vulnerability? The lifecycle of a vulnerability cannot be modeled without a precise definition of the term *vulnerability*. However, defining vulnerabilities is

a delicate undertaking that depends significantly on the parties involved and their intent. For example, whether a specific software flaw is considered *a defect*, *a feature*, or *a vulnerability* differs whether you talk to a researcher, the vendor, or different users of the software. In the field of information security, many competing definitions of a vulnerability have been proposed [26, 38]. As we are mainly interested in accurately reflecting the processes of the security ecosystem, we delegate the decision on what counts as a vulnerability to the Common Vulnerabilities and Exposures (CVE) consortium [33]. CVE is a *de facto* industry standard that has achieved wide acceptance in the security industry, academia, and a number of government organizations since its launch in 1999. According to CVE, a vulnerability is a mistake in software that can be directly used by an attacker to gain access to a system or network [32]. For this research, we consider only vulnerabilities listed in the CVE database, thereby delegating the decision on what counts as a vulnerability to the CVE editorial board:

Definition 6.1. For this research, only a security issue with an assigned CVE identifier is considered a **vulnerability**.

This definition explicitly does not try to define technical properties of security issues, as we are interested in capturing the real-world impact of security issues in order to shed light on the processes of the security ecosystem. Given the high acceptance of the CVE process in academia and industry we assume that any security issue *of relevance* will eventually get a CVE number assigned.

2002	2003	2004	2005	2006	2007
Microsoft	Microsoft	Microsoft	Microsoft	Microsoft	Microsoft
Cisco	Sun	Gentoo	Apple	Apple	Apple
HP	Apple	Red Hat	Linux	Oracle	IBM
Sun	IBM	Apple	Mozilla	Mozilla	Oracle
Oracle	Red Hat	Linux	Sun	Linux	PHP
IBM	SGI	SuSE	IBM	IBM	Sun
SGI	HP	SGI	Oracle	Sun	Cisco
Apache	Apache	Mozilla	Red Hat	Cisco	Mozilla
FreeBSD	Cisco	Mandrake	Ethereal	Joomla	HP
Mozilla	Linux	Sun	SuSE	Novell	Linux

Fig. 6.2 List of the top-10 vendors by number of vulnerabilities in their products. Source: NVD

Vulnerability Lifecycle Events The lifecycle of a vulnerability $v \in V$ (with V denoting the set of vulnerabilities listed by CVE) can be divided into phases between distinctive events. Each phase reflects a specific state of the vulnerability and an associated risk exposure for the users of the software affected. To capture these phases we define the events *creation*, *discovery*, *exploit availability*, *disclosure*, *patch availability*, and *patch installation* for each vulnerability, as shown in Fig. 6.3. With some restrictions, the exact sequence of these events varies among individual vulnerabilities.

Time of creation (t_{creat}) Vulnerabilities are typically created by accident as the result of a coding mistake, often involving the mismanagement of memory. If a

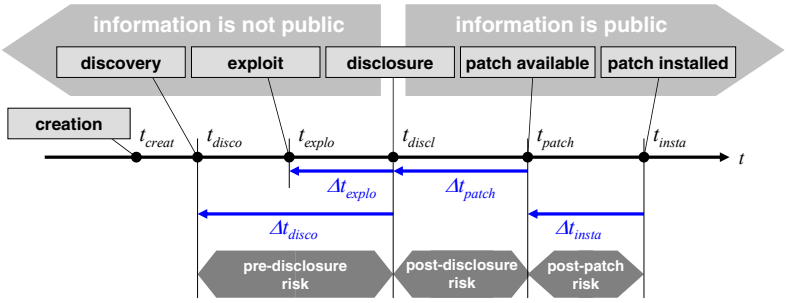


Fig. 6.3 The lifecycle of a vulnerability defined by distinctive events. The exact sequence of events varies between vulnerabilities.

vulnerability remains undetected in the code throughout the development and testing phases, chances are it will make it into generally available code that is then released [18]. In this research we consider only vulnerabilities discovered *after* the release of the software. The time of vulnerability creation is typically unknown by definition, however it may be determined in retrospect, after the discovery or disclosure of the vulnerability. If the creation of a vulnerability is malicious and thus intentional, discovery and creation time coincide [3]. In this paper we do not further investigate the time of vulnerability creation.

Time of discovery (t_{disco}) The *time of discovery* is the earliest time a software vulnerability is recognized to pose a security risk. Vulnerabilities do exist before they are discovered, but prior to the discovery of the vulnerability the underlying defect is not recognized to pose a security risk. Usually the time of discovery of a vulnerability is not publicly known until *after* its disclosure.

Time of exploit availability (t_{explo}) An exploit is a piece of software, a virus, a set of data, or sequence of commands that takes advantage of a vulnerability in order to cause unintended or unanticipated behavior to occur in software or an embedded device. Proof-of-concept code or exploits provided within security research and analysis tools are also deemed exploits[1]. Typically, it is a trivial exercise for criminals to turn such code into a working exploit. The *time of exploit* is the earliest time an exploit for a vulnerability is available.

Time of public disclosure (t_{discl}) The purpose of *disclosure* is to make security information available to the public in a standardized, understandable format. Disclosure is an important event in the security ecosystem. In the literature, definitions of *disclosure* range from ”made public to wider audience”, ”made public through forums or by vendor”, ”reported by CERT or Securityfocus”, or ”made public by anyone before vendor releases a patch” as in [3, 4, 35]. To normalize this set of definitions, we define the disclosure time as follows:

[1] E.g. *Metasploit*, a tool for developing and executing exploit code to aid in penetration testing and IDS signature development.

Definition 6.2. The **time of disclosure** $t_{discl}(v)$ of a vulnerability v is the first time a vulnerability is described on a channel where the *information disclosed* and the *information channel* publishing the vulnerability satisfy the following requirements:

1. *Free Access:* The disclosed vulnerability information is available to the public for free.
2. *Independence:* The vulnerability information is published by a widely accepted and independent source.
3. *Validation:* The vulnerability has undergone analysis by security experts such that risk rating information is included.

These requirements ensure the quality of vulnerability information threefold: From the security perspective only a free and public disclosure of vulnerability information can ensure that all interested, affected, or concerned parties get the relevant security information (*free access*). *Independence* is a prerequisite for unbiased and complete information, while the *validation* requirement builds confidence in the quality of the information delivered. The mere discussion of a potential flaw in a mailing list or vague information from a vendor therefore does not qualify. We call viable sources of vulnerability information *Security Information Providers (SIP)*, which we discuss in detail in Section 6.5. Furthermore, only an information source not dependent on a vendor or government is unbiased and ensures a fair dissemination of security critical information[2]. This implies the use of several sources to determine the time of disclosure, as many of the organizations that publish security information are associated with vendors or governments. In combination, these three requirements ensure that the disclosure date reflects the first time when trusted, widely understandable information about a new vulnerability is publicly available to everyone concerned. Correlation using CVE identifiers allows to handle dissimilar publication dates from diverse sources: The publication date of the first SIP (as listed in the Appendix) reporting a given vulnerability is used as the disclosure date t_{discl} for a vulnerability.

Time of patch availability (t_{patch}) The *time of patch availability* is the earliest time that the vendor releases a patch that provides protection against the exploitation of the vulnerability. Unfortunately, software vendors typically cannot make security patches available instantly after the discovery of new vulnerabilities or exploits. While some vendors publish patches as soon as these are available, others publish patches on a predefined schedule to ease the planning of patch installation (e.g. monthly or quarterly scheduled release of new patches). We analyze the patch release performance of various software vendors in detail in Section 6.6. In many cases a patch may be available before public disclosure (e.g. the DNS vulnerabilities of 2008 and service pack roll-ups for new operating systems).Fixes and patches offered by third parties are not considered as a patch, we deem the vendor as the only authoritative source to provide patches for its software. The complexity of patches varies from simple configuration fixes to extensive changes in the foundation of

[2] In the following of this paper we use the term *vendor* to name the manufacturer of the software for *commercial products*, *freeware*, and *open-source software* alike

the software. Other security mechanisms such as signatures for intrusion prevention systems or anti-virus tools are not considered as patches.

Time of patch installation (t_{insta}) Software users can only benefit from the correction of a vulnerability after a patch is installed on their systems. The processes leading from patch availability to patch installation vary considerably among different kinds of software users. Hence, the time to patch installation is not a specific point in time for a vulnerability, it can only be given as a distribution for a specific population of users (e.g. corporate or home users).

6.4.1 Risk Exposure Times

Between the discovery of a vulnerability and its elimination through the installation of a patch, a system is potentially at risk. This exposure period can be separated into three phases: the "pre-disclosure", the "post-disclosure" and the "post-patch" phase as shown in Fig. 6.3. We analyze the relation and evolution of these periods to distinguish and understand important processes in the security ecosystem.

Pre-disclosure phase (Δt_{disco}) During the time from discovery to disclosure Δt_{disco}, only a unknown group is aware of the vulnerability. This group could be anyone from lone hackers to cyber-criminals likely to misuse their knowledge. On the other hand, this group could also consist of researchers and vendors working together to provide a patch for the identified vulnerability. We call the risk exposure arising from this period as "pre-disclosure" risk because the vulnerability is known to have a security impact whereas the public has no access to this knowledge.

$$\Delta t_{disco}(v) = t_{disco}(v) - t_{discl}(v) \tag{6.1}$$

Post-disclosure phase (Δt_{patch}) During the time from disclosure to patch availability Δt_{patch} the user of the software waits for the vendor to release a patch. We call the risk exposure arising from this period the "post-disclosure" risk because the public is aware of this risk but has not yet received remediation from the software vendor/originator. However, users of the vulnerable software can assess their individual risk and implement a workaround based on the information provided with the disclosure of the vulnerability.

$$\Delta t_{patch}(v) = t_{patch}(v) - t_{discl}(v) \tag{6.2}$$

Post-patch phase (Δt_{insta}) The time from patch availability to patch installation Δt_{insta} is called the "post-patch" risk. The duration of this period is typically under direct control of the user of the affected software or embedded device. Typically, business and private users face different challenges to timely patch installation. Installing a patch or changing security-relevant configuration settings on a mission-critical business system is a non-trivial task for a typical enterprise. Further, we found considerable delays of patch installation timing of end-users' Web browsers

in [10, 12, 13], mostly attributed to the degree of automation available for patch installation. Note that an ever-increasing number of embedded control devices are deployed in support of our networked society, many of which cannot be patched by their users.

$$\Delta t_{insta}(v) = t_{insta}(v) - t_{patch}(v) \tag{6.3}$$

Exogenous vs. Endogenous We designate "pre-disclosure" and "post-disclosure" phases as *exogenous*, since the operator of the vulnerable system cannot exert direct influence on the length of these periods. The length of these phases can only be **influenced on a macro perspective** through the interplay of the processes in the security ecosystem, as shown in Fig. 6.4 and discussed in Section 6.5. Likewise, the nature of the "post-patch" phase is *endogenous* as the operator of the system determines the time when the patch is installed.

6.5 The Security Ecosystem

In this section we introduce and discuss the *major players* and *main processes* in security ecosystem followed by a review of the "disclosure debate" which is central to understand these processes and the incentives. In the last decade, the number of players and their roles and interactions within the security ecosystem have evolved considerably. A variety of legislative and social issues directly influence the processes of vulnerability research, detection, publication, and response. Vendors, developers, customers, cyber-criminals, and the security community have divergent perspectives on the impact of vulnerabilities. The processes and interactions between these actors are driven by the continuous discovery of new vulnerabilities and the subsequent constant need of the public (the software users) for security information and patches. In Fig. 6.4 we model the main processes in the security ecosystem, starting with the discovery of a new vulnerability on top and the public disclosure of vulnerability information at the bottom. The flow of vulnerability information from the discoverer to the public can take several paths, each describing a different process with implications for the resulting risk exposure. The boxes *Discovery*, *Exploit*, *Patch*, and *Disclosure* in our model identify important events in the security ecosystem that can be related to events in the vulnerability lifecycle as introduced in Section 6.4. Examination of the exact sequence of vulnerability lifecycle events for a large sample of vulnerabilities allows us to identify the prevalence of particular processes and the dynamics of the security ecosystem.

6.5.1 Major Players

We start the discussion of the security ecosystem model with the introduction of its major players, namely the *discoverer*, commercial-, and underground *vulnerability*

markets, *cyber-criminals*, the software *vendors*, *security information providers*, and the *public*.

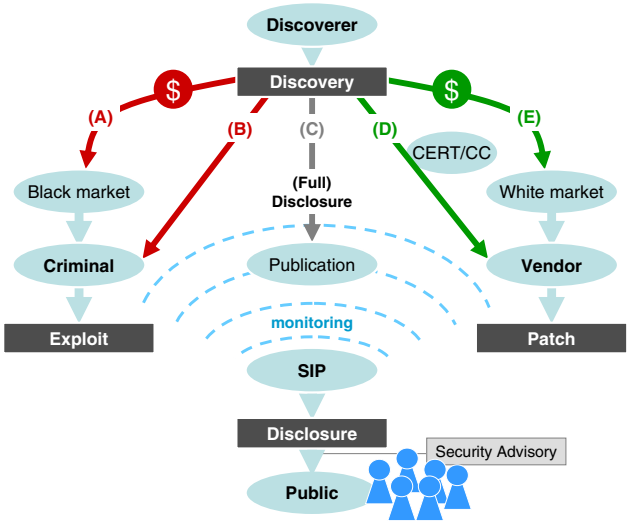


Fig. 6.4 Main processes of the security ecosystem and relation to vulnerability lifecycle events.

6.5.1.1 Discoverer

The *discoverer* of a vulnerability is an individual or organization (e.g. the vendor, independent researcher, cyber-criminal, government agency) that discovers a new vulnerability. How the discoverer proceeds with this information depends on his intrinsic motivation and the incentives offered by the environment. Whatever the choice, it ultimately impacts the risk exposure time of the public. There are many different motivations to direct the discoverer of a vulnerability:

- malicious intent for profit, Path (A) or Path (B)
- altruism, Path (C), Path (D)
- recognition or fame, Path (C)
- forcing unresponsive vendors to address a vulnerability, Path (C), Path (D), or Path (E)
- curiosity and the challenge of vulnerability analysis, Path (C)
- political motives, Path (A) or Path (B)

It is important to note that the number of third party software vulnerability discoveries has not declined over the last decade, as shown in Fig. 6.1, despite massive efforts of the security and software industry.

6.5.1.2 Vulnerability Markets

Information about security vulnerabilities can be a valuable asset. Vulnerability information is traded in both the underground "black market" and the commercial services "white market". While a market for vulnerabilities has developed, vulnerability commercialization remains a hotly-debated topic tied to the concept of vulnerability disclosure. Responsible disclosure fails to satisfy security researchers who expect to be financially compensated, while reporting vulnerabilities to the vendor with the expectation of compensation might be viewed as extortion [11]. On the other hand, cyber-criminals not bound by legal or ethical considerations are willing to invest considerable amounts in suitable vulnerability information. H. D. Moore[3] claims that he was offered between $60k and $120k for critical vulnerabilities in Microsoft products as reported in [6, 28, 30]. Researchers that intend to sell a vulnerability face the possibility that the same vulnerability is discovered, patched, and published independently. This threat of independent discovery pressures them to sell the vulnerability to the quickest bidder instead of the highest one. Factors that determine the market price of a vulnerability are:

- *Exclusivity of information*. This is the key factor, once the vulnerability becomes widely known the value of the information tends to zero.
- *Security impact*. The higher the security impact, the higher the value of the vulnerability.
- *Product popularity*. A vulnerability affecting a popular product has a higher value.

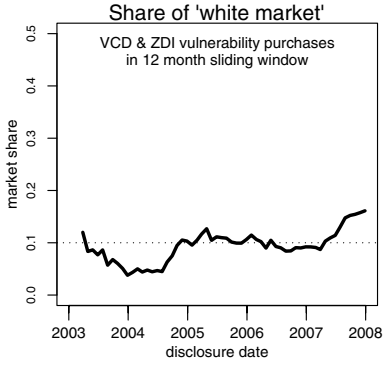
Black Market The black market has developed around the illegal or malicious use of the vulnerability information. Sellers are not driven by ethical considerations. The black-market trade is not openly advertised, and the information is used in a way that generally increases the risk exposure of the public. The lack of trust between sellers and buyers potentially exposes both parties to fraud. Due to the nature of the market accurate information on the number and type of trades completed is not systematically available. Only specific investigations provide some insight into the inner workings, e.g. by Symantec's "Underground Economy Report" [50].

White Market Players in the white market offer commercial services and openly advertise their vulnerability handling policies. Demonstrating and ensuring that buyers and sellers don't have malicious intent is a major challenge for the players in the commercial vulnerability market. White market buyers typically purchase vulnerability information to protect their customers before the vulnerability becomes public knowledge, and inform the vendor of the affected software. Such buyers advertise their ethics and ask security researchers to accept lower compensation with the promise that the information will be used for benevolent purposes [28]. Incentives for the buyers are:

[3] H. D. Moore founded the Metasploit project, an open platform for developing and testing exploit code.

- Publicity generated from disclosing newsworthy vulnerabilities drives interest in their commercial services.
- Providers of intrusion detection and prevention systems include additional protection, which customers might perceive as an advantage.
- They provide the information as a paid service to their customers.

Today, the two primary players in the commercial vulnerability market are *iDefense*, which started their vulnerability contributor program (VCP) in 2003, and *TippingPoint*, with their zero-day initiative (ZDI) started in 2005.TippingPoint's ZDI receives an average of about 40 new vulnerabilities per month, and buys about one out of 10. Vulnerability prices are not disclosed but ZDI runs a "frequent-flyer" style program that can pay out bonuses as high as $20k to top researchers. Together, VCP and ZDI published 793 vulnerabilities affecting 192 different vendors since their start in March 2003 to December 2007. In the same period a total of 8,111 vulnerabilities were published for the same group of 192 vendors, including the 793 bought by VCP and ZDI. We normalize the number of "white market" vulnerabilities with



	Vendor	Total CVEs	Market CVEs	Market Share
1	Microsoft	866	65	7.5%
2	Apple	534	40	7.5%
3	Computer Associates	94	37	39.4%
4	IBM	356	32	9.0%
5	Novell	127	23	18.1%
6	Symantec	127	16	12.6%
7	Gentoo	117	15	12.8%
8	Sophos	26	15	57.7%
9	Ipswitch	57	14	24.6%
10	Sun	332	14	4.2%

Fig. 6.5 Share of commercial vulnerability purchase programs in 12 month moving window (left). Top-10 vendors for which the "white market" brought vulnerabilities from 2003 to 2007 (right)

respect to the total number of vulnerabilities disclosed for the group of affected vendors in the same period to estimate the prevalence of the "white market", Path (E). Using a 12 month sliding window approach, we calculate the share of the "white market" within the group of vendors for which VCP and ZDI already bought vulnerabilities, shown in Fig. 5(a). We observe an almost constant share of about 10% of these commercial programs since the end of 2004 and a rise to over 15% starting in 2007. In Table 5(b) we list the top 10 vendors for which the "white market" bought vulerabilities. We find that the share of vulnerabilities bought varies considerably between vendors, e.g. 4.2% of Sun's and 57.7% of Sophos vulnerabilities follwed Path (E). These numbers shed a first light to what extent "white markets" contribute to the vulnerability ecosystem. Fig. 5(a) shows the prevalence of Path (E), which at the same time provides a *minimum estimate* of the number of vulnerabil-

ities *not* discovered by the vendors themselfs. For example, between March 2003 and December 2007 in average 7.5% of the vulnerabilities affecting Microsoft and Apple were processed by either VCD or ZDI, while other vendors achieved higher shares.

6.5.1.3 Criminal

Any individual or organization misusing vulnerability information for its own profit regardless of motivation is denoted as *criminal* in the model of Fig. 6.4. This can be anyone from an individual hacker to cyber-criminals or government agencies. In this context *misuse* stands for any operation on the targeted system that the user of the system neither approved nor is aware of. Criminals develop or buy exploit material in order to make use of a vulnerability, and typically install malicious software to spy on the user, launch further attacks, and build botnets. Security vulnerabilities in widely used software prove to be a formidable instrument in the hands of cyber-criminals to either enable or expand their business.

6.5.1.4 Vendor

The vendor is the originator of the software affected by a vulnerability. We use the term vendor for commercial products, freeware, and open-source software alike. It is up to the vendor to produce and release a patch once he becomes aware of a vulnerability in his software. In Section 6.6 we measure the zero-day patch share as a metric to measure the performance of vendors' patching and security communication processes.

6.5.1.5 Security Information Provider (SIP)

In the face of a rapidly evolving and hostile environment, businesses and private users alike are in constant need of accurate and validated security information to assess their risk exposure and to protect their systems. However, for the majority of businesses and users it is infeasible and prohibitively costly to monitor, understand and validate all the possible primary information sources in order to extract the security information relevant for them. Several private and government organizations specialize in collecting and publishing security information. Some of these organizations run security research labs, sell security tools, or provide paid security and consulting services. These organizations efficiently monitor the primary sources of security information, validate the content found, and publish their findings as *security advisories* which describe vulnerabilities in a standardized format. These organizations have an important role in the security ecosystem and we denominate them *Security Information Providers (SIP)*. This monitoring of the (in)security environment by SIPs is depicted by dashed curves in Fig. 6.4. Through SIP services, the

public has systematic access to independent, validated, timely, and understandable security information. The availability of trusted security information from SIPs has an important impact on the *behavior* and *incentives* on the actors in the security ecosystem. The combined effect of the efforts of SIPs is a major pillar building the incentives for the actors in the security ecosystem [48]: Collectivity, the role of security information providers in the security ecosystem is comparable to the role of the **free and independent press** in an open society: Issues addressed by them can hardly be ignored, hidden or downplayed.

6.5.1.6 Public

All users, individuals, or organizations, that use software affected by a vulnerability comprise the public. These users typically are in need of accurate and validated security information to assess their risk and to protect their systems until a patch is released by the vendor.

6.5.2 Processes of the Security Ecosystem

Whether ethical or mischievous parties first get information about a new vulnerability impacts the risk exposure of software users. After the discoverer finds a new vulnerability we distinguish five principal paths, denoted Path (A) to Path (E), to proceed as depicted by solid arrows in Fig. 6.4.

6.5.2.1 Path (A) and Path (B)

Cyber-criminals discover security vulnerabilities through their own research or by purchasing the needed information from *black markets* for vulnerabilities [40, 54], represented by Path (A) and Path (B) respectively. For a vulnerability following Path (A) or Path (B) we typically observe the following sequence of events:

$$Discovery \rightarrow Exploit \rightarrow Disclosure \rightarrow Patch \tag{6.4}$$

$$t_{disco}(v) < t_{explo}(v) < t_{discl}(v) < t_{patch}(v) \tag{6.5}$$

The time of vulnerability discovery is likely not available as criminals typically do not share information about their operations. The vendor can only start developing a patch after the vulnerability is actively exploited. Cyber-criminals basically have two options to take advantage of an exploit: *stealthy exploitation* or *full scale exploitation*:

In case of *stealthy exploitation*, cyber-criminals use the exploit only against a few, carefully-selected, high-profile targets, and actively avoid detection to extend the time they can profit from the unknown vulnerability [36]. This phenomenon

is known as "customized malware". However, as described in Section 6.5.3, it is not possible to keep security information secret forever. Eventually, information about the vulnerability spreads to a wider audience. When the disclosure of the vulnerability or the release of a patch is imminent, cyber-criminals may maximize their return of investment by moving on to *full scale exploitation* of the exploit.

In case of *full scale exploitation*, cyber-criminals release the exploit against a large population of targets to take advantage of a greater proportion of unprotected systems. With the higher percentage of compromised systems comes the greater risk of exposure of their activity, which eventually exposes the vulnerability to detection and subsequent disclosure. SIPs and other organizations monitor the (in)security scene, exploit archives, and research malicious activity:

- Anti-virus vendors or providers of managed security services (MSS) capture a sample of the exploit for analysis.
- Hoenypots and honeynets capture a sample of the exploit for analysis [24]
- Vendors capture a sample of the exploit through their error reporting mechanisms [29] (usually if the exploit crashes on certain configurations).

These activities lead to the timely disclosure of the underlying vulnerability. Thus, Path (A) and Path (B) favor the malicious use of vulnerability information resulting in an increase of security impact and exposure to risk for users: a decrease of social welfare given the ubiquitous use of computer and communication technologies in our society.

6.5.2.2 Path (C)

The discoverer publishes information about the vulnerability on a suitable channel (e.g. in a security conference or on a security mailing list[4]). Following Path (C), the vulnerability information is available to all interested parties at the same time: the criminals, the vendor, and the public. SIPs monitoring the security landscape spot this information and report it in a new security advisory. However, usually writing an exploit based on vulnerability information is less complex and faster than writing and releasing a patch. In the extreme case of *full disclosure*, the discoverer includes proof-of-concept code and exploit material. A discoverer following Path (C) is typically not financially motivated. He either decides to publish the vulnerability firsthand, or he does so because the vendor was not responsive. We discuss these options in Section 6.5.3. For a vulnerability following Path (C) we typically observe the following sequence of events:

$$Discovery \rightarrow Disclosure \rightarrow Exploit \rightarrow Patch \tag{6.6}$$

$$t_{disco}(v) < t_{discl}(v) < t_{explo}(v) < t_{patch}(v) \tag{6.7}$$

[4] FullDisclosure and BugTraq are two well known security mailing lists

6.5.2.3 Path (D) and Path (E)

The discoverer reports the vulnerability either directly to the vendor, Path (D), or through a commercial vulnerability market, Path (E). In case the vulnerability affects several vendors the discoverer can do so using the services of a CERT/CC[5]. The discoverer and the vendor then typically follow the responsible disclosure process described in Section 6.5.3: the vulnerability information is kept secret until the vendor has a patch ready for release. If the vendor is not responsive or uncooperative, the discoverer might fail over to Path (C). When the patch is ready, the discoverer publishes his advisory at the same time as the vendor releases the patch. Criminals can only start with the development of an exploit after a patch is available. For a vulnerability following Path (D) or Path (E) we typically observe the following sequence of events:

$$Discovery \rightarrow \left\{ \begin{matrix} Disclosure \\ Patch \end{matrix} \right\} \rightarrow Exploit \tag{6.8}$$

$$t_{disco}(v) < t_{discl}(v) = t_{patch}(v) < t_{explo}(v) \tag{6.9}$$

Path (E) is an option for a financially motivated discoverer who does not want to sell the vulnerability in the underground where misuse is very likely. The prevalence of commercial vulnerability markets is shown in Fig. 5(a). Path (D) and Path (E) are more favorable for public risk exposure, as the vendor gets the information about the vulnerability before mischievous parties do. On the other hand, cyber-criminals have also refined their ability to analyze vulnerability information from vulnerability disclosures and reverse engineering of patches. Recent research demonstrated the potential of automated exploit generation based on a patch [9]. Cyber-criminals quickly create exploits upon the availability of such information.

6.5.3 The Disclosure Debate

Appreciation of vulnerability disclosure concepts and the accompanying incentives of the players involved is a prerequisite to understand the processes of the security ecosystem. The disclosure debate discusses the question of how to handle information about security vulnerabilities in order to minimize the security impact for the society:

- On the one hand, public disclosure of security information enables informed consumer choice and inspires vendors to be truthful about flaws, repair vulnerabilities and build more secure products [11]. This is the *security through transparency* stance of Kerckhoff [23].
- On the other hand, vulnerability information can give attackers (not sophisticated enough to identify a vulnerability on their own) the very information they

[5] CERT Coordination Center

need to exploit a security hole in a computer or system and cause harm. This is the *security through obscurity* stance[6].

The process of *responsible disclosure* evolved as a middle way between the opposing stances found in the disclosure debate. It has evolved and become a accepted way to handle security information [35].

Full disclosure is a security philosophy that holds that the details of security vulnerabilities should be available to everyone in a timely fashion. Before the systematic publication of software vulnerabilities, vendors typically would not bother to spend the time and money to fix vulnerabilities, believing in the security of secrecy [7, 11, 25, 41, 46]. Public disclosure or the threat of disclosure give vendors a strong incentive to fix the problem quickly. It is inevitable that cyber-criminals get the information alike with the public disclosure. This disadvantage is more than compensated by providing benign users the information needed to defend their systems as there is no way to assure that cyber-criminals do not already possess the same vulnerability information.

Responsible Disclosure Process The key insight from the disclosure debate is that secrecy mainly prevents people from assessing their own risks, which contributes to a false sense of security [42]. The process of *responsible disclosure* evolved as a middle course between the extremes of *full disclosure* and *security through obscurity*: The researcher discloses full information only to the vendor, expecting that the vendor will start the process to develop a patch, as in Path (D) or Path (E). In return, the vendor is expected to expeditiously issue a patch and give credit to the researcher for his discovery. The vendor is well incentivized to collaborate, as the discoverer can revert to *full disclosure* Path (C) if the vendor becomes unresponsive or the vulnerability is reported through other channels. In the last phase the discoverer will coordinate the publication of his advisory with the vendor's publication of the vulnerability information and the patch. An increasing number of vendors and security organizations adopted some form of *responsible disclosure* over the last decade [7, 8, 20].

6.6 The Dynamics of (In)Security

In this section, we focus on the evolution of the dynamics between security *(availability of patches)* and insecurity *(availability of exploits)*, based on the vulnerability lifecycle normalized to the time of disclosure. The intimate relation between the vulnerability lifecycle events and the processes in the security ecosystem are depicted in Fig. 6.4. The availability of an exploit poses a security threat, whereas the availability of a patch neutralizes this threat if the patch gets installed on the vulnerable system. Assuming that both the exploit and the patch work as intended by the respective originator, the resulting security risk for software users will depend

[6] also often referred to as *bug secrecy*

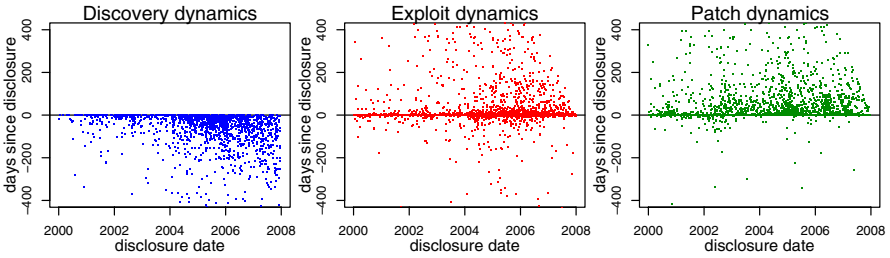


Fig. 6.6 Scatter plot of time of vulnerability discovery (left), exploit availability (center), and patch availability (right) by disclosure date.

strongly on the timing or dynamics of the availability of these. We measure the current state and identify global trends. For all vulnerabilities we know the time of the vulnerability disclosure $t_{discl}(v)$ taken from the fastest SIPs reporting this CVE with a resolution of one calendar day. Fig. 6.7 shows the number of vulnerabilities for which we found the time of discovery $|V_{disco}|$, time of exploit availability $|V_{explo}|$, and the time of patch availability $|V_{patch}|$ for every year from 2000 to 2007. The absolute number of vulnerabilities disclosed in a given year (100%) is visibile in Fig. 6.1. In the following of this section we individually discuss the dynamics of vulnerability *discovery*, *exploit availability*, and *patch availability* and describe the data sources used to build V_{disco}, V_{explo}, and V_{patch}. We examine the vulnerability

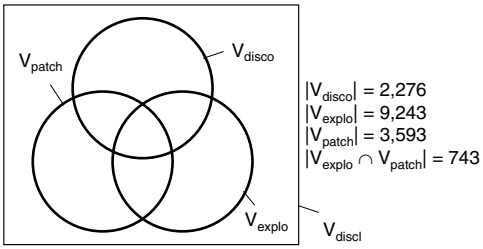


Fig. 6.7 Number of observed events within all vulnerabilities disclosed from 2000 to 2007.

lifecycle by looking at how the time of the events $\alpha \in E = \{disco, explo, patch\}$ relate to the respective disclosure time $t_{discl}(v)$ of the vulnerability. For all vulnerabilities from 2000 to 2007 and each type of event, we present a scatter plot, the associated distribution function, and yearly summaries to evaluate the evolution and identify trends. Normalization of the vulnerability lifecycle events with respect to the disclosure time is key to evaluate the aggregated dynamics of thousands of vulnerabilities. We build Δt_{disco}, Δt_{explo}, and Δt_{patch} as follows:

$$\Delta t_\alpha(v) = t_\alpha(v) - t_{discl}(v) \qquad \alpha \in E,\ v \in V_\alpha \tag{6.10}$$

Essentially $\Delta t_\alpha(v)$ represents the number of days event $\alpha \in E$ happened *before* or *after* the disclosure of vulnerability v:

$$sgn(\Delta t_\alpha(v)) = \begin{cases} -1 & \alpha \text{ occurs } \textit{before} \text{ disclosure} \\ 0 & \alpha \text{ occurs } \textit{at} \text{ disclosure} \\ 1 & \alpha \text{ occurs } \textit{after} \text{ disclosure} \end{cases}$$

Δt_{disco} is an estimator of the "pre-disclosure" risk and Δt_{patch} is an estimator of the "post-disclosure" risk period as introduced in Section 6.4.1.

Scatter plots We first use scatter plots of Δt_α to visualize the distribution and the evolution of events $\alpha \in E$ over the last eight years. In the scatter plots of Fig. 9.4 each point $P_\alpha(v)$ of event α is built according to

$$P_\alpha(v) \rightarrow (x,y) \quad \begin{cases} x = t_{discl}(v) \\ y = \Delta t_\alpha(v) \end{cases} \quad \alpha \in E,\ v \in V_\alpha \tag{6.11}$$

In all scatter plots, the x-axis is the calendar day of the disclosure of vulnerability v. The y-axis represents the time difference of event α to the disclosure of vulnerability v.

Distribution function To further analyze the dynamics, we plot and discuss the cumulated distribution $\mathscr{P}_{\leq}(X \leq x)$ of the same data used to generate the scatter plots. The $ecdf_\alpha(x)$ of event $\alpha \in E$ is

$$\begin{aligned} \mathscr{P}_{\leq}(X \leq x) &= ecdf_\alpha(x) \\ &= \left| \{ v \in V_\alpha \mid \Delta t_\alpha(v) \leq x \} \right| \end{aligned} \tag{6.12}$$

In Fig. 6.8, Fig. 6.9, and Fig. 6.10 we plot the $ecdf_\alpha(x)$ for discovery, exploit, and patch availability for the range of $x = \pm 400$ days around disclosure. These plots give insight in to the aggregated dynamics of the vulnerability lifecycle.

6.6.1 Discovery Dynamics

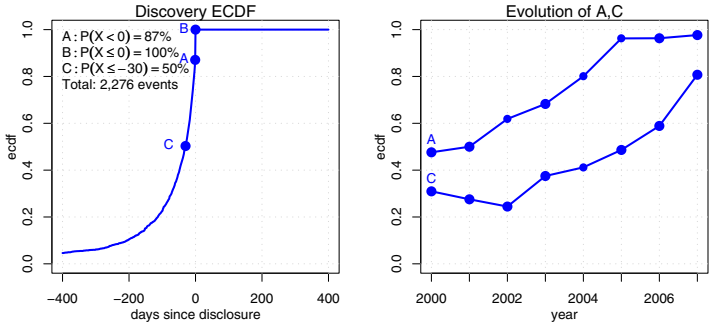


Fig. 6.8 Empirical cumulated distribution of the *discovery time* (left) and yearly evolution of selected points in the ecdf (right).

Usually the time of discovery of a vulnerability is not publicly known until *after* its disclosure. Indeed, for many vulnerabilities the time of discovery will never be known or reported to the public, depending on the motives of the discoverer. Cyber-criminals - and most software vendors - won't provide information about their vulnerability discoveries to the public. However, there are a few sources from which we can derive the time of vulnerability discovery. One source is the Open Source Vulnerability Database (OSVDB); the security bulletins of commercial vulnerability markets are another source. When *iDefense* or *TippingPoint* buy a vulnerability, they record the time of purchase or the time at which they notified the vendor of the affected software. Upon public release, this date can be retrieved from the disclosure timeline of the security advisory. Using this methodology we determined the time of discovery $t_{disco}(v)$ for a subset $V_{disco} \subset V$ of all vulnerabilities. Further, as the disclosure of a vulnerability implies its discovery we can state

$$t_{disco}(v) \leq t_{discl}(v) \qquad \forall\, v \in V_{disco} \tag{6.13}$$

Using Eq. 6.1 we can calculate $\Delta t_{disco}(v)$, a minimum estimator for the "pre-disclosure" risk. The true "pre-disclosure" risk period is always longer than what we can estimate based on publicly available data. In Fig. 6.8, the values for $x < 0$ show the distribution of the "pre-disclosure" risk from 2000 to 2007. For $x \geq 0$ $\mathscr{P}_{\leq}(X \leq x)$ equals 1 as disclosure implies discovery (Eq. 6.13). In Fig. 6.8 we plot the values for (A) $\mathscr{P}_{\leq}(X < 0)$ and (C) $\mathscr{P}_{\leq}(X < -30)$ for each year. The rise of (A) since 2000 points out that over time we observe more events with $t_{disco} < t_{discl}$ compared to $t_{disco} \leq t_{discl}$. The course of line (C) $\mathscr{P}_{\leq}(X < -30)$ shows that since 2000 more than 24% of the vulnerabilities were known to insiders more than 30 days before disclosure. In 2007 this share rose to 80% of the vulnerabilities. The course of line (C) is a minimum estimator of the "pre-disclosure" risk, of which one part is desirable - as it partially measures the success of the responsible disclosure process. However, for most vulnerabilities (mostly the ones discovered and abused by cyber-criminals) we never learn the discovery date. E.g. we only know the discovery date for 12% percent of the vulnerabilities patched in the last 5 years. We therefore consider our measurement of the "pre-disclosure" risk as a minimum estimator for the amount of time any privileged party has access to security critical information. This clearly shows the potential of the abuse of vulnerability information, especially as we have no data on vulnerability discoveries made by cyber-criminals or traded on the "black market". We conclude that vulnerabilities are systematically known to insiders (good and bad) well before the public learns about it.

6.6.2 Exploit Availability Dynamics

From the public exploit archives listed in Section 13.4 we can find the time of exploit availability for a subset $V_{explo} \subset V$ of all vulnerabilities. These exploit archives report the date when the exploit was published. The actual number of exploits avail-

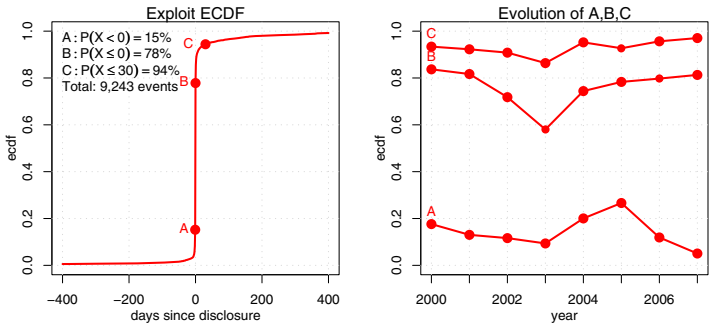


Fig. 6.9 Empirical cumulated distribution of the *exploit availability time* (left), yearly evolution of selected points in the ecdf (right).

able on these exploit archives is larger than $|V_{explo}|$ as we exclude exploits that cannot be correlated to a given CVE. Cyber-criminals use their exploit material for profit and have no incentive to publish their material on public exploit archives. Eventually, some of the exploits used exclusively by cyber-criminals make their way into exploit archives (as an exploit, proof of concept, test for patch). However, these postings are delayed. On the other hand, cyber-criminals monitor exploit archives and quickly enhance their repository of malware, should they find material previously unknown to them. As a result, we can only estimate the extent of yet undisclosed exploit information available to cyber-criminals at any time. V_{explo}, based on the content of public exploit archives, is therefore a *minimum estimate* for the true number of exploits available to cyber-criminals at a any given date. The time of exploit availability is $t_{explo}(v)$ with $v \in V_{explo} \subset V$. The scatter plot in Fig. 9.4 (center) shows the distribution of these exploits from 2000 to 2007. We observe that exploits are available both *before* and *after* the disclosure of the vulnerability, with an increasing density of exploit availability close to the disclosure day as of 2004. The plot of the cumulated distribution $\mathscr{P}_{\leq}(X \leq x)$ of Fig. 6.9 (left) quantifies the high dynamics of exploit availability close to the vulnerability disclosure. The sudden rise of $\mathscr{P}_{\leq}(X \leq x)$ from 15% before disclosure to 78% at disclosure from 2000 to 2007 quantifies the so called zero-day exploit phenomena [25]. A zero-day exploit is an exploit that takes advantage of a vulnerability at or before the day the vulnerability is disclosed. In other words, the vendor and the public have zero days to prepare for the security breach. The plot on Fig. 6.9 (right) shows that the zero-day exploit availability is above 70% for the last eight years with the only exception of 58% in 2003. Several mechanisms lead to the very high exploit availability at the time of disclosure. The combined effect of prior vulnerability knowledge and rapid analysis of disclosed vulnerability information (as discussed in Section 6.5.2.1) is readily seen by the increased activity at the disclosure day, and measured with a zero-day exploit availability of close to 80% since 2003. We cannot distinguish these mechanisms due to the limited scope and resolution (one calendar day) of publicly available information. Further, exploit availability reaches 94% 30 days after disclosure. Cyber-criminals systematically take advantage of users failing to

install patches quickly, or not having the latest patches installed. We analyzed and measured Internet users' discipline of patching their Web browsers in [12, 13].

6.6.3 Patch Availability Dynamics

A vendor typically reports the date when a new patch is released together with the patch bulletin or security advisory. To measure the dynamics of patch releases we download, parse, and correlate patch release bulletins of the seven vendors *Adobe, Apache, Apple, Microsoft, Mozilla Foundation, Oracle,* and *RedHat*. We chose these vendors to cover major players of the industry and with respect to the distribution of vulnerabilities among vendors as of Fig. 6.2. Using the release date posted in these vendor bulletins we determine the time of patch availability $t_{patch}(v)$ for a subset of vulnerabilities $V_{patch} \subset V$. Fig. 6.7 shows the number of vulnerabilities for which we have patch information available through the analysis of these seven vendors. The scatter plot in Fig. 9.4 (right) shows the distribution of the availability of these patches from 2000 to 2007. We observe that patches are mostly available

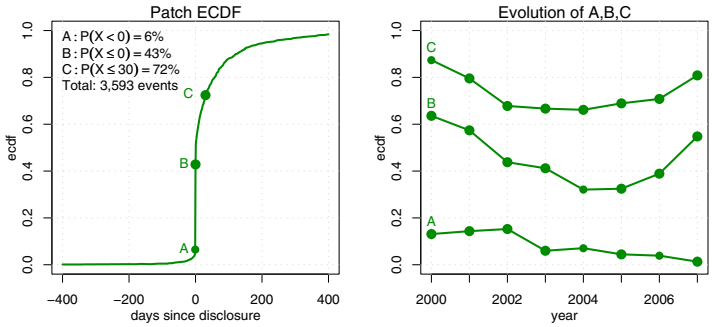


Fig. 6.10 Empirical cumulated distribution of the *patch availability time* (left), yearly evolution of selected points in the ecdf (right).

at or *after* the disclosure of the vulnerability. The plot of the cumulated distribution $\mathscr{P}_{\leq}(X \leq x)$ of Fig. 6.10 (left) quantifies the dynamics of patch availability close to vulnerability disclosure. Essentially, Δt_{patch} reveals the performance of the software industry in providing patches, a measure of the "post-disclosure" risk introduced in Section 6.4.1 and estimator of Path (D) and Path (E). Patch availability 30 days before the time of disclosure is at 2%. There are only few vulnerabilities found for which a patch already exists before the disclosure. The sudden rise of $\mathscr{P}_{\leq}(X \leq x)$ from 6% one day before disclosure to 43% at disclosure from 2000 to 2007 quantifies what we call the *zero-day patch* phenomena. The fraction of zero-day patches can be interpreted as a measure of the *responsible disclosure process*, implying Path (D) or Path (E) in our security ecosystem model. Before a patch is ready for publication the vendor needs time to analyze the vulnerability, develop,

test, document, and finally release the patch. Typically, a vendor is unable to release a patch within twenty-four hours of vulnerability discovery. Thus, to achieve a zero-day patch the vendor needs early notification of the vulnerability, typically through the responsible disclosure process Path (D), which includes contributions by the white market Path (E). The rise of $\mathscr{P}_{\leq}(X \leq x)$ for $x > 0$ measures how fast vendors react to vulnerability disclosures. Patch availability increases from 46% at disclosure to 72% at 30 days after the disclosure (equalling 28% unpached vulnerabilities 30 days after disclosure). This is a low number compared to the exploit availability of 94% 30 days after disclosure. Further, 13% of the vulnerabilites are still unpatched 90 days after the disclosure.

To determine how the risk of a vulnerability affects the patch performance we separately analyze the data for the three risk classes “high”, “medium”, and “low”. The results indicate that patch performance of “low” risk vulnerabilities consistently lags behind the performance of “high” and “medium” risk vulnerabilities, especially after disclosure. At disclosure we measure $\mathscr{P}_{\leq}(X \leq 0)$ to be 45%, 43% and 34% for “high”, “medium”, and “low” risk vulnerabilities repsectively. After disclosure we measure $\mathscr{P}_{\leq}(X \leq 30)$ to be 77%, 72% and 56% for “high”, “medium”, and “low” risk vulnerabilities repsectively. From these observations, we assume that the risk class of a vulnerability marginally effects the patch release performance in the sense that patches for “high” and “medium” risk vulnerabilities are prioritized against patches for “low” risk vulnerabilities. If the technological complexity of a fix to vulnerability were the dominant parameter to determine patch performance, then our measurements would lead to the conclusion that “low” risk vulnerabilities are generally more complex to fix than “high” or “medium” risk vulnerabilities, which we consider unlikely. We rather assume that work flow processes and priorization (and with it incentives) are at least as important as technical complexity to determine patch performance. Note that the discovery of a vulnerability by the vendor itself is also considered as responsible disclosure. An appropriately motivated employee discovering a vulnerability could also choose to offer this information to cyber-criminals instead. The share of zero-day patches indicates the sum of vulnerability discoveries by the vendor and vulnerabilities reported to the vendor through the “responsible disclosure” process. Applying these results to our model of the processes in the security ecosystem, Fig. 6.4, we conclude that between 6% and 43% of the vulnerabilites of the analyzed vendors followed the process Path (D) or Path (E). A detailed analysis of Microsoft and Apples zero-day patch performance is published in [14].

6.6.4 (In)security Dynamics

6.6.4.1 The Gap of Insecurity

An interesting aspect of our analysis is the direct comparison of the exploit and patch availability distributions and their trends over the last five years. For this we analyze

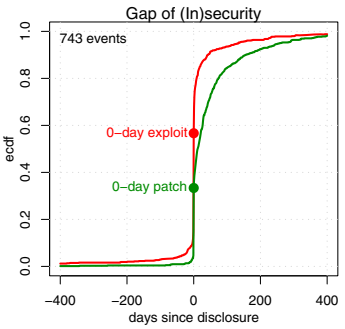


Fig. 6.11 Direct comparison of patch availability vs. exploit availability.

the cumulated distribution of $\Delta t_{patch}(v)$ for all vulnerabilities $v \in V_{patch}$ together with the cumulated distribution of $\Delta t_{explo}(v)$ for all $v \in V_{explo}$. Through vendor Web sites we have systematic access to *all patches* published by a given vendor and V_{patch} contains *all patches* published by our seven vendors. However, not all exploits are made available on public exploit archives, as explained in Section 6.6.2, so the distribution of $\Delta t_{explo}(v)$ is a *lower estimate* of the exploit availability. True exploit availability is always faster. Fig. 6.11 shows that exploit-availability continuously

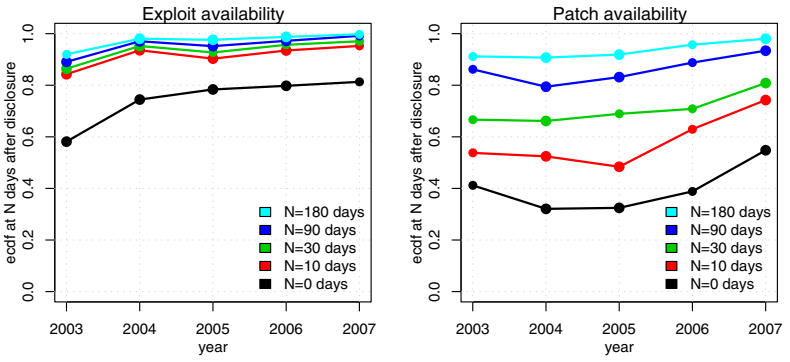


Fig. 6.12 Evolution of exploit availability and patch availability at $N \in \{0, 10, 30, 90, 180\}$ days after disclosure.

exceeds patch-availability for the full range ± 400 days around the day of disclosure. Exploit availability also consistently exceeds patch availability in every single year since 2000. This gap, which quantifies the difference between exploit- and patch-availability, is an indicator of the risk exposure and its development over time. This systematic gap also stresses the importance for the availability of independent and timely security information, the role of SIPs explained in Section 6.5.1.5. In Fig. 6.12 we plot distinct points at 0, 10, 30, 90 and 180 days of the cdf of Δt_{explo} and Δt_{patch} to visualize their evolution over time. Generally, both exploit and patch availability were increased over the last five years. With the exception of 2005, exploit availability increased steadily since 2003, and we observe a greater rise closer to the disclosure day. Exploit availability 30 days after disclosure continuously exceeds

90% since 2004. We observe high exploit dynamics within 10 days of disclosure; thereafter exploit availability rises only very slowly. We attribute this observation to the following causes:

- Exploits already known to cyber-criminals *before* public disclosure of the vulnerability.
- Increased capability to generate exploits either through reverse-engineering of patches or based on disclosed vulnerability information.
- Automated attack tools for Web application vulnerabilities that can actually *discover and exploit* a vulnerability. It is only afterward that the consultant/user of the tool realizes that the vulnerability exists - and then informs them that they need to fix it.

We cannot distinguish these causes based on our data, so we measure the aggregate effect. Note again that our data is a minimum estimate of the true availability of exploits. On the other hand, also patch availability increases almost steadily over the last years, although starting from a lower level than exploit availability. Closer to the disclosure, patch availability first dipped around 2005 and then caught up in the last three years. Again, patch availability is always lower than exploit availability at any day. Patch availability 90 days after disclosure does not surpass exploit availability 10 days after disclosure. We attribute patch availability performance to two different processes:

Patch release at zero-day: The release of a patch at the same day as the public disclosure of the vulnerability implies the vendor had early notification of the vulnerability ("responsible disclosure"), Path (D) or Path (E). A vendor is typically not able to analyze vulnerability information, then develop, test, and release a patch in less than a day. However, whether a vendor receives early notification from vulnerability discoverers is only partially under control of the vendor. This is to a high degree an exogenous factor that the vendor can only control in the long term, by establishing a trust relationship with the security community.

Patch release after disclosure: The time needed to release a patch upon knowing the vulnerability is under control of the vendor, a endogenous factor. Here we measure what a vendor *can do*, and what he is *willing to do* given technological complexity to fix the software, and economic incentives or constraints.

We believe that a good relationship with the security community can provide a higher share of early notifications of vulnerabilities which benefits a vendor in the following ways:

- Within responsible disclosure the vendor has more control of the time available to develop and release a patch than under the pressure of an already published vulnerability. This will typically result in a more efficient allocation and use of available resources of the vendor.
- A higher share of zero-day patches will be perceived as a better service to the customer.

Further, the systematic gap between patch and exploit availability underlines the role and importance of SIPs. During these periods, software users are exposed to risk of

exploit without already having received remediation from the vendor. It is during this time that security information on the threats is most important. The observed trend toward increased patch availability *at* and *after* the public disclosure indicates that the processes involved to release patches (technological, economic, incentives) have not yet reached saturation. A detailed analysis of Microsoft and Apples patch release performance since 2002 was published in [14]. Continued measurements using the methodologies presented in this chapter should be able to identify the limits of such processes at macroscopic scale.

Limitations The presented analysis is a first attempt at making the processes in the vulnerability ecosystem measurable. As there exists no systematic access to data on cyber-criminals operations, such an analysis comes with limitations. The *zero-day patch* share implies Path (D) or Path (E), however without excluding prior discovery through cyber-criminals. While we measured the extent of the *zero-day exploit* phenomena, the one day resolution of our data does not allow to distinguish between exploits that were derived from patches from exploits available before disclosure. Given the skewed distribution of vulnerabilities per vendor, the analysis must be viewed in the context of the specific vendors measured.

6.7 Conclusion

We introduced a model of the security ecosystem to capture its major players and processes. This is the first model of the security ecosystem that consolidates hitherto separately discussed aspects of the security processes. On the basis of the model we analyzed and discussed the roles and incentives of the players involved, backed with empirical data of more than 27,000 vulnerabilities. We enumerated the options of vulnerability discoverers, and visualized the security impact of their choices. For the first time we estimated the success of the "responsible disclosure process" backed with measurements, using the zero-day patch share as a metric. Our measurement revealed that commercial vulnerability markets cannot be neglected; on average they handle between 10% and 15% of the vulnerabilities of major software vendors. We found that exploit availability has consistently exceeded patch availability since 2000. This systematic gap between the availability of exploits and patches highlights the rapid dynamics around the day of vulnerability disclosure and the all-important role of *security information providers (SIP)* within the security ecosystem. The complexity and delay of installing patches paired with the fact that we can only provide an minimum estimate for exploit availability stresses the need for third party protection *and* timely availability of security information to the public. Our measurement methods are based entirely on publicly available information and provide a useful tool to measure the state of the security ecosystem and its evolution over time.

References

1. Packetstorm Security. http://packetstormsecurity.org
2. Anderson, R., Moore, T.: The Economics of Information Security. Science **314**(5799), 610–613 (2006). http://dx.doi.org/10.1126/science.1130992
3. Arbaugh, W.A., Fithen, W.L., McHugh, J.: Windows of vulnerability: A case study analysis. Computer **33**(12), 52–59 (2000). DOI http://doi.ieeecomputersociety.org/10.1109/2.889093
4. Arora, A., Krishnan, R., Nandkumar, A., Telang, R., Yang, Y.: Impact of vulnerability disclosure and patch availability – an empirical analysis. In: R. Anderson (ed.) Workshop on the Economics of Information Security (WEIS). Cambridge, UK (2004)
5. Arora, A., Telang, R., Xu, H.: Optimal policy for software vulnerability disclosure. In: Workshop on the Economics of Information Security (WEIS) (2004)
6. Boehme, R.: Vulnerability markets. what is the economic value of a zero-day exploit? In: Private Investigations (Proc. of 22nd Chaos Communication Congress). CCC (2005). DOI http://doi.acm.org/10.1145/1162666.1162671
7. Chambers, J.T., Thompson, J.W.: Niac vulnerability disclosure framework. Department of Homeland Security DHS (2004)
8. Christey, S., Wysopal, C.: Responsible vulnerability disclosure process (2002). http://tools.ietf.org/html/draft-christey-wysopal-vuln-disclosure-00
9. David, B., Pongsin, P., Dawn, S., Jiang, Z.: Automatic patch-based exploit generation is possible. In: IEEE Security and Privacy, 2008, pp. 143–157 (2008)
10. Duebendorfer, T., Frei, S.: Why Silent Updates Boost Security. Tech. Rep. 302, TIK, ETH Zurich (2009). http://www.techzoom.net/silent-updates
11. Electronic Frontier Foundation EFF: Coders' Rights Project Vulnerability Reporting FAQ
12. Frei, S., Dubendorfer, T., Ollmann, G., May, M.: Understanding the web browser threat. Tech. Rep. 288, ETH Zurich (2008). http://www.techzoom.net/papers
13. Frei, S., Duebendorfer, T., Plattner, B.: Firefox (In)Security Update Dynamics Exposed. Computer Communication Review **39**(1) (2009)
14. Frei, S., Tellenbach, B., Plattner, B.: 0-day patch - exposing vendors (in)security performance. BlackHat Europe (2008). http://www.techzoom.net/papers
15. FrSIRT: French Security Incident Response Team. http://www.frsirt.com
16. Hasan Cavusoglu, H.C., Raghunathan, S.: Emerging issues in responsible vulnerability disclosure. In: WITS (2004)
17. H.D. Moore: The Metasploit Project. http://www.metasploit.com
18. IBM Internet Security Systems: The Lifecycle of a Vulnerability. www.iss.net/documents/whitepapers/ISS_Vulnerability_Lifecycle_Whitepaper.pdf (2005)
19. IBM Internet Security Systems - X-Force: X-Force Advisory. http://www.iss.net
20. IBM Internet Security Systems - X-Force: Responsible vulnerability disclosure process (2004). http://documents.iss.net/literature/vulnerability_guidelines.pdf
21. iDefense: Vulnerability Contributor Program. Http://labs.idefense.com/vcp
22. Kannan, K., Telang, R.: An economic analysis of market for software vulnerabilities. In: Workshop on the Economics of Information Security (WEIS) (2004)
23. Kerckhoffs, A.: La cryptographie militaire. Journal des sciences militaires **IX**, 5–83 (1883)
24. Leita, C., Dacier, M., Wicherski, G.: SGNET: a distributed infrastructure to handle zero-day exploits. Tech. Rep. EURECOM+2164, Institut Eurecom, France (2007)
25. Levy, E.: Approaching zero. IEEE Security and Privacy **2**(4), 65–66 (2004). DOI http://doi.ieeecomputersociety.org/10.1109/MSP.2004.33
26. Lindner, F.F.: Software security is software reliability. Commun. ACM **49**(6), 57–61 (2006). DOI http://doi.acm.org/10.1145/1132469.1132502
27. Maillart, T., Sornette, D.: Heavy-tailed distribution of cyber-risks (2008). URL http://www.citebase.org/abstract?id=oai:arXiv.org:0803.2256
28. McKinney, D.: Vulnerability bazaar. IEEE Security and Privacy **5**(6), 69–73 (2007). DOI http://doi.ieeecomputersociety.org/10.1109/MSP.2007.180
29. Microsoft: Windows Error Reporting. Http://technet.microsoft.com/en-us/library/bb490841.aspx

30. Miller, C.: The legitimate vulnerability market: Inside the secretive world of 0-day exploit sales. In: Workshop on the Economics of Information Security (WEIS) (2007)
31. Milw0rm: Milw0rm Exploit Archive. http://www.milw0rm.com
32. MITRE : CVE Vulnerability Terminology 3. http://cve.mitre.org/about/terminology.html
33. MITRE: Common Vulnerabilities and Exposures (CVE). http://cve.mitre.org
34. Oborne, M.W.: The Security Economy. OECD, Paris : (2004). ISBN 92-64-10772-X
35. OISA Organization for Internet Safety: Guidelines for Security Vulnerability Reporting and Response. http://www.oisafety.org/guidelines/
36. Ollmann, G.: The evolution of commercial malware development kits and colour-by-numbers custom malware. Computer Fraud & Security **2008**(9), 4 – 7 (2008). http://dx.doi.org/10.1016/S1361-3723(08)70135-0
37. OSVDB: Open Source Vulnerability Database. Http://www.osvdb.org
38. Ozment, A.: Improving vulnerability discovery models. In: QoP '07: Proceedings of the 2007 ACM workshop on Quality of protection, pp. 6–11. ACM, New York, NY, USA (2007). DOI http://doi.acm.org/10.1145/1314257.1314261
39. Pfleeger, S.L., Rue, R., Horwitz, J., Balakrishnan, A.: Investing in cyber security: The path to good practice. The RAND Journal **Vol 19, No. 1** (2006)
40. Radianti, J., Gonzalez, J.J.: Understanding hidden information security threats: The vulnerability black market. Hawaii International Conference on System Sciences **0**, 156c (2007). DOI http://doi.ieeecomputersociety.org/10.1109/HICSS.2007.583
41. Schneier, B.: Locks and Full Disclosure. IEEE Security and Privacy **01**(2), 88 (2003)
42. Schneier, B.: The nonsecurity of secrecy. Commun. ACM **47**(10), 120 (2004)
43. Secunia: Vulnerability Intelligence Provider. http://www.secunia.com
44. SecurityTracker: SecurityTracker. http://www.SecurityTracker.com
45. Securityvulns: Computer Security Vulnerabilities. http://securityvulns.com/
46. Shepherd, S.A.: Vulnerability Disclosure. SANS InfoSec Reading Room (2003)
47. Shostack, A., Stewart, A.: The new school of information security. Addison-Wesley (2008)
48. Stefan Frei and Martin May: Putting private and government CERT's to the test. In: 20th Annual FIRST Conference, June 22-27, 2008, Vancouver, Canada (2008)
49. Symantec: SecurityFocus. http://www.securityfocus.com/vulnerabilities
50. Symantec: Report on the Underground Economy (2008)
51. Thomas, B., Clergue, J., Schaad, A., Dacier, M.: A comparison of conventional and online fraud. In: CRIS'04, 2nd Int. Conf. on Critical Infrastructures, Oct 25-27, 2004 - Grenoble
52. TippingPoint: Zero day initiative (zdi). http://www.zerodayinitiative.com/
53. US-CERT: US-CERT. http://www.us-cert.gov/aboutus.html
54. Whipp, M.: Black market thrives on vulnerability trading. PCpro (2006). http://www.pcpro.co.uk/news/84523