The Heavy Tails of Vulnerability Exploitation*

Luca Allodi

DISI - University of Trento
Via Sommarive 9, Povo, TN, Italy
luca.allodi@unitn.it

Abstract. In this paper we analyse the frequency at which vulnerabilities are exploited in the wild by relying on data collected worldwide by Symantec's sensors. Our analysis comprises 374 exploited vulnerabilities for a total of 75.7 Million recorded attacks spanning three years (2009-2012). We find that for some software as little as 5% of exploited vulnerabilities is responsible for about 95% of the attacks against that platform. This strongly skewed distribution is consistent for all considered software categories, for which a general take-away is that less than 10% of vulnerabilities account for more than 90% of the attacks (with the exception of pre-2009 Java vulnerabilities). Following these findings, we hypothesise vulnerability exploitation may follow a Power Law distribution. Rigorous hypothesis testing results in neither accepting nor rejecting the Power Law Hypothesis, for which further data collection from the security community may be needed. Finally, we present and discuss the *Law of the Work-Averse Attacker* as a possible explanation for the heavy-tailed distributions we find in the data, and present examples of its effects for Apple Quicktime and Microsoft Internet Explorer vulnerabilities.

1 Introduction

Many natural phenomena have been observed to follow heavy-tailed distributions: some notable examples are the frequency distribution of words in a language, the density of metropolitan areas, and the topology of the Internet. Heavy-tailed phenomena significantly differ from 'usual' phenomena that can be easily described by a few point estimations of the distribution. For example, one may consider life-expectancy in a particular country as a quantity that varies only little with respect to the average. In this sense, the average and the standard deviation of the distribution are enough to 'give an idea' of how the distribution

* The author would like to thank Prof. Fabio Massacci at the University of Trento, Julian Williams at the University of Durham (UK) and Matthew Elder at Symantec Corp. for their many useful comments. This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 285223 (SECONOMICS). This work is also supported by the Italian PRIN Project TENACE. Our results can be reproduced by utilizing the reference data set WINE-2012-008, archived in the WINE infrastructure.

looks like. For heavy-tailed distributions this does not necessarily hold. For example, if one considers GDP worldwide, the infamous Pareto law kicks in (also known as the *80-20 rule*): 20% of the world population owns 80% of the wealth. In this case, the average income does not provide any real indication of how the distribution looks like, as the top 20% of the population is orders of magnitude richer than the remaining 80%. These distributions are often interesting as they are typically generated by complex phenomena lying behind the observation.

In this paper we provide clear evidence that vulnerability exploitation is described by a heavy-tailed distribution and hypothesise that the distribution may follow a Power Law model. We compare our Power Law hypothesis with two additional candidate models for the data: a Log-Normal hypothesis and an Exponential hypothesis. We proceed by rigorously comparing each generating model against the data, following the methodology described in [8]. We find that the negative exponential distribution hypothesis is ruled out, and that both the power law and the log-normal distribution can be suitable models for the data. These results are in line with those of previous research on malware arrival timings [15], and point toward more research to further investigate the process that generates the observed data.

The results presented in this paper have three main implications:

1. Vulnerability exploitation may be described by laws similar to those followed by natural phenomena (like earthquakes) and self-organizing structures (like cities). In this sense, much in the same way as most earthquakes do not represent a threat for the population, most vulnerabilities may carry negligible risk. This indicates that the classical approach *'I have a vulnerability'* $\rightarrow$ *'I must fix it'* may be a largely disproportionate reaction to the real threat. An equivalent to this would be to completely evacuate an area typically affected by earthquakes even if the almost totality of earthquakes does not represent a threat for the population.
2. Commonly-used, industry standard definitions of vulnerability risk based on a single number (e.g. scores assigned by security-testing tools) may be incapable of describing the distribution of attacks: a point estimate (a score or an average) is not enough to describe the phenomena and may lead to substantial overspending / misallocation of resources as most events may be orders of magnitude away from the point estimate.
3. A deeper understanding of the attack-generating process may be needed to explain the clear effect we show in the data. In Section 7 we propose the *Law of the Work-Averse Attacker* as a first, informal attempt to explain the heavy-tail effect we observe.

The paper continues as follows: Section 2 introduces the dataset used for the analysis. Section 3 presents *prima facie* evidence of the heavy-tailed distribution of attacks. The paper continues by introducing the models considered for the data (Section 4) and by presenting the methodology and its limitations (Section 5). Results are presented in Section 6. We then discuss this work's implications and present a first attempt to explain the observed effect (Section 7). Related work is discussed in Section 8. Finally, Section 9 concludes the paper.

Table 1. Categories for vulnerability classification and respective number of vulnerabilities and attacks recorded in WINE.

Category	Sample of Software names	No. Vulns.	Attacks (Millions)
PLUGIN	Acrobat reader, Flash Player	86	24.75
PROD	Microsoft Office, Eudora	146	3.16
WINDOWS	Windows XP, Vista	87	47.3
BROWSER	Internet Explorer, Firefox	55	0.55
Tot		**374**	**75.76**

2 Data collection

Symantec runs a data sharing program, the Worldwide Intelligence Network Environment, or WINE in short[1]. The intrusion-prevention telemetry dataset within WINE provides information about network-based attacks detected by Symantec's products. WINE is indexed by *attack signatures IDs*, unique identifiers for an attack detected by the firm's security solutions, which can be linked to the affected CVE, if any, through Symantec's *Security Response*[2] dataset. Further details on the collection process are available in [3]. This experiment's data is referenced and available for sharing at Symantec Research Labs under the WINE Experiment ID *WINE-2012-008*.

We take additional precautions in handling the data to consider for the fact that the prevalence of an attack may depend on the affected software's exposure to attacks. For example, browsers may be mainly exposed to web attacks, while productivity software like MS Outlook may be targeted more often through social engineering and malicious email attachments. We inspected WINE's vulnerabilities and grouped them in eight software categories: Browser, Plugins, Windows, Productivity, Other Operating Systems, Server, Business Software, Development Software. Because WINE consists largely of data from Symantec's consumer security products, we may have a self-selection problem in which certain software categories are not well represented in our sample. We therefore limit our analysis to the first four categories, for which we consider our sample to be representative of exploits in the wild: BROWSER, PLUGIN, WINDOWS and PROD(uctivity). A more detailed discussion on this rationale is given in [5, 3]. Our analysis comprises 374 vulnerabilities and 75.7 Million attacks recorded from July 2009 to December 2012. Table 1 reports the identified categories and the number of respective vulnerabilities in WINE.

3 The heavy tails of vulnerability exploitation

To visualize the heavy tail distribution effect, we report in Figure 1 the histogram distribution of the (logarithmic) attack volumes for each vulnerability in the category (top row) and the respective Lorentz curve distribution (bottom

[1] `https://www.symantec.com/about/profile/universityresearch/sharing.jsp`
[2] `https://www.symantec.com/security_response/`

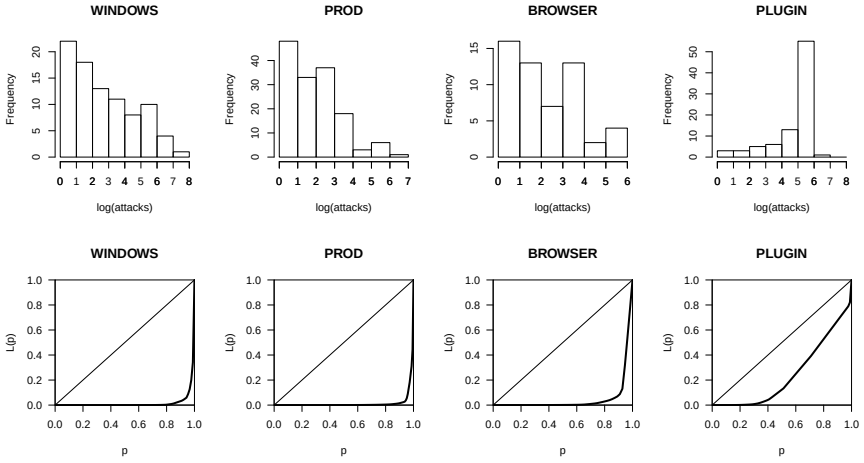

Fig. 1. Top row: histogram distribution of logarithmic exploitation volumes. Bottom row: Lorentz curves for exploitation volumes in the different categories. p % of the vulnerabilities are responsible for L(p)% of the attacks.

row). The histogram distribution clearly shows that (PLUGIN being an exception we further investigate in Section 6) for WINDOWS, PROD and BROWSER the frequency of vulnerabilities with x attacks is inversely proportional to the logarithm of x. In other words, a (very) small fraction of vulnerabilities is responsible for orders of magnitude more attacks than the remaining vulnerabilities.

A clear way to visualize this is through a Lorentz curve. A Lorentz curve describes the p percentage of the population (of vulnerabilities) that are responsible for the $L(p)$ percent of attacks. The diagonal represents an 'equilibrium state' where each vulnerability is responsible for the same volume of attacks. The further away the two curves are, the higher the 'disparity' in the distribution of attacks per vulnerability. As depicted in Figure 1, for WINDOWS, PROD and BROWSER the two curves are very markedly apart, indicating that the great majority of vulnerabilities are responsible for only a negligible fraction of the risk in the wild. Table 2 reports the distribution of attacks recorded in the wild per vulnerability. We report the top 20, 10 and 5 percent of vulnerabilities and the percentage of attacks in the wild they are responsible for. The most extreme results are obtained for WINDOWS and PROD, for which the top 5% vulnerabilities carry more than 90% of the attacks and the top 10% the almost totality. 'Milder' results are obtained for BROWSER: the top 10% carries 90% of the attacks, but the top 5% carries 'only' 68%, meaning that among the top 10% vulnerabilities attacks are distributed more equally than in other categories. The less extreme result is obtained for PLUGIN, where the distribution of exploitation attempts seems more equally distributed among vulnerabilities.

Table 2. $p\%$ of vulnerabilities responsible for $L(p)\%$ of attacks, reported by software category.

Category	Top $p\%$ vulns.	$L(p)\%$ of attacks
WINDOWS	20%	99.6%
	10%	96.5%
	5%	91.3%
PROD	20%	99.5%
	10%	98.3%
	5%	94.4%
BROWSER	20%	97.1%
	10%	91.3%
	5%	68.2%
PLUGIN	20%	46.9%
	10%	31%
	5%	24%

With this last exception, we observe that a general rule for vulnerability exploitation is that, within any software category, less than 10% of attacked vulnerabilities are responsible for more than 90% of the attacks.

4 Possible models for the data

In general, when looking at empirical data it is often difficult to find a perfect fit for a specific distribution. The most cautious way to proceed in this case is to compare different hypotheses against the data. In the heavy-tailed case, models commonly considered as candidates for the data are the *Power Law distribution*, the *Log-Normal distribution*, and the *Exponential distribution* [20].

4.1 Power Law distribution

A power-law distribution describes a phenomenon whereby the probability of observing an event of size x is proportional to a power of x. Many natural phenomena are known to follow power-law distributions. Earthquakes are a clear example: the probability of observing an earthquake of magnitude x rapidly decreases with the destructiveness of the earthquake[3] [20]. In general, a power law is expressed as:

$$p(x) \sim x^{-\alpha} \tag{1}$$

where x is the measured quantity and α is a *scaling factor* of the distribution. It is easy to see that, if one applies the logarithm on both sides of the equation, one ends up with an equation form of the type $ln(p(x)) = -\alpha ln(x) + c$ which

[3] 'Destructiveness' is expressed by measure of the Richter scale which represents the base 10 logarithm of the maximum amplitude of a wave as detected by a seismograph.

is a straight line with (negative) slope α and intercept c. On a log-log plot a distribution following a power law would therefore follow a straight line.

As to the scaling parameter α, most power-law distributions found in Nature are in the range $2 \leq \alpha \leq 3$ [8]. When describing a power law phenomenon the parameter α has some interesting properties attached to it. By calculating the second and third momentum of the normalized power law distribution it is possible to see that depending on the value assumed by α the distribution may have infinite mean ($\alpha < 2$) and infinite variance / standard deviation ($\alpha < 3$). The interested reader can refer to [20] for further details.

In practical terms, a distribution with infinite mean and variance is a distribution that can not be described by point estimates.

4.2 Log-normal distribution

Log-normal distributions can be thought as emerging from a *multiplicative effect*. [18] suggests this is for example how one can model biological organisms' growth in weight: as a percentage C of the current weight W_t, such as $W_{t+1} = (C \times W_t) + W_t$. This generates a rapidly growing distribution. If growth in each step of the process is randomly distributed and has finite mean and variance, than because of the central limit theorem one ends up with a normal distribution $N(\sigma, \mu)$ defined in the logarithm of the measure. The function form of a log-normal distribution can therefore be derived from a normal distribution. For further details we refer the reader to [18].

A log-normal distribution has always finite mean and variance, which are therefore more meaningful to consider than in the general power law case.

4.3 Exponential distribution

An exponential distribution is often used to describe the probability distribution of the distance between independent events that arrive (on average) at a constant rate. A negative exponential is often a less good alternative model to a power law than a log-normal distribution is [8], but we still consider it here for the sake of completeness.

5 Methodology

The central hypothesis around which we build our analysis is:

Hypothesis 1 *Vulnerability exploitation follows a Power Law distribution.*

Following the methodology indicated in [8], we: 1) estimate the parameters for the hypothesised Power Law; 2) Test the suitability of the Power Law model for the data; 3) Compare the Power Law model with alternative possible explanations (i.e. log-normal and exponential)[4].

[4] We use the statistical tool `R` and the PoweRlaw package [26, 13]. The scripts are available at `https://securitylab.disi.unitn.it/doku.php?id=software`

Parameter Estimation. Empirical data is often noisy; in particular when fitting a power law to it, one may find that the data follows a power law only above a certain threshold x_{min}. This is intuitive as in the lower tail small variations in the magnitude of the observation would cause significant noise in the fit. It is generally observed that data points below x_{min} are often better modelled by distributions other than a power law [18]. Exploiting this observation, Clauset et al. [9] suggest to estimate x_{min} by selecting the cutoff that minimizes the distance between the fitted Power Law distribution and the probability distribution of the data. The distance is calculated as the Kolmogorov-Smirnov (KS) statistic, which simply returns the maximum absolute distance between two curves. This way one obtains the x_{min} cutoff that provides the best fit for all $x > x_{min}$. The scaling parameter α is estimated as the parameter that maximizes the likelihood of observing the data given a certain value of α (*maximum likelihood estimation*).

Hypothesis testing. We now need to estimate how likely the Power Law model is for the data. *Bootstrapping* [11] provides a powerful method to verify the likelihood of the Power Law hypothesis. For each separate data sample DS (e.g. BROWSER) of length n, a *bootstrapped sample* is obtained from the data by randomly choosing with replacement n vulnerabilities from DS. We create 10 thousand bootstrapped samples for each DS. For each bootstrapped sample we then compute the parameter estimation and the relative KS statistic. Then, a *p-value* for the power law hypothesis is obtained by computing the fraction of KS statistics KS' obtained from the sample that are *above* the KS statistic seen from the data. The closest the *p-value* is to the unity, the greatest the evidence for the Power Law Hypothesis[5]. We reject the power-law hypothesis if the resulting p-value is below $p < 0.1$. As noted by Clauset et al., a very good fit ($p > 0.9$) is very unlikely to be found in field data such as ours. We will consider the Power Law model to be *not unreasonable* for $p > 0.1$. This threshold is the same indicated in [8].

Comparison with other models. The p-value alone may not be a good-enough indicator of the models' suitability, especially when the data is noisy. Therefore, to more rigorously evaluate the Power Law Hypothesis we compare it with the alternative distributions defined in Section 4.

To compare the models we perform a *log likelihood test*. The idea behind a log likelihood test is to compute the likelihood of observing the data assuming two different originating models: the model with the highest likelihood is, intuitively, the preferred one. A way to see this is to compute the difference in the log likelihoods for the two distributions, R: if R is close to zero, the data has the same likelihood under the two hypotheses; if R is far from zero, the sign of the difference indicates which model is the most suitable for the data. We compute R as $R = log(L(PowerLaw)) - log(L(Alternative))$ where $L()$ is the likelihood function; therefore, a positive sign favours the Power Law hypothesis; a negative sign favours the Alternative.

[5] An alternative approach would be to measure the fraction of estimated α' from the bootstrapped sample higher than the α for the original data.

Table 3. Power laws' parameters. The reported α is the median resulting from the bootstrapped process. Significance is reported in bold for $p > 0.1$.

Category	x_{min}	$n_{x \geq x_{min}}$	α	95% Con. In.	p-value
WINDOWS	20	64	1.31	1.22 - 1.64	**0.44**
BROWSER	1010	19	1.60	1.20 - 2.27	**0.52**
PLUGIN	118	80	1.35	1.26 - 2.14	0.00
PROD	267	49	1.50	1.34 - 1.78	**0.84**

The significance of the difference between the two models is given by the size of $|R|$. For values of R close to zero, the sign does not indicate a significant difference between the two models. We use the Vuong test [29] to evaluate the statistical significance of the sign. If the resulting p-value is below 0.1 ($p < 0.1$) we consider the difference to be significant. If not, the two models (Power Law and the Alternative) are effectively indistinguishable with respect to the data.

Limitations. Fitting models to the data asks for as many data points as possible. [8] shows that, indicatively, a distribution with at least 100 points is desirable to make sound conclusions. However, a general estimation of this threshold valid for any distribution is hard to make. Unfortunately exploitation data, especially collected on a significant scale (i.e. worldwide), is difficult to find. No precaution can completely rule out the "overfitting" problem caused by too few data points [8]. In our experiment the worst case is that of BROWSER vulnerabilities, for which we refrain from making any definitive conclusion. For WINDOWS, PLUGIN and PROD we will be slightly bolder. To the best of our knowledge, WINE is the most comprehensive dataset of records of attacks in the wild that is publicly available.

Another limitation is represented by the data collection itself and, indirectly, by the type of software and attacks our results can be considered representative of. The WINE dataset reports mostly attacks recorded against 'consumer platforms', unlikely to receive targeted or 0-days attacks [6]. Our results and conclusions are therefore relevant only for untargeted attack scenarios, and are not representative of 'black swan' attacks for which a dedicated attacker aims at a particular target. This is obvious as a single targeted attack is for us negligible, as it would be in the lower left hand of the Lorentz curves in Figure 1.

6 The power law hypothesis

Parameter estimation and Hypothesis Testing. Figure 2 reports the log-log plot and a linear fit for the four categories. Attack volumes are reported on the x-axis; the y-axis reports the probability of observing an attack volume equal to x. It is easy to see that for all categories, with the exception of PLUGIN, the data shows a linear trend, which is expected with a Power Law distribution [8, 18]. Table 3 reports the model parameters for each category and the p-value for the power law fit. The estimation is done for values $x \geq x_{min}$. This means that the datasets are further truncated and the estimation is limited to the

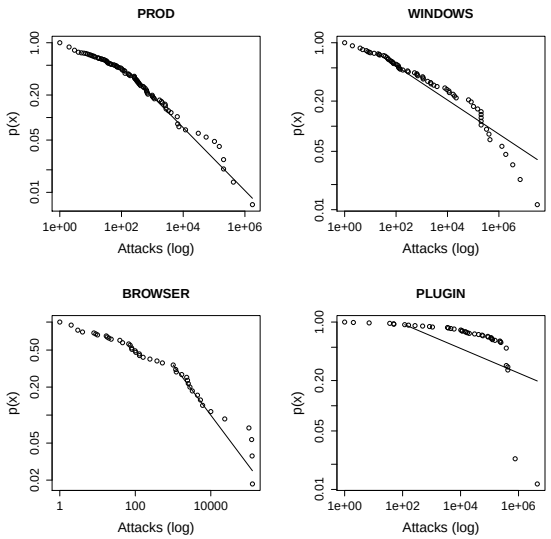

Fig. 2. Log-log plot of vulnerability exploitation by vulnerability rank

datapoints left. These are reported in the table under the column $n_{x \geq x_{min}}$. The data points left still allow for an (albeit cautious) discussion. The BROWSER case is critical, as only 19 vulnerabilities are available for the model fitting. For this category the resulting p-value ($p = 0.52$) is higher than the significance threshold of $p > 0.1$ identified by [8], but we refrain from considering this as evidence for the Power Law case. A more significant discussion can be made for the remaining categories. In particular, the Power Law model could be a good candidate to explain the WINDOWS and PROD exploitation distributions. For PLUGIN as a whole, instead, the hypothesis is *ruled out* completely. We will analyse this exception in more detail later in this Section. The α parameter lies in the 1.2-2.2 region for all software categories, indicating a mildly steep to steep curve.

We *do not reject* Hyp. 1 in the cases of WINDOWS, BROWSER and PROD vulnerabilities. We reject Hyp. 1 for PLUGIN.

Comparison with other models. In Table 4 we report the results of the log likelihood comparison between Hyp. 1 and the alternative models. A negative sign indicates that the evidence points toward the alternative hypothesis; a positive sign supports the Power Law model. We also report the two-tailed Vuong's significance test; the result is considered significant if the p-value is below 0.1. The log likelihood ratio test for the exponential distribution returns "Not a number" as the fit between the estimated curve and the data is so poor it tends to zero, and the logarithm goes to infinity. The log-normal distribution results slightly favored in the likelihood ratio test for BROWSER and PROD vulnerabilities, but the small distance from 0 does not make for a solid margin,

Table 4. Difference in likelihood of alternative models. ∞ indicates a fit so poor that the log likelihood for the alternative goes to infinity. We report significant conclusions for $p \leq 0.1$ in bold.

Category	Alternative	Likelihood difference	Favoured Model
WINDOWS	Log-normal	-1.49	Alternative
	Exponential	∞	**Power Law**
BROWSER	Log-normal	-0.29	Alternative
	Exponential	2.18	**Power Law**
PLUGIN	Log-normal	-5.39	**Alternative**
	Exponential	∞	**Power Law**
PROD	Log-normal	-0.34	Alternative
	Exponential	∞	**Power Law**

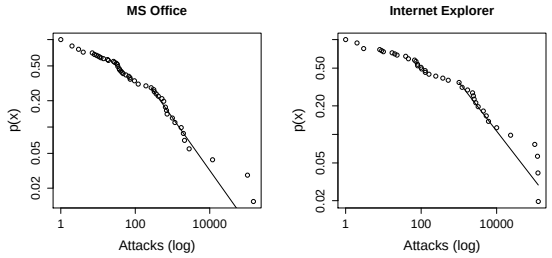

Fig. 3. Log-log plot of volume of exploits in the wild for Microsoft Office (left) and Internet Explorer (right).

as the difference may as well be due to sole chance. In general, we find that a log-normal distribution does not perform significantly better than a Power Law in describing our data. For WINDOWS vulnerabilities the evidence is more markedly toward the log-normal distribution, but the difference is again not significant. The case for PLUGIN is, unsurprisingly, sharply in favor of a log-normal distribution. With the exception of PLUGIN, none of the alternative hypothesis in Table 4 provides a better explanation to the data than a Power Law distribution does.

We now narrow down the data analysis to single instances of 'representative' software in each category. We however do not report any more data-fitting results as the fewer and fewer data points would make their interpretation a particularly tricky one.

6.1 Breakdown by software

Figure 3 reports the distribution of exploitation volume for vulnerabilities affecting Microsoft Office (PROD) and Internet Explorer (BROWSER). For these two software, the log-log plot shows a good linear fit along the data points. The numerical results are equivalent to those reported for the respective macro categories. Software in WINDOWS, not reported here for brevity, also confirms the

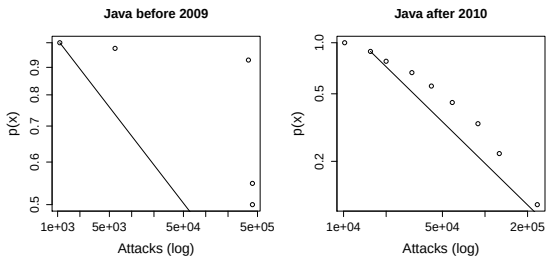

Fig. 4. Log-log plot of volume of exploits in the wild for Java vulnerabilities disclosed before 31-Dec-2009 (left) and after 1-Jan-2010 (rigth).

general result. For PLUGIN software, the Power Law fitting is always very low regardless of the considered software. To further investigate this, in Figure 4 we report the distribution for Java vulnerabilities grouped by year of disclosure. The different distribution in attacks for vulnerabilities disclosed before and after 2009 is immediate to see. While for 2009 the Power Law model is clearly a bad fit, for Java vulnerabilities disclosed after 2010 it is supported by the evidence ($p = 0.76$). Neither the log-normal nor the exponential distribution provide a better model for the data. A possible explanation to this temporal effect is that software running in background (such as PLUGIN software generally is) may be seldom updated by users [30]. This may have an influence on the exploitation volumes recorded: looking at the Java case, 2009 is the last year Java was owned by Sun Microsystems, before being acquired by Oracle. This may suggest that pre-2009 vulnerabilities for Sun Microsystem's Java accumulated high exploitation volumes possibly because of users' latency in switching to Orcale's Java. For Java vulnerabilities disclosed after 2010 our results are equivalent to those we obtained for the other categories. This suggests that the heavy-tail effect we observe is present regardless of the software type.

7 Discussion

In this paper we presented evidence that vulnerability exploitation follows a heavy-tailed distribution.

The heavy-tail effect we find is (qualitatively) similar to that shown by the *80-20 Pareto law* of income distribution: the majority of the impact is caused by a small fraction of the population. We showed that, depending on the type of software affected by the vulnerability, as low as 10% of the vulnerabilities may be responsible for more than 90% of the attacks in the wild against that software. The most extreme result is obtained for PROD vulnerabilities, for which 5% of vulnerabilities account for 95% of the attacks.

This observation alone could have significant impact on the way security quantification and prioritization is done. Vulnerabilities represent a significant source of uncertainty when managing infrastructural and system security. Clearly

all vulnerabilities represent a potential risk, but it is effectively unclear *how much* risk is attached to a software flaw. Many regulatory and administrative initiatives try to give an estimate of this by suggesting simple rules to prioritize vulnerability treatment. Notable examples of this are the NIST SCAP protocol [25] and guidance provided by the PCI DSS standard for credit card security [10]: a high risk score vulnerability is considered on average dangerous enough to need immediate treatment. This approach has already been questioned in literature [4], and our results point in the same direction: point estimates of vulnerability risk may be widely inappropriate in practice.

7.1 An explanation attempt: the Law of the Work-Averse Attacker

We make an attempt at giving an explanation to the possible mechanisms that underlie the heavy-tail effect shown in this paper. We label this the "*Law of the Work-Averse Attacker*", according to which the average attacker is not interested in procuring and using new reliable exploits if he or she already owns one. The rationale behind this is that once the attacker can attack n systems with one exploit, as long as n is high enough a new reliable (and possibly expensive [2, 5]) exploit would not increase n enough to justify the cost (economic or in terms of effort) of deploying a new attack. The effect of this is that attackers focus their efforts in attacking a limited set of vulnerabilities for which reliable exploits exist and are available (for example in the black markets [5]). As a consequence, only a handful of vulnerabilities are consistently attacked over time, and this may generate the heavy-tailed effect shown in this paper. In general,

$$\exists v_{0,t_0}, n : P(v_0, t_0, n) \approx 1 \rightarrow \forall v_i \neq v_0 \; P(v_i, t_0, n) \approx 0 \tag{2}$$

where $P(v, t, n)$ is the probability that an attack against the vulnerability v is successful at time t_0 against n systems, with $n >> 1$.

If this holds, by looking at exploitation trends in time we would expect that:

Hypothesis 2 *Exploits alternate in 'popularity', i.e. a new one appears only when an old one decays.*

Hypothesis 3 *No two exploits are at the same level of exploitation at the same moment.*

In Figure 5 we report as an example the trends of exploitation of vulnerabilities disclosed in 2010 for Internet Explorer and QuickTime. We plot on a logarithmic scale the volume of attacks against each vulnerability, represented by a distinct line. A simple illustrative example is that of Quicktime, for which it is visible how the emergence of the second exploit follows a sharp decline in the popularity of the already-present one. This same effect can also be found in the more complex scenario of Internet Explorer: in 2010 we have three main exploits (a fourth is collapsed several orders of magnitude below the others). Let's call them A (full line, dots), B (dashed lined, squares) and C (short dashes, crosses). We note that:

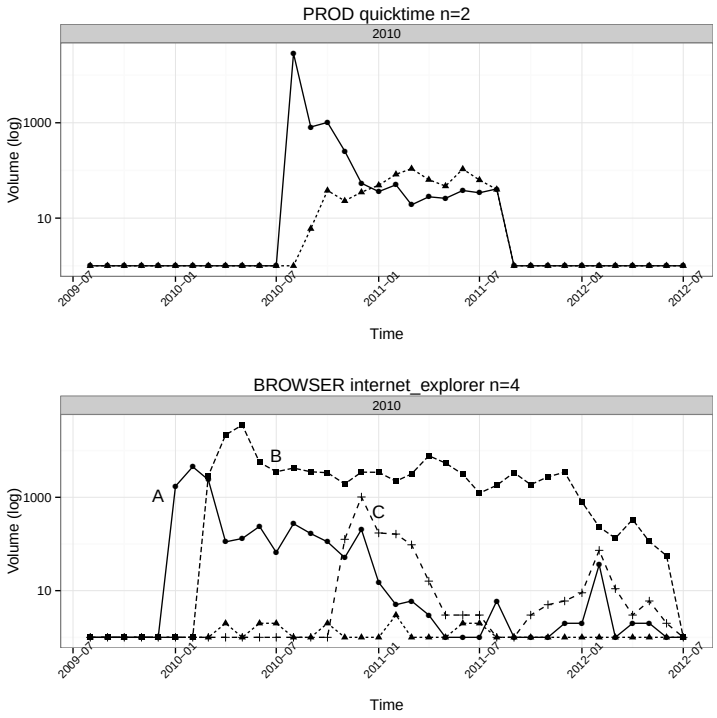

Fig. 5. Trends in attacks for vulnerabilities disclosed in 2010 for QuickTime and Internet Explorer.

- when A falls, B rises
- after a sharp decline in B, C rises
- when B goes up again C falls and A disappears
- when B finally dies, first C rises and then A (since then dead) rises up again

We therefore find supporting evidence for Hyp 2. We find Hyp 3 to be supported as well as in both cases one exploit dominates all the others at least by one order of magnitude. We keep a more precise and formal characterization of this model for future research.

8 Related Work

Shahzad et al. [28] have recently presented a general overview of software vulnerabilities. Many descriptive trends in timings of vulnerability patching and release of proof-of-concept exploits are presented. Frei et al. [12] showed that exploits are often quicker to arrive than patches are. An analysis of the same flavour is provided by [27] and [7]. Other studies focused on the modeling of the vulnerability

discovery processes. Reference works in this area are [1] and [23]. Current vulnerability discovery models are however not general enough to represent trends for all software [21]. Moreover, vulnerability disclosure and discovery are complex processes [7, 22], and can be influenced by {black/white}-hat community activities [7] and economics [17]. The different risk levels coming from different vulnerability types and exploit sources is outlined in [4]. Our study, rather than presenting an overview of vulnerabilities, exploits and patches releases, focuses on volumes of exploitation attempts in the wild.

By analysing attack data in WINE Nayak et al. [19] concluded that attackers focus on few vulnerabilities only and that, as a consequence, risk measurements based solely on knowledge of vulnerability may be inaccurate. Holm [15] analyses attack data on the systems of an organisation and fits it to several models. His analysis concerns the time of arrival of malware alerts. His results are on the same lines as ours: a log-normal distribution and a Pareto distribution are usually a better fit to the data than other models. Differently from [15], we focus on the volume of vulnerability exploitation attempts rather than on the timings of malware detection.

Bilge and Dumitras [6] provide an analysis of 0-day exploits by analysing in hindsight historical records of attacks in WINE. Provos et al. [24] also provide a quantitative estimation of cyber-attacks by analysing iFrame traffic; they find that about 60% of the threats against the final user are web attacks. An analysis of the mechanisms responsible for the generation of these attacks can be found in [14], that uncovers the Exploit-as-a-Service architecture for cyberattacks used by cybercriminals. Following this line of research, an estimation of the fraction of attacks generated by cybercrime market activities is given in [5]. An in-depth, empirical analysis of the tools used by cybercriminals to deliver their attacks is given in [16] and [2]. Rather than focusing on the general volume of attacks affecting the final user, in this work we evaluate how vulnerability exploitation is distributed in the wild.

9 Conclusions

In this paper we analysed the frequency with which vulnerabilities are exploited in the wild. Our findings clearly show that the distribution of attacks follows a heavily tail distribution, showing that a small fraction of vulnerabilities is responsible for the great majority of attacks against a software.

We hypothesise that this distribution may follow a Power Law, but this hypothesis is only inconclusively supported by our evidence: an alternative, equally good explanation to the data may be provided by a log-normal distribution. The statistical power needed to accept one or the other hypothesis is reduced by the relatively low number of vulnerabilities present in our dataset which nonetheless represents, at the best of our knowledge, the most comprehensive collection of attacks in the wild publicly available at the moment of writing.

To further explain our results we present the *Law of the Work-Averse Attacker*, according to which attackers only select one vulnerability to exploit at

a time, per software. This results in a distribution of attacks whereby only one vulnerability out of many represent a relevant risk for the user. This model is qualitatively supported by the evidence we find by analysing two case scenarios for Apple Quicktime and Microsoft Internet Explorer. We leave a more formal analysis of this model to future work.

References

1. Alhazmi, O., Malaiya, Y.: Modeling the vulnerability discovery process. In: Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05). pp. 129–138 (2005)
2. Allodi, L., Kotov, V., Massacci, F.: Malwarelab: Experimentation with cybercrime attack tools. In: Proceedings of the 2013 6th Workshop on Cybersecurity Security and Test (2013)
3. Allodi, L., Massacci, F.: A preliminary analysis of vulnerability scores for attacks in wild. In: Proceedings of the 2012 ACM CCS Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (2012)
4. Allodi, L., Massacci, F.: Comparing vulnerability severity and exploits using case-control studies. ACM Transaction on Information and System Security (TISSEC) 17(1) (August 2014)
5. Allodi, L., Woohyun, S., Massacci, F.: Quantitative assessment of risk reduction with cybercrime black market monitoring. In: In Proceedings of the 2013 IEEE S&P International Workshop on Cyber Crime. (2013)
6. Bilge, L., Dumitras, T.: Before we knew it: an empirical study of zero-day attacks in the real world. In: Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS'12). pp. 833–844. ACM (2012)
7. Clark, S., Frei, S., Blaze, M., Smith, J.: Familiarity breeds contempt: the honeymoon effect and the role of legacy code in zero-day vulnerabilities. In: Proceedings of the 26th Annual Computer Security Applications Conference. pp. 251–260 (2010), http://doi.acm.org/10.1145/1920261.1920299
8. Clauset, A., Shalizi, C.R., Newman, M.E.: Power-law distributions in empirical data. SIAM Review 51(4), 661–703 (2009)
9. Clauset, A., Young, M., Gleditsch, K.S.: On the frequency of severe terrorist events. Journal of Conflict Resolution 51(1), 58–87 (2007), http://jcr.sagepub.com/content/51/1/58.abstract
10. Council, P.: Pci dss requirements and security assessment procedures, version 2.0. (2010), https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf
11. Efron, B., Tibshirani, R.J.: An introduction to the bootstrap, vol. 57. CRC press (1994)
12. Frei, S., May, M., Fiedler, U., Plattner, B.: Large-scale vulnerability analysis. In: Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense. pp. 131–138. ACM (2006)
13. Gillespie, C.S.: Fitting heavy tailed distributions: the poweRlaw package (2013), r package version 0.20.2
14. Grier, C., Ballard, L., Caballero, J., Chachra, N., Dietrich, C.J., Levchenko, K., Mavrommatis, P., McCoy, D., Nappa, A., Pitsillidis, A., Provos, N., Rafique, M.Z., Rajab, M.A., Rossow, C., Thomas, K., Paxson, V., Savage, S., Voelker, G.M.: Manufacturing compromise: the emergence of exploit-as-a-service. In: Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS'12). pp. 821–832. ACM (2012)

15. Holm, H.: A large-scale study of the time required to compromise a computer system. IEEE Transactions on Dependable and Secure Computing 11(1), 2–15 (2014)
16. Kotov, V., Massacci, F.: Anatomy of exploit kits. preliminary analysis of exploit kits as software artefacts. In: Proc. of ESSoS 2013 (2013)
17. Miller, C.: The legitimate vulnerability market: Inside the secretive world of 0-day exploit sales. In: Proceedings of the 6th Workshop on Economics and Information Security (2007)
18. Mitzenmacher, M.: A brief history of generative models for power law and lognormal distributions. Internet Mathematics 1(2), 226–251 (2004)
19. Nayak, K., Marino, D., Efstathopoulos, P., Dumitraş, T.: Some vulnerabilities are different than others. In: Proceedings of the 17th International Symposium on Research in Attacks, Intrusions and Defenses, pp. 426–446. Springer (2014)
20. Newman, M.E.: Power laws, pareto distributions and zipf's law. Contemporary Physics 46(5), 323–351 (2005)
21. Nguyen, V.H., Massacci, F.: An independent validation of vulnerability discovery models. In: Proceeding of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS'12) (2012)
22. Ozment, A.: The likelihood of vulnerability rediscovery and the social utility of vulnerability hunting. In: Proceedings of the 4th Workshop on Economics and Information Security (2005)
23. Ozment, A.: Improving vulnerability discovery models: Problems with definitions and assumptions. In: Proceedings of the 3rd Workshop on Quality of Protection (2007)
24. Provos, N., Mavrommatis, P., Rajab, M.A., Monrose, F.: All your iframes point to us. In: Proceedings of the 17th USENIX Security Symposium. pp. 1–15 (2008)
25. Quinn, S.D., Scarfone, K.A., Barrett, M., Johnson, C.S.: Sp 800-117. guide to adopting and using the security content automation protocol (scap) version 1.0. Tech. rep., National Institute of Standards & Technology (2010)
26. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2013), `http://www.R-project.org`
27. Ransbotham, S.: An empirical analysis of exploitation attempts based on vulnerabilities in open source software. In: Proceedings of the 9th Workshop on Economics and Information Security (2010)
28. Shahzad, M., Shafiq, M.Z., Liu, A.X.: A large scale exploratory analysis of software vulnerability life cycles. In: Proceedings of the 34th International Conference on Software Engineering. pp. 771–781. IEEE Press (2012)
29. Vuong, Q.H.: Likelihood ratio tests for model selection and non-nested hypotheses. Econometrica: Journal of the Econometric Society pp. 307–333 (1989)
30. Wash, R.: Folk models of home computer security. In: Proceedings of the Sixth Symposium on Usable Privacy and Security (2010)