

An Economic Analysis of the Strategic Interaction among Computer Security Attackers

Pei-yu Chen, Gaurav Kataria and Ramayya Krishnan
Carnegie Mellon University

Abstract: It is commonly assumed that there are only two kinds of actors in the field of computer security – ones who defend and others who attack. Network administrators, software developers and anti-virus firms are the ones who defend users, whereas, attackers (hackers) are the ones who breach defenses to violate users' security. This simplified view of computer security sometimes overlooks the strategic interaction between the players involved. Specifically, while it is acceptable to assume that all defenders are brothers-in-arms against all attackers, it may be incorrect to assume that all attackers are acting in concert too, because they do not have a common goal. In fact, attackers often see themselves as playing against each other in a zero-sum-game, where, if one attacker is successful in launching a worm/virus before other attackers, then he can compromise a number of vulnerable computers, thereby depriving other attackers of the possibility of compromising those same computers. To successfully defend oneself, it is important that we understand the incentives of the attackers and the strategic interactions among them. In this paper, we analyze the data on release of computer worms during the past two years to establish the competitive interaction among attackers. Specifically, we investigate 1) the factors that impact the likelihood of a vulnerability being exploited at any given time, and 2) the exploitation patterns over time given a vulnerability. Our preliminary results indicate some "herding" behaviors by the attackers, i.e., attackers are more likely to compete for exploiting certain vulnerabilities than others. In addition, we find that attackers are more likely to reuse technology developed by other attackers than to develop their own; this suggests that it may be advisable to look out for vulnerabilities that have just been exploited than the vulnerabilities that have never been exploited. These findings have important strategic implications and policy consequences.

1. Introduction

The explosive growth of Internet has brought along with it some unexpected challenges. As the size of Internet economy has grown so has the threat of online scams and malicious software like worms and viruses. Symantec, a computer security firm, reported that over twenty thousand new worms and viruses were created in year 2005. This high growth in the computer underworld is indicative of large economic gains that attackers (hackers) are able to realize by attacking home and corporate computers (Symantec 2006). It is reasonable to believe that the high expected rewards might have attracted more attackers into the field making the attack process more competitive. In this paper, we use a novel dataset to test our hypothesis regarding the competitive interaction among attackers. The dataset comprises of information on release of computer worms that exploited vulnerabilities in Microsoft products over a two year period from May 2004 to May 2006. Our findings indicate presence of competitive interaction among attackers.

While there are many studies developing economic models of defender-side of information security (Cavusoglu et al 2004, Arora et al 2004, Gordon et al 2003), to the best of our knowledge, there are no studies modeling the strategic interaction among attackers. We believe that an equally important aspect of information security analysis is to understand the economic incentives of attackers and their attacking behaviors over time. Our research goal is to formally model the interaction among attackers to gain insights into the exploit patterns of future information security threats, so that appropriate security interventions are developed towards minimizing those threats.

2. Model

Though attackers may have many motivations to launch an attack, their attacking process can be

broadly classified into two categories – opportunistic and targeted (Collins et al. 2006). Collins et al. describe opportunistic attacks as those attacks which are indiscriminate in their target, e.g., a computer worm using random scanning to discover all vulnerable computers. In contrast, targeted attacks have a clear premeditated target, e.g. attack on a competitor’s server for industrial espionage purpose. Although, both kinds of attacks use software (or human) vulnerabilities to penetrate a secure system, opportunistic attacks in particular, consider targets indistinguishable except for their vulnerabilities i.e. they are designed with the sole purpose of compromising as many vulnerable computers as possible. These compromised computers, called zombies or bots, are in turn used by attackers to gain economic benefit through stealing confidential data and by illegally using their networking resources for sending spam and/or launching Distributed Denial of Service (DDoS) attacks.

The primary goal of an opportunistic attacker is to compromise as many computers as possible, while expending minimum effort. By being first – among a group of attackers – in exploiting a vulnerability an attacker can expect to compromise a higher proportion of the vulnerable computer population. At the same time, by choosing to attack sooner than later an attacker ensures that a larger proportion of computer population is vulnerable, as with time computer users patch the vulnerabilities. Consequently, we can assume that the rewards of attacking are diminishing in time and in rank order. However, the cost of attacking also diminishes with time and rank order, because attackers who attack later benefit from the experience and the technology developed by earlier attackers. More specifically, due to lack of any intellectual property issues in the computer underworld, attackers freely copy the attack technologies that have been developed by other attackers before them (sometimes with consent, and many times without consent, e.g. worm *Blaster.B* was developed by an attacker whose computer got infected by the original worm *Blaster.A*).

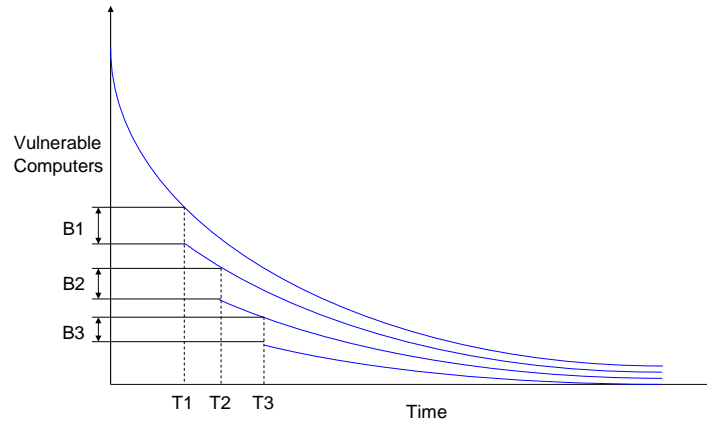


Figure 1: The benefit (computers compromised) by attackers as a function of time and their rank order. T1-T3 are launch times of attacks 1-3, and B1-B3 are the no. of computers compromised.

Given the aforementioned dynamics of attacking process, we develop a model for competitive interaction among opportunistic attackers. The expected benefit to an attacker can be derived as follows. The population of computers that are vulnerable (due to a certain vulnerability in a software) is a declining function of time, assumed to be $M \cdot e^{-rt}$, where M is the original market size of the software product, r is the rate of patching (Figure 1). The attacker who attacks at time t gets a proportion, S (success ratio of an attack), of vulnerable computers available at that time (the attacker gain is assumed to be instantaneous; this assumption is becoming truer as worm propagation and defensive mechanisms – like network filters – are getting quicker). The computers gained by the first attacker is,

$$B1 = S \times M \times e^{-r t_1}$$

Where, r_1 is the initial rate of patching. The gain by second attacker is,

$$B2 = S \times \left((1 - S) \times M \times e^{-r_1 t_1} \right) \times e^{-r_2 (t_2 - t_1)}$$

Where, r_2 is the new patching rate, as computer users may respond by increasing their patching rate. The gain for the i^{th} attacker is,¹

$$\begin{aligned} Bi &= S \times (1 - S)^{i-1} \times M \times e^{-r_1 t_1} \times e^{-r_2 (t_2 - t_1)} \times \dots \times e^{-r_i (t_i - t_{i-1})} \\ &= S \times (1 - S)^{i-1} \times M \times e^{-r t_i} \quad ; \text{for } r_1 = r_2 \dots = r_i \end{aligned}$$

We can see that $\frac{\partial Bi}{\partial i} < 0, \frac{\partial Bi}{\partial t} < 0$, i.e. the benefit declines with time and with rank order,

however as mentioned before the cost also declines with time and rank order, suggesting that an attacker may decide either way upon observing a number of existing exploits for a vulnerability. Note here, that we are not attempting to solve the equilibrium rank order of attackers, which arguably depends on their individual characteristics (that are unknown in our case as we only observe the attack but nothing about the attacker himself); instead, we are considering the response of an arbitrary attacker when presented with a situation. In section 4, we present our estimation approach based on survival analysis to determine that strategic interaction. The data is described next.

3. Data

The Symantec Corporation maintains a publicly available database that contains a list of anti-virus signatures, and write-ups describing a significant subset of those signatures. The write-ups specify which vulnerabilities a particular worm² is exploiting. In general, vulnerabilities are identified by their CVE-ID³; however, for the worms exploiting Microsoft products, the write-ups usually specify the vulnerabilities by the Microsoft Security Bulletin (MSB) number.

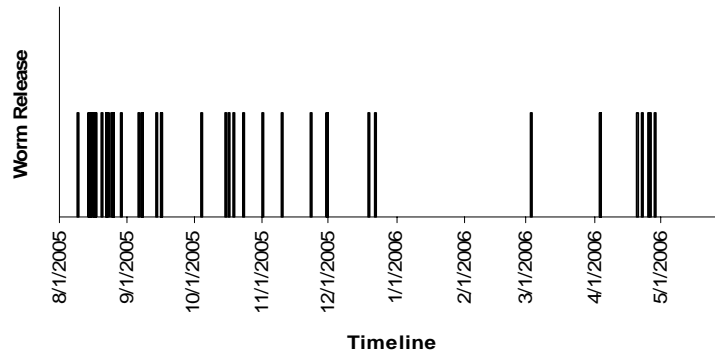


Figure 3: Release of worms exploiting vulnerabilities in MSB 05-039 (published on Aug 9, 2005).

Our dataset is comprised of information on release of computer worms that exploited vulnerabilities (MSB 04-016 to 05-055, i.e. all bulletins from May 1 2004 to Dec 31 2005) in Microsoft products over a two year period from May 2004 to May 2006 (the patches for the vulnerabilities are typically released in conjunction with the bulletin, so release of a timely patch was not an issue). In total there were 85 security bulletins, which were exploited by a total of 90

¹ By i^{th} attacker we do not necessarily mean i^{th} individual, it is the i^{th} attack, which may have been a repeated attack launched by an attacker whose previous attempts were not successful (success ratio S was too low). For example, Code Red II worm was launched by the same attacker who had launched Code Red I; after fixing the software bugs he found in Code Red I.

² The Symantec anti-virus signatures cover various kinds of malware – worms, viruses, torjans, downloaders, adwares – however, for our analysis we use the generic term, worm, to refer to any malcode exploiting a vulnerability.

³ See, <http://cve.mitre.org/>, for details.

worms in the two year period that we analyzed.⁴ Worm release cycle for bulletin MS05-039 (released on Aug 9, 2005) is shown in Figure 3. For each bulletin we identified a) whether the vulnerabilities can be exploited remotely or does it require local access (0= remote, 1= local), b) the criticality rating of the bulletin (0= low|moderate|important, 1= critical), c) whether the vulnerabilities affect the operating system or applications (0= app, 1= OS), d) the time of launch of each worm, and e) a censoring variable to indicate the launch (or not) of a worm.

Interestingly, as mentioned before, not all worms are written from scratch – attackers release new worms by modifying the source code of other worms that they encounter to suit their own needs. These new worms, which consist of the original worm code with additional or reconfigured features, are called worm variants. We use the Symantec’s nomenclature to identify which worms are original and which ones are their variants. Symantec names worms according to a 3-part <platform.name.variant> format, such that W32.Blaster.C is the variant C of Blaster worm which affects Win32 (32 bit Windows) family of operating systems.

4. Estimation Approach

We use the survival analysis to determine the *hazard* of launch of a worm given other explanatory variables. The hazard function is the probability of a spell ending at time t , given that it has lasted until at least t . The hazard rate $\lambda(t)$ can be written as a function of survivor function and the probability density.

$$\lambda(t) = \lim_{h \rightarrow \infty} \frac{Lt}{h} \Pr(t \leq T < t + h | T \geq t) = \lim_{h \rightarrow \infty} \frac{Lt}{h} \frac{\Pr(t \leq T < t + h)}{\Pr(T \geq t)} = \frac{f(t)}{S(t)}$$

In a proportional hazard model the hazard rate can be written as,

$$\lambda(t; x) = e^{x\beta} \lambda_0(t)$$

Where, $\lambda_0(t)$ is the baseline hazard rate. The actual hazard $\lambda(t)$ differs from baseline by a factor of $\exp(x\beta)$, x being the vector of explanatory variables. In our preliminary analysis we have assumed Weibull distribution for the time spell as it allows easy interpretation of estimation results. In case of Weibull, baseline hazard $\lambda_0(t)$ is equal to $\alpha t^{\alpha-1}$, where the shape parameter α determines whether the hazard rate is increasing or decreasing with time.

Table 1: Maximum Likelihood Estimation of Weibull Proportional Hazard Model

	Coefficient	Std. Error
OS	0.651	0.679
Local	-0.316	1.063
Critical	2.693***	1.046
Constant	-5.877***	1.239
Shape Parameter (α)	0.274***	0.075

- * indicates significant at 10% level, ** indicates significant at 5% level, and *** indicates significant at 1% level.

5. Preliminary Results

Our first specification, tests for the significance of various explanatory variables in explaining the launch of the first worm for every MSB (results in Table 1). Note that while OS and Local variables are insignificant, critical vulnerabilities have a hazard rate that is about 15 times ($= e^{2.7}$) larger than the baseline. This finding suggests that users should definitely patch such

⁴ There could be other worms written to exploit these vulnerabilities, but they were probably not significant enough to be included in the Symantec’s write-ups. In our analysis we focus only on worms which were found in the Symantec’s database.

vulnerabilities as soon as possible, while the vendors should try to avoid them in the first place. Since 2005, Microsoft has offered, as beta version, automatic patching for critical-rated vulnerabilities. This was indeed a good step in reducing the window of exposure for their users (it is now said that, from Windows Vista automatic, patching would become a standard feature). It is important to observe that while proportional hazard for critical vulnerabilities is very high, the shape parameter for Weibull is significantly less than 1 (declining hazard), which implies that attackers either attack soon after a vulnerability is discovered or they do not even bother (in our dataset only 12 out of 85 MSBs had any worm exploiting them during the time period considered). Earlier research by Arora et al (2004) has found that sometimes vendors take too long to release patches (sometimes they do not even release them). In light of current findings we are inclined to think that that it may not be as bad as initially supposed given that many vulnerabilities are never exploited.

Table 2: Maximum Likelihood Estimation of Competing Risks Model

	Single Risk Model		Original Worm		Worm Variant	
	Coefficient	Std Er	Coefficient	Std Er	Coefficient	Std Er
OS	0.548*	0.289	0.555	0.477	0.629	0.424
Local	-0.451	0.602	0.009	0.754	-13.64	631.1
Critical	2.391***	0.524	3.011***	1.028	1.672***	0.627
Worms_sofar	0.187***	0.030	0.055	0.056	0.260***	0.040
Worms_sofar_sq	-0.004***	0.001	-0.001	0.001	-0.005***	0.001
Constant	-5.736***	0.612	-6.530***	1.142	-6.026***	0.753
Shape Parameter (α)	0.480***	0.039	0.375***	0.058	0.482***	0.052

- * indicates significant at 10% level, ** indicates significant at 5% level, and *** indicates significant at 1% level.

Given that most attackers are focusing on few key vulnerabilities leaving others untouched, we next explore the significance of rank order of attacks on the hazard rate. We test three specifications, the results for which are presented in Table 2. The first specification looks at hazard of release of a worm from the time the previous worm was released (vulnerability discovery date in case of first worm). We include two new explanatory variables: worms_sofar and worms_sofar_sq (square of worms_sofar). The square term should capture any non-linearity in the influence of rank order. The following two specifications are similar to the one described only it considers the competing hazards i.e. release of an original worm vis-à-vis release of a worm variant.

We find that the rank order dependence is non-linear implying that later worm developers benefit from the release of first few worms by copying their technology, whereas, the benefit fades away as the proportion of vulnerable population declines to near zero (because of success of earlier worms and the patching of computers with time). Furthermore, analyzing the competing hazards model (last two columns in Table 2) we find that the baseline hazard of variants is higher than that of original worms (constant for variant, $-6.026 > -6.530$, constant for worm), confirming our hypothesis that attackers are more likely of launching a variant soon after initial worms are launched than to develop their own original worms.

Overall, our findings have important implications for information systems policy as well as for implementation of worm defense techniques for ensuring the safety of our network systems. In future work, we will be exploring alternative specifications to test the robustness of our estimates. In addition, we hope to identify additional factors that may explain inter-vulnerability and inter-worm variation.