Investigating Zero-Day Attacks

LEYLA BILGE AND TUDOR DUMITRAS

Leyla Bilge became a Senior Research Engineer at Symantec Research Labs in February 2012 after obtaining her Ph.D. from EURECOM, based in the south of France, with work focusing on network-based botnet detection. In her thesis, she proposed three different network-based botnet detection schemes, one of which is Exposure.
Leylya_Yumer@symantec.com

Tudor Dumitras is an Assistant Professor in the Electrical and Computer Engineering Department at the University of Maryland, College Park. His research focuses on Big Data approaches to problems in system security and dependability. In his previous role at Symantec Research Labs he built the Worldwide Intelligence Network Environment (WINE). He has received the 2011 A. G. Jordan Award, from the ECE Department at Carnegie Mellon University, for an outstanding Ph.D. thesis and for service to the community; the 2009 John Vlissides Award, from ACM SIGPLAN, for showing significant promise in applied software research; and the Best Paper Award at ASP-DAC '03.
tudor.dumitras@gmail.com

We conducted a systematic study on data available through Symantec's Worldwide Intelligence Network Environment to help us to understand the duration and prevalence of zero-day attacks. Based on what we learned, we developed a methodology that automatically identifies zero-day attacks that have affected a large number of real hosts worldwide. Our methodology was not only able to detect already known zero-day attacks but also some that were previously unknown. Moreover, we discovered that the majority of zero-day attacks were able to stay undercover for a surprisingly long time.

A *zero-day attack* is a cyberattack exploiting a vulnerability that has not been disclosed publicly. There is almost no defense against a zero-day attack: while the vulnerability remains unknown, the software affected cannot be patched and antivirus products cannot detect the attack through signature-based scanning. For cybercriminals, unpatched vulnerabilities in popular software, such as Microsoft Office or Adobe Flash, represent a free pass to any target they might want to attack, from Fortune 500 companies to millions of consumer PCs around the world. For this reason, the market value of a new vulnerability ranges between $5,000 and $500,000 [7]. Examples of notable zero-day attacks include the 2010 Hydraq trojan, also known as the "Aurora" attack, which aimed to steal information from several companies; the 2010 Stuxnet worm, which combined four zero-day vulnerabilities to target industrial control systems; and the 2011 attack against RSA. Unfortunately, very little is known about zero-day attacks because, in general, data is not available until *after* the attacks are discovered. Prior studies rely on indirect measurements (e.g., analyzing patches and exploits) or the post-mortem analysis of isolated incidents, and they do not shed light on the duration, prevalence, and characteristics of zero-day attacks.

We conducted a systematic study of zero-day attacks from 2008 to 2011 and developed a technique for identifying and analyzing zero-day attacks from the data available through the Worldwide Intelligence Network Environment (WINE), a platform for data-intensive experiments in cybersecurity [8]. WINE includes field data collected by Symantec on 11 million hosts around the world. These hosts do not represent honeypots or machines in an artificial lab environment; they are real computers that are targeted by cyberattacks. For example, the *binary reputation* data set includes information on binary executables downloaded by users who opt in for Symantec's reputation-based security program, which assigns a reputation score to binaries that are not known to be either benign or malicious. The *antivirus telemetry* data set includes reports about host-based threats (e.g., viruses, worms, trojans) detected by Symantec's antivirus products.

The key idea behind our technique is to identify executable files that are linked to exploits of known vulnerabilities. We start from the public information about disclosed vulnerabilities (i.e., vulnerabilities that have been assigned a CVE identifier), available from vulnerability databases and vendor advisories. We use the public Threat Explorer Web site to determine threats identified by Symantec that are linked to these vulnerabilities, and then we query the antivirus telemetry data set in WINE for the hashes of all the distinct files (malware variants) that are detected by these signatures. Finally, we search the history of binary reputation submissions for these malicious files, which allows us to estimate *when* and *where* they appeared on the Internet.

Using this method, we identified and analyzed 18 vulnerabilities exploited in the real world before disclosure. Our findings include the following:

- Out of these 18 zero-day vulnerabilities, 11 were not previously known to have been employed in zero-day attacks, which suggests that zero-day attacks are more common than previously thought.
- A typical zero-day attack lasts 312 days on average and hits multiple targets around the world; however, some of these attacks remain unknown for up to 2.5 years.
- After these vulnerabilities are disclosed, the volume of attacks exploiting them increases by up to five orders of magnitude.

These findings have important technology and policy implications. The challenges for identifying and analyzing elusive threats, such as zero-day attacks, emphasize that experiments and empirical studies in cybersecurity must be conducted at scale by taking advantage of the resources that are available for this purpose, such as the WINE platform. This will allow researchers and practitioners to investigate mitigation techniques for these threats based on empirical data rather than on anecdotes and back-of-the-envelope calculations. For example, the fact that zero-day attacks are rare events, but that the new exploits are reused for multiple targeted attacks, suggests that techniques for assigning reputation based on the prevalence of files [3] can reduce the effectiveness of the exploit. Furthermore, because we quantify the increase in the volume of attacks after vulnerability disclosures, we provide new data for assessing the overall benefit to society of the *full disclosure* policy, which calls for disclosing new vulnerabilities publicly, even if patches are not yet available.

What Is a Zero-Day Attack?

In general, a zero-day attack is an attack that exploits vulnerabilities not yet disclosed to the public. The origins of this term are unclear. Accounts of events from World War II often use "zero day " when referring to the date set for a planned attack or the day when the attack actually occurred:

> October 20 [1943] was fixed as zero day for [V2] rocket attacks [on London] to begin.
>
> —Winston Churchill, *Closing the Ring,* 1951.

In the computer world, the term is used in the warez community when referring to any copyrighted work (e.g., software, movies, music albums) that is cracked, copied, and re-released on the same day as the official release. Additionally, naming a folder "0day" placed it at the top of the list on a file sharing server as an attempt to draw attention to the item, because zero-day warez is usually sought after by downloaders.

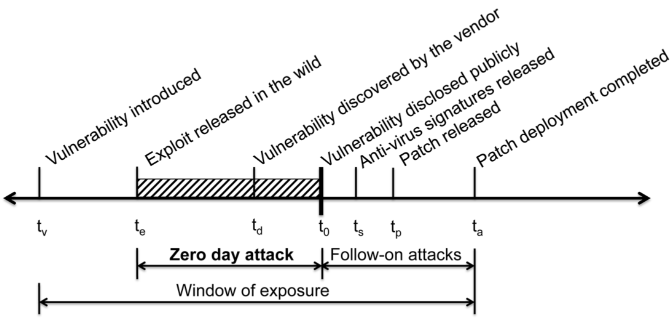
Vulnerability introduced
Exploit released in the wild
Vulnerability discovered by the vendor
Vulnerability disclosed publicly
Anti-virus signatures released
Patch released
Patch deployment completed
t_v
t_e
t_d
t_0
t_s
t_p
t_a
Zero day attack
Follow-on attacks
Window of exposure

Figure 1: Attack timeline. These events do not always occur in this order, but $t_a > t_p \geq t_d > t_v$ and $t_0 \geq t_d$. The relation between t_d and t_e cannot be determined in most cases. For a zero-day attack, $t_0 > t_e$.

When the "zero-day" qualifier was first applied to software vulnerabilities and what the original meaning was is unclear. Today, a "zero-day attack" usually is understood to be a cyberattack that exploits a vulnerability before the vulnerability's public disclosure date (rather than on the same day as the disclosure). These exploits correspond to vulnerabilities that the security community is generally unaware of, and such vulnerabilities are called "zero-day vulnerabilities." This is illustrated in the vulnerability timeline from Figure 1: a vulnerability is created when an exploitable programming bug is introduced in a popular software product (t_v), the vulnerability is then discovered by attackers and is exploited in the wild for conducting stealthy attacks (t_e), the vulnerability is discovered by the vendor (t_d) and disclosed to the public (t_0), leading to the release of countermeasures such as antivirus signatures for the exploit (t_s) and patches for the vulnerability (t_p). The vulnerability ceases to present a threat only when the patch deployment is completed (t_a). The disclosure dates (t_0) of vulnerabilities are tracked and recorded in several public databases, such as Common Vulnerabilities and Exposures (CVE). A zero-day attack is characterized by a vulnerability that is exploited in the wild before it is disclosed, i.e., $t_0 > t_e$. Our goals are to estimate te for, to measure the prevalence and duration of zero-day attacks, and to compare the impact of zero-day vulnerabilities before and after t_0.

Software vendors fix bugs and patch vulnerabilities in all their product releases, and as a result some vulnerabilities are never exploited or disclosed. We only consider vulnerabilities that have a CVE identifier. Similarly, in some cases vendors learn about a vulnerability before it is exploited, but consider it low priority, and cybercriminals may also delay the release of exploits until they identify a suitable target, to prevent the discovery of the vulnerability. Although the CVE database sometimes indicates when vulnerabilities were reported to the vendors, generally, determining the exact date when the vendor or the cybercriminals discovered the vulnerability or even which discovery came first is impossible. Moreover, some exploits are not employed for malicious activities before the disclosure date and are disseminated as proofs-of-concept, to help the software vendor understand the vulnerability and the antivirus vendors to update their signatures. We therefore focus on exploits that have been used in real-world attacks before the disclosure of the corresponding vulnerabilities.

What Do We Know About Zero-Day Attacks?

Most prior work has focused on the entire window of exposure to a vulnerability (see Figure 1), first defined by Schneier [9]. Arbaugh et al. evaluated the number of intrusions observed during each phase of the vulnerability lifecycle and showed that a significant number of vulnerabilities continue to be exploited even after patches become available [1]. Frei compared how fast Microsoft and Apple react to newly disclosed vulnerabilities and, although significant differences exist between the two vendors, both have some vulnerabilities with no patch available 180 days after disclosure [4]. A Secunia study showed that 50% of Windows users were exposed to 297 vulnerabilities in a year and that patches for only 65% of these vulnerabilities were available at the time of their public disclosure [5].

Although the market for zero-day vulnerabilities has not been studied as thoroughly as other aspects of the underground economy, the development of exploits for such vulnerabilities is certainly a profitable activity. For example, several security firms run programs, such as HP's Zero Day Initiative and Verisign's iDefense Vulnerability Contributor Program, that pay developers up to $10,000 for their exploits [7], with the purpose of developing intrusion-protection filters against these exploits. Between 2000 and 2007, 10% of vulnerabilities were disclosed through these programs [4]. Similarly, software vendors often reward the discovery of new vulnerabilities in their products, offering prizes up to $60,000 for exploits against targets that are difficult to attack, such as Google's Chrome browser [6]. Moreover, certain firms and developers specialize in selling exploits to confidential clients on the secretive, but legal, market for zero-day vulnerabilities. Sources from the intelligence community suggest that the market value of such vulnerabilities can reach $500,000 [7]. In particular, the price of exploits against popular platforms, such as Windows, iOS, or the major Web browsers, may exceed $100,000, depending on the complexity of the exploit and on how long the vulnerability remains undisclosed.

Identifying Zero-Day Attacks Automatically

To identify zero-day attacks automatically, we analyzed the historical information provided by multiple data sets. We conducted our study on the Worldwide Intelligence Network Environment (WINE), a platform for data-intensive experiments in cybersecurity [8]. WINE was developed at Symantec Research Labs for sharing comprehensive field data with the research community. WINE samples and aggregates multiple terabyte-size data sets, which Symantec uses in its day-to-day operations, with the aim of supporting open-ended experiments at scale. The data included in WINE is collected on a representative subset of the hosts running Symantec products, such as Norton Antivirus. These hosts do not represent honeypots or machines in an artificial lab environment; they are real computers, in active use around the world, that are targeted by cyberattacks. WINE also enables the reproduction of prior experimental results by archiving the reference data sets that researchers use and by recording information on the data collection process and on the experimental procedures employed.

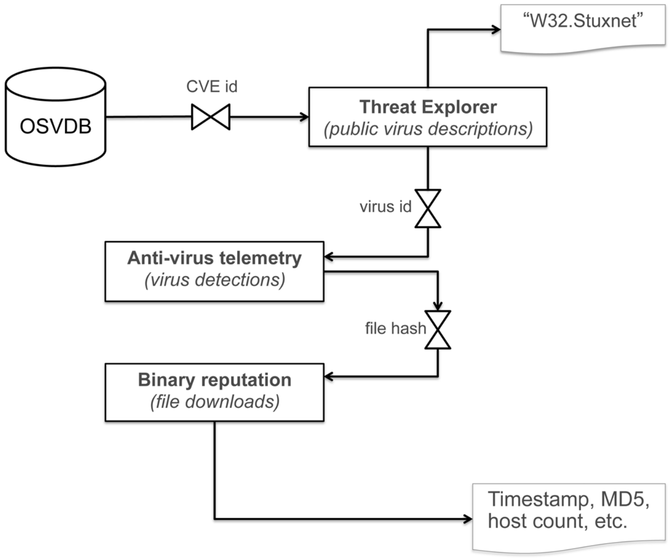

Figure 2: Overview of our method for identifying zero-day attacks systematically

We analyzed two WINE data sets: *antivirus telemetry* and *binary reputation*. The antivirus telemetry data records detections of known threats for which Symantec generated a signature that was subsequently deployed in an antivirus product. The antivirus telemetry data was collected between December 2009 and August 2011, and it includes 225 million detections that occurred on 9 million hosts. The binary reputation data reports all the binary executables—whether benign or malicious—that have been downloaded on end-hosts around the world. The binary reputation data has been collected since February 2008, and it includes 32 billion reports about approximately 300 million distinct files, which were downloaded on 11 million hosts. These files may include malicious binaries that were not detected at the time of their download because the threat was unknown. We note that this data is collected only from the Symantec customers who gave their consent to share it.

We correlated the WINE data sets with information from two additional sources: the Open Source Vulnerability Database

(OSVDB) and Symantec's Threat Explorer. OSVDB is a public vulnerability database, similar to CVE, providing information on the discovery, disclosure, and exploit release date of the vulnerabilities. Threat Explorer is a public Web site with historical information about most threats for which Symantec has generated antivirus signatures—including signatures for exploits of vulnerabilities with known CVE identifiers. Because the Microsoft Windows platform has been the main target for cyberattacks over the past decade, we focus on vulnerabilities in Windows or in software developed for Windows.

Analysis Method

Figure 2 illustrates our four-step analysis method. We start from the known vulnerabilities recorded in OSVDB, and we search Symantec's Threat Explorer for the CVE numbers of these vulnerabilities in order to identify the names of the viruses or worms that exploit them. We manually filter out the generic virus detections (e.g., "Trojan horse") listed on Threat Explorer, to compile a list of threat names that identify vulnerability exploits. We then search for these threat names in the antivirus telemetry, and we record the MD5 and SHA2 hashes of the exploits detected in the field. Having identified which executables exploit known CVE vulnerabilities, we search for each executable in the binary reputation data to estimate when they first appeared on the Internet. If at least one of these executables was downloaded before the disclosure date of the corresponding vulnerability, we conclude that we have identified a zero-day attack. More information about this analysis method is available in the conference version of this article [2].

Threats to Validity

As WINE does not include data from hosts without Symantec's antivirus products, our results may not be representative of the general population of platforms in the world. Although we cannot rule out the possibility of selection bias, the large size of the population in our study (11 million hosts and 300 million files) and the number of zero-day vulnerabilities we identified using our automated method (18, which is on the same order of magnitude as the 43 reported by Symantec analysts during the same period) suggest that our results have a broad applicability. Moreover, for the zero-day vulnerabilities detected toward the beginning of our data collection period, we may underestimate the duration of the attacks. We therefore caution the reader that our results for the duration of zero-day attack are best interpreted as *lower bounds.*

Analysis Results and Findings

Using this method, we identified 18 zero-day vulnerabilities: 3 disclosed in 2008, 7 in 2009, 6 in 2010, and 2 in 2011. From the annual vulnerability trends reports produced by Symantec and the SANS Institute, as well as blog posts on the topic of zero-day vulnerabilities, we found out that seven of our vulnerabilities are generally accepted to be zero-day vulnerabilities (see Table 1). For example, CVE-2010-2568 is one of the four zero-day vulnerabilities exploited by Stuxnet, and it is known to have also been employed by another threat for more than two years before the disclosure date (July 17, 2010). As shown in Table 1, most of these vulnerabilities affected Microsoft and Adobe products. More information about the zero-day vulnerabilities identified is available in [2].

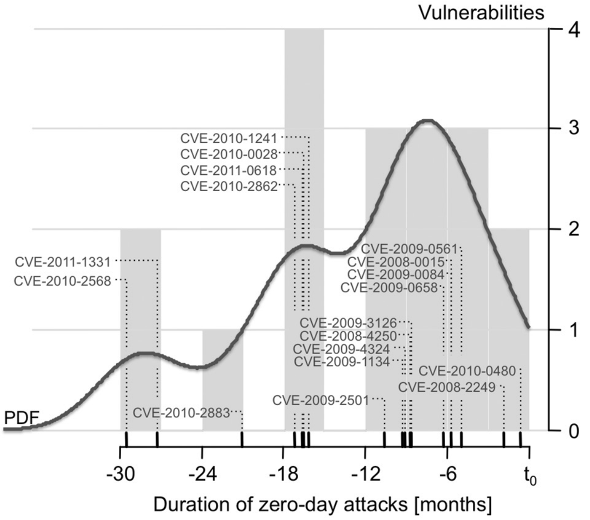

Figure 3: Duration of zero-day attacks. The histograms group attack durations in three-month increments, before disclosure, and the red rug indicates the attack duration for each zero-day vulnerability.

Figure 3 illustrates the duration of zero-day attacks and their distribution: they lasted between 19 days (CVE-2010-0480) and 30 months (CVE-2010-2568), and the average duration of a zero-day attack was 312 days. Figure 3 illustrates this distribution. Fifteen of the zero-day vulnerabilities targeted fewer than 1,000 hosts, out of the 11 million hosts in our data set. On the other hand, three vulnerabilities were employed in attacks that infected thousands or even millions of Internet users. For example, Conficker exploiting the vulnerability CVE-2008-4250 managed to infect approximately 370,000 machines without being detected for more than two months. This example illustrates the effectiveness of zero-day vulnerabilities for conducting stealth cyberattacks.

Zero-Day Vulnerabilities After Disclosure

We also analyzed the increase in the number of malware variants exploiting these vulnerabilities over time. Figure 4 shows that, after the vulnerability disclosure, 183–85,000 more variants are recorded each day. One reason for observing the large number of new different files that exploit the zero-day vulnerabilities might be that they are repacked versions of the same exploits; however, it is doubtful that repacking alone can account

0-day vulnerability	Unknown	Description
CVE-2008-0015		Microsoft ATL Remote Code Execution Vulnerability (RCEV)
CVE-2008-2249	Yes	Microsoft Windows GDI WMF Integer Overflow Vulnerability
CVE-2008-4250	Yes	Windows Server Service NetPathCanonicalize() Vulnerability
CVE-2009-0084	Yes	Microsoft DirectX DirectShow MJPEG Video Decompression RCEV
CVE-2009-0561	Yes	Microsoft Excel Malformed Record Object Integer Overflow
CVE-2009-0658		Adobe Acrobat and Reader PDF File Handling JBIG2 Image RCEV
CVE-2009-1134	Yes	Microsoft Office Excel QSIR Record Pointer Corruption Vulnerability
CVE-2009-2501		Microsoft GDI+ PNG File Processing RCEV
CVE-2009-3126	Yes	Microsoft GDI+ PNG File Integer Overflow RCEV
CVE-2009-4324		Adobe Reader and Acrobat newplayer() JavaScript Method RCEV
CVE-2010-0028	Yes	Microsoft Paint JPEG Image Processing Integer Overflow
CVE-2010-0480	Yes	Microsoft Windows MPEG Layer-3 Audio Decoder Buffer Overflow Vulnerability
CVE-2010-1241	Yes	NITRO Web Gallery 'PictureId' Parameter SQL Injection Vulnerability
CVE-2010-2568		Microsoft Windows Shortcut 'LNK/PIF' Files Automatic File Execution Vulnerability
CVE-2010-2862	Yes	Adobe Acrobat and Reader Font Parsing RCEV
CVE-2010-2883		Adobe Reader 'CoolType.dll' TTF Font RCEV
CVE-2011-0618	Yes	Adobe Flash Player ActionScript VM Remote Integer Overflow Vulnerability
CVE-2011-1331		JustSystems Ichitaro Remote Heap Buffer Overflow Vulnerability

Table 1: New zero-day vulnerabilities discovered and their descriptions

for an increase by up to five orders of magnitude. More likely, this increase is the result of the extensive reuse of field-proven exploits in other malware.

Figure 5 shows the time elapsed until all the vulnerabilities disclosed between 2008 and 2011 started being exploited in the wild. Exploits for 42% of these vulnerabilities appear in the field data within 30 days after the disclosure date. This illustrates that the cybercriminals watch closely the disclosure of new vulnerabilities, in order to start exploiting them, which causes a significant risk for end-users.

Other Zero-Day Vulnerabilities

Every year, Symantec analysts prepare an "Internet Security Threats Report" (ISTR) in which new threats, vulnerabilities, and malware trends are reported. This report includes information about the zero-day vulnerabilities identified during the previous year. These reports identify 43 between 2008 and 2011: 9 in 2008, 12 in 2009, 14 in 2010, and 8 in 2011. For each year, our automated method discovers on average three zero-day vulnerabilities that were not known before and on average two zero-day vulnerabilities from the list reported by Symantec; however, we were not able to identify on average eight known zero-day vulnerabilities per year; these vulnerabilities are linked to Web attacks, polymorphic malware, non-executable exploits, or targeted attacks [2], which illustrates the current limitations of our method and suggests interesting directions for future research.

Discussion

Zero-day attacks are difficult to prevent because they exploit unknown vulnerabilities, for which there are no patches and no antivirus or intrusion-detection signatures. As long as software will have bugs and the development of exploits for new vulnerabilities will be a profitable activity, we will be exposed to zero-day attacks, it seems. In fact, 60% of the zero-day vulnerabilities we identify in our study were not known before, which suggests that there are many more zero-day attacks than previously thought—perhaps more than twice as many; however, reputation-based technologies, which assign a score to each file based on its prevalence in the wild and on a number of other inputs [3], single out rare events such as zero-day attacks and can reduce the effectiveness of the exploits.

The large fraction of new zero-day vulnerabilities we identify also emphasizes that zero-day attacks are difficult to detect through manual analysis, given the current volume of cyberattacks. Automated methods for finding zero-day attacks in field data, such as the method we propose in this paper, facilitate

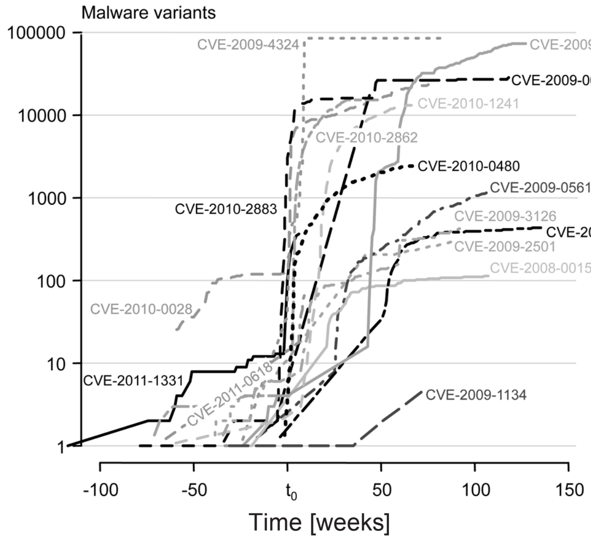

Figure 4: Increase in the number of malware variants exploiting zero-day vulnerabilities after they are disclosed (at time = t_0)

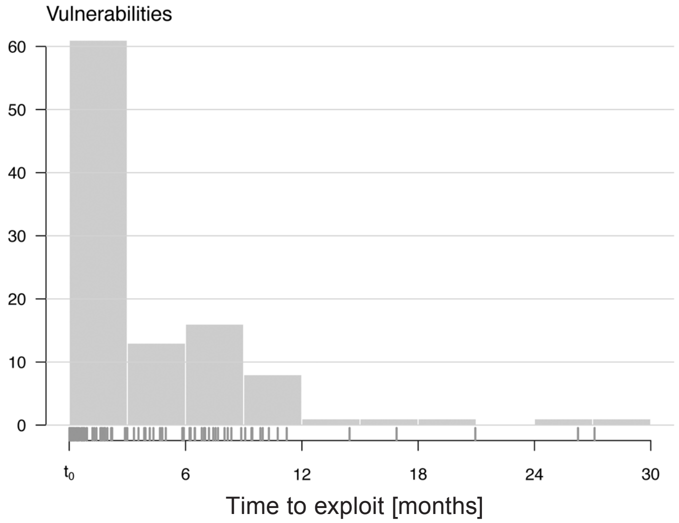

Figure 5: Time before vulnerabilities disclosed between 2008–2011 started being exploited in the field. The histograms group the exploitation lag in three-month increments, after disclosure, and the red rug indicates the lag for each exploited vulnerability. The zero-day attacks are excluded from this figure.

the systematic study of these threats. For example, our method allows us to measure the duration of zero-day attacks (Figure 3). While the average duration is approximately 10 months, the fact that all but one of the vulnerabilities disclosed after 2010 remained unknown for more than 16 months suggests that we may be underestimating the duration of zero-day attacks, as the data we analyze goes back only to February 2008. In the future, such automated techniques will allow analysts to detect zero-day attacks faster, e.g., when a new exploit is reused in multiple targeted attacks; however, this will require establishing mechanisms for organizations to share information about suspected targeted attacks with the security community.

Our findings also provide new data for the debate on the benefits of the full disclosure policy. This policy is based on the premise that disclosing vulnerabilities to the public, rather than to the vendor, is the best way to fix them because this provides an incentive for vendors to patch faster, rather than to rely on security-through-obscurity [9]. This debate is ongoing, but most participants agree that disclosing vulnerabilities causes an increase in the volume of attacks. Indeed, this is what the supporters of full disclosure are counting on, to provide a meaningful incentive for patching; however, the participants to the debate disagree about whether trading off a high volume of attacks for faster patching provides an overall benefit to the society.

The root cause of these disagreements lies in the difficulty of quantifying the real-world impact of vulnerability disclosures and of patch releases without analyzing comprehensive field data. We took a first step toward this goal by showing that the disclosure of zero-day vulnerabilities causes a significant risk for end-users, as the volume of attacks increases by up to five orders of magnitude; however, vendors prioritize which vulnerabilities they patch, giving more urgency to vulnerabilities that are disclosed or about to be disclosed. For example, 80% of the vulnerabilities in 2007 were discovered more than 30 days before the disclosure date [4]. At the same time, anecdotal evidence suggests that attackers also adapt their strategies to the expected disclosure of zero-day vulnerabilities. Because early disclosure reduces the value of zero-day vulnerabilities, the fees for new exploits are sometimes paid in installments, with each subsequent payment depending on the lack of a patch [7]. Additional research is needed for quantifying these aspects of the full disclosure tradeoff, e.g., by measuring how quickly vulnerable hosts are patched in the field, following vulnerability disclosures. Like our study of zero-day attacks, answering these additional research questions will require empirical studies conducted at scale, using comprehensive field data.

Conclusion

Zero-day attacks have been discussed for decades, but no study has yet measured the duration and prevalence of these attacks in the real world *before the disclosure of the corresponding vulnerabilities.* We take a first step in this direction by analyzing field data collected on 11 million Windows hosts over a period of four years. The key idea in our study is to identify executable files that are linked to exploits of known vulnerabilities. By searching for these files in a data set with historical records of files downloaded on end-hosts around the world, we systematically identify zero-day attacks and we analyze their evolution in time.

We identified 18 vulnerabilities exploited in the wild before their disclosure, of which 11were not previously known to have been employed in zero-day attacks. Zero-day attacks last on average 312 days, and up to 30 months, and they typically affect few hosts; however, there are some exceptions for high profile attacks, such as Conficker and Stuxnet, which we respectively detected on hundreds of thousands and millions of the hosts in our study, before the vulnerability disclosure. After the disclosure of zero-day vulnerabilities, the volume of attacks exploiting them increases by up to five orders of magnitude. These findings have important implications for future security technologies and for public policy.

Acknowledgments

We thank Jonathan McCune and Michael Hicks for stimulating discussions on the topic of zero-day attacks. We also thank Marc Dacier for his early feedback on our results, and Sam Perl from CERT for sharing his research into the origins of the term “zero-day attack.” Finally, this research would not have been possible without the WINE platform, built and made available to the research community by Symantec. Our results can be reproduced by utilizing the reference data set WINE 2012-003, archived in the WINE infrastructure.

References

[1] W.A. Arbaugh, W.L. Fithen, and J. McHugh, “Windows of Vulnerability: A Case Study Analysis,” *IEEE Computer,* vol. 33, no. 12, December 2000.

[2] L. Bilge and T. Dumitras, “Before We Knew It: An Empirical Study of Zero-Day Attacks in the Real World,” ACM Conference on Computer and Communications Security, Raleigh, NC, October 2012.

[3] D.H.P. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos, “Polonium : Tera-Scale Graph Mining for Malware Detection,” SIAM International Conference on Data Mining (SDM), Mesa, AZ, April 2011.

[4] S. Frei, “Security Econometrics: The Dynamics of (In) Security,” Ph.D. thesis, ETH Zürich, 2009.

[5] S. Frei, “End-Point Security Failures, Insight Gained from Secunia PSI Scans,” Predict Workshop, February 2011.

[6] Google Inc., “Pwnium: Rewards for Exploits,” February 2012: http://blog.chromium.org/2012/02/pwnium-rewards-for-exploits.html.

[7] A. Greenberg, “Shopping for Zero-Days: A Price List for Hackers’ Secret Software Exploits,” Forbes, March 23, 2012: http://www.forbes.com/sites/andygreenberg/2012/03/23/shopping-for-zero-days-an-price-list-for-hackers-secret-software-exploits/.

[8] T. Dumitras and D. Shou, “Toward a Standard Benchmark for Computer Security Research: The Worldwide Intelligence Network Environment (WINE),” EuroSys BADGERS Workshop, Salzburg, Austria, April 2011.

[9] B. Schneier, “Full Disclosure and the Window of Exposure,” September 2000: http://www.schneier.com/crypto-gram-0009.html.