A large-scale study of the time required to compromise a computer system

Hannes Holm

Abstract—A frequent assumption in the domain of cyber security is that cyber intrusions follow the properties of a Poisson process, i.e., that the number of intrusions are well modeled by a Poisson distribution and that the time between intrusions is exponentially distributed. This paper studies this property by analyzing all cyber intrusions that have been detected across more than 260,000 computer systems over a period of almost three years. The results show that the assumption of a Poisson process model might be unoptimal – the log-normal distribution is a significantly better fit in terms of modeling both the number of detected intrusions and the time between intrusions, and the Pareto distribution is a significantly better fit in terms of modeling the time to first intrusion. The paper also analyzes whether time to compromise increase for each successful intrusion of a computer system. The results regarding this property suggest that time to compromise decrease along the number of intrusions of a system.

Index Terms—Invasive software (viruses, worms, Trojan horses), Risk management, Network management

1 Introduction

Research on the topic of dependable computing is often based on assumptions regarding properties of failures in systems and knowledge concerning how system failures behave is as such critical. Choosing appropriate models for describing the number of failures in a system, and the time between failures, is of particular importance as employment of an inappropriate statistical model can result in improper research conclusions. In today's business environment where IT is a cornerstone of almost all business, choosing appropriate models is critical also to practitioners - an inaccurate reliability estimation as a result of having used an inappropriate probability distribution can put an enterprise out of business.

Consequently, various research efforts (e.g, [1], [2], [3]) have been made to gather evidence towards empirically identifying the distribution(s) best fit to model failure rates, providing credibility to distribution assumptions made by researchers and practitioners.

Cyber security is an area closely related to the domain of dependable computing [4] and also a topic which is becoming increasingly more important for enterprises to properly manage - successful intrusions can cause significant economical deficits as a result of, for example, unavailable software services, data loss and disclosed confidential information. In the same fashion as in the area of dependable computing, various research efforts have been made to enable estimating and predicting the security of a system (e.g., [5], [6], [7], [8], [9], [10]). Many such models origin from the domain of dependable computing [4], for example, how many intrusions that occur for a certain system (compared to faults). As with failure rates, to properly model the intrusion process there is a need to also model the time between intrusions, often referred to as Time To Compromise (TTC) – the time spent by one or more attackers to successfully compromise a system. Modeling of cyber intrusions and TTC is, as modeling of failure rates in dependable computing, frequently applied by both researchers and practitioners. Thus, utilizing appropriate theoretical distribution(s) is of immense importance.

Unfortunately, there are next to no empirical research efforts on the topic of analyzing the distribution of cyber intrusions or TTC. The most likely reason behind this is that appropriate data are considered too sensitive to share with the public domain. So far, there is to the author's knowledge only a single publication [11] empirically analyzing the distribution of TTC, and this study involved a single distributed system, students doing the penetration testing and a mere 59 samples. As a consequence, there is dire need to further explore properties of cyber intrusions.

This paper brings important insight into this aspect through analysis of 5,602,097 malware alarms corresponding to 203,025 intrusions that have occurred across 261,757 computer systems of a large international enterprise between October 2009 and August 2012.

The first question concerns cyber intrusions as a function of computer systems. In dependable computing, system failures are typically modeled as Poisson processes. This requires failures as a function of systems to follow a Poisson distribution, i.e., to have single parameter determining shape and position, and assuming that each fault is independent of others. This assumption has been criticized by empirical observations on various occasions [1], [2], [3], but is still applied by both researchers and practitioners (e.g., [12], [13], [14]) - much due to its mathematical simplicity. In the domain of cyber security, system compromises are often assumed to follow a Poisson distribution (e.g., [5], [6], [7], [8]). However, this assumption has never been thoroughly tested.

RQ1: *What statistical distribution is best fit to model the number of intrusions of a computer system?*

• *H. Holm is with the department of Industrial Information and Control Systems, The Royal Institute of Technology, 100 44 Stockholm, Sweden. E-mail: hannes.holm@ics.kth.se.*

The assumption of a Poisson process model for system failures requires the time between failures to be exponentially distributed. Consequently, a common assumption within dependable computing is that the time between failures is an exponentially distributed random variable. As for the assumption of the Poisson distribution, this premise has been criticized with respect to its validity (cf. Section 2), but it is still frequently applied by both researchers and practitioners. This assumption has been carried over to the domain of cyber security, where it is frequently used to describe TTC (e.g., [9], [8]). However, in this domain there is not enough empirical evidence to conclude which statistical distribution that is best suited to model TTC.

RQ2: *What statistical distribution is best fit to model the time to compromise a computer system?*

The third and final research questions concerns the common philosophy that we learn from our mistakes [15]. Interpreted in the area of cyber security: an individual that has more experience of cyber intrusions should be more aware of such threats, thus decreasing the likelihood of additional intrusions from occurring. For instance, (s)he should possess more security aware behavior when managing software updates, choosing passwords, configuring read/write access to shared folders, clicking links in emails, browsing the web and utilizing USB drives. Given such a philosophy, the time required to compromise a computer system should increase for each successful intrusion.

RQ3: *Does time to compromise increase for each successful intrusion?*

This final research question is possible to formalize as an hypothesis, *H1*, which is described below.

H1: *Time to compromise is positively correlated to the number of intrusions of a computer system*

The rest of the paper unfolds as follows. Section 2 presents related work. Section 3 describes the tested statistical distributions and the method for comparing goodness of fit. Section 4 presents the studied data, and the method for gathering it. Section 5 presents the results and Section 6 critically examines these findings. Finally, Section 7 concludes the paper.

2 Related work

The related works can be classified in three categories: 1) previous work that empirically analyze distributions of cyber intrusions, 2) previous work that empirically analyze properties of system failures and 3) previous work that examines how security awareness affects vulnerability. This chapter is divided according to these three categories.

2.1 Distribution of cyber intrusions

There are various research efforts on modeling cyber intrusions, for example, to measure the security of a system (or system-of-systems) [5], [6], [7], [8], [9], [10]. These studies make various assumptions to support their claims; a frequent one being that attacks, or intrusions, follow a Poisson process, i.e., that the number of attacks or intrusions is well modeled by a Poisson distribution and that the time between such events is exponentially distributed.

There are however few empirical studies which analyze the validity of these models and their assumptions. One of the few available studies was conducted by Holm et al. [16] who analyzed the validity of 18 different cyber security estimation models of 6 different types through correlating their estimations against the TTC values of 34 intrusions gathered from a cyber defense exercise. The low correlation scores observed during this study indicated that the tested models might need revision. The authors did however not study the actual distribution behind the cyber intrusions.

To the best of the author's knowledge, there is only a single publication that explores the distribution of TTC using empirical data. That is, Jonsson and Olovsson [11] modeled the security intrusion process from the perspective of targeted cyber attacks. The authors had 12 groups of students (with two students in each group) try to breach a distributed UNIX system (24 workstations and one file-server) as part of an academic course. The authors observed a total of 59 breaches, each of which they categorized in one of six classes depending on the amount of hours required to perform it (0-1 hours, 1-2.5 hours, 2.5-5 hours, 5-7.5 hours, 7.5-10 hours, and 10- hours). The observed frequencies within these categories were then compared to the expected number of frequencies (according to a chi-square test), given that one would believe that TTC is an exponentially distributed random variable. The results from this test suggested that TTC, given a reasonably skilled attacker that tries to breach a system with a minimum amount of effort, indeed is exponentially distributed. While the results from this study certainly are interesting, the low number of samples in combination with the very specific scenario could be a threat to their validity.

2.2 Distribution of system failures

It is clear that there are crucial differences between reliability and security. In particular, the latter involves an actor with the intent of compromising an asset. However, there are also similarities: Many security estimation models have been developed based on models used for reliability estimates [4]. For instance, attack trees [17] are related to fault trees [18] and state-based techniques such as Markovian modeling, which have been extensively developed and used in classical dependability contexts, are now frequently employed for cyber security estimations (e.g., for modeling known system vulnerabilities using privilege graphs [19]). Given the lack of relevant theory in the cyber security domain, reliability studies pose a reasonable starting point for harvesting potential statistical distributions for analysis. The remainder of this section describes significant

empirical studies conducted on system failure properties to extract statistical distributions that should be tested.

Schroeder and Gibson [1] studied various characteristics of 23,000 failures that had occurred during 9 years and on 20 different systems (most being cluster solutions). Relevant to the scope of the present study, they modeled the number of failures per node and per system according to the normal distribution, the Poisson distribution and the log-normal distribution. This analysis showed that the assumption of Poisson failure rates could be suspect - the Poisson distribution showed rather poor fit compared to the normal and log-normal distributions. The authors also studied the time between failures for single nodes of system, and the aggregated time between failures for all nodes of systems. For this analysis, the authors compared model fit for the Weibull, log-normal, gamma and exponential distribution. The authors found that the Weibull and gamma distributions provided the best fit for both node time between failures and system-wide time between failures.

Nurmi et al. [2] studied the model fit for the exponential, Pareto, Weibull and hyperexponential distribution for three datasets describing time between failures (two gathered from systems running at academic institutions and the data from [20]). The authors found that the Weibull and hyperexponential distributions provided the greatest degree of fit. Of these two, the hyperexponential provided the best fit, but is in turn also more complex due to its additional parameters.

Heath et al. [3] studied time between failures as the time between reboots of commodity workstations. A total of 126 workstations and 3343 observed reboots were observed. The authors tested the model fit of the exponential, Weibull, Pareto and Rayleigh distributions and found that of these models, the Weibull was the best at describing time between reboots.

2.3 Does time to compromise increase for each successful intrusion?

There has to the author's knowledge not been any studies involving real-world intrusion data on the topic of how the security of a system change after successful compromise(s). There have however been various empirical studies that are related to this topic. While there are important aspects not covered by these studies, e.g., whether a more security aware individual apply software security updates more frequently than a less security aware individual, their results should hint towards the general value of security awareness.

Sheng et al. [21] developed an online computer game aimed to teach users about how to avoid phishing attempts. The authors studied the effectiveness of the software through testing the ability of eight participants at identifying phishing web sites from a set of real and phishing web sites. They observed that the amount of correctly identified phishing attempts increased from 69% to 83% after that the participants had played the game.

Downs et al. [22] conducted a survey of 232 computer users to examine determinants phishing success. The results from this study showed that those who correctly answered a knowledge question about the definition of phishing were significantly less likely to fall for phishing attacks.

Shaw et al. [23] tested eight hypotheses related to cyber security awareness on a dataset consisting of 154 university students. Their results showed that users who are more aware of security risks are better at projecting potential security risks.

Stanton et al. [24] found that presence of security awareness training correlated with positive password behavior of computer users through a survey involving 1167 participants.

Workman [25] combined phishing experiments with surveys (a total of 612 samples) to examine if an individual's perception of vulnerability and severity of social engineering correlates to whether or not that individual succumbs to social engineering attacks. The results from the study showed that individuals who perceived social engineering to be of higher severity or vulnerability were less susceptible to such attacks.

3 Analyzed Statistical Distributions

This chapter is divided in two sections - Section 3.1 describes the statistical distributions that are tested and Section 3.2 how their goodness of fit is measured. The first part of Section 3.1 covers distributions tested for RQ1 and the second distributions tested for RQ2.

3.1 Studied distributions

3.1.1 Number of intrusions of a computer system

Based on the previous work presented in Section 2, there are three distributions that can be considered in terms of modeling the number of intrusions of a computer system – the Poisson (PO), normal (N) and log-normal (LN). These were all tested by [1] to model failure rate as a function of systems. While [1] did not concern cyber intrusions, it is the most closely related study available, and thus also the best possible starting point.

3.1.2 Time to compromise a computer system

Five statistical distributions can be considered based on the related work – the exponential (EXP), log-normal (LN), Weibull (WBL), gamma (GAM) and the two-parameter generalized Pareto (PAR) – in terms of degree of fit to TTC of computer systems. EXP is commonly applied to model time between failures and has previously been shown to be a good fit for TTC [11]. LN, WBL, GAM and PAR are typically tested in studies regarding time between failures [1], [2], [3], [26], [27], [28].

3.2 Measuring degree of fit

There are various metrics that can be applied to measure how well a statistical distribution model observed data. For example, Kolmogorov–Smirnov, Cramér-von Mises, Anderson-Darling, Shapiro-Wilk and Chi Square. This research utilizes the Akaike Information Criterion (AIC), a standard technique for ranking alternative models, to compare the relative goodness of fit for the distributions. AIC has many advantages compared to other goodness-of-fit metrics; significant advantages are described in Section 3.2.1.

An issue with AIC (and other goodness of fit metrics) is however that it is unable to portray the relative goodness of fit at different locations of a distribution. For instance, the

fit might be superb for the majority of data, but terrible at modeling the tails. To study such aspects, this paper employs *quantile-quantile (QQ) plots* [29]. The straight line in a QQ-plot denotes what is expected by the tested statistical model. Consequently, a well fit model should have most datapoints following this line.

3.2.1 The Akaike Information Criterion

The AIC was introduced to extend the method of maximum likelihood estimation to the situation of multimodel choice [30]. As pointed out in [31], Akaike's criterion links Boltzmann's entropy, Kullback-Leibler information and maximum likelihood – thus tying together information theory with statistics. Essentially, the AIC is an estimator of the expected relative Kullback-Leibler information [31].

Conceptually, the AIC can be seen as adding a penalty to models with many parameters, thus rewarding not only fit but also simplicity [32]. The AIC is a measure of the badness of fit of a model defined with parameters estimated by the maximum likelihood method [30]. It is defined as:

$$AIC = -2 \cdot \log \text{likelihood} + 2 \cdot \text{number of parameters} \quad (1)$$

A central concept in AIC is the maximum likelihood estimation. Myung [33] describes it as (where PDF = Probability Density Function): *"Given the observed data and a model of interest, find the one PDF, among all the probability densities that the model prescribes, that is most likely to have produced the data*".

It might be objected that the log likelihood by itself is enough to compare models. However, [34] stress the importance of the bias correction term employed by AIC.

The lower its AIC, the better a distribution fits the data. To evaluate distributions against each other, the difference $\Delta_i = AIC_i - AIC_{\min}$ is formed, where AIC_i is the value of the model being evaluated and $AIC_{\min}$ is the value of the model with the lowest AIC. By definition $AIC_{\min}$ thus has $\Delta_i = 0$. Some rough rules of thumb given by [31] are given in Table 1 (there are other models with similar rules, e.g., [35], [32]).

TABLE 1: Empirical support for models using AIC differences; rules of thumb reprinted from [31].

Δ_i	**Level of empirical support of model** i
0-2	Substantial
4-7	Considerably less
> 10	Essentially none

4 DATA

The studied enterprise is an enterprise that specializes in IT and has businesses in various locations all over the world. This research concerns all computer systems employed in the enterprise, and any malware that have been detected on these between the 15th of October 2009 and the 10th of August, 2012.

4.1 Information about computer systems

A registry keeps information about systems that are (and have been) employed at the enterprise. The registry also keeps timestamps of when systems have been installed *or* removed. That is, there is either a timestamp for when a system was added (given that it is currently operational), or when it has been removed (given that the system is no longer employed). This data allowed extracting all systems that were presumably operational at some point during the studied period of time. In total, 261,757 computer systems were employed during the next to three studied years. However, this figure should be viewed with some care as the registry is not automatically updated - some systems which are employed by the enterprise are as such not detailed in it and timestamps are not always accurate or available.

The registry also details various aspects regarding the systems. For example, the individual that a system corresponds to, it's location, it's hardware specifications (e.g., HP EliteBook 690p) and the employed operating system type and version (e.g., Microsoft Windows Vista Enterprise Service Pack 2).

In total, the enterprise had 697 different operating systems and versions, and 727 different hardware setups during the studied period of time. The three most common operating systems were Windows Vista (53.7% of all systems), Windows 7 (4.4%) and Windows 2000 Workstation (4.2%). A total of 16,642 (or 6.4%) systems had a UNIX operating system (e.g., Linux or Solaris) and 245,115 (or 93.6%) a Windows operating system.

As the operating systems and hardware models were detailed, it allowed viewing which systems that were workstations and which that were servers. Out of the 261,758 studied systems, 23,144 (or 9.8%) were servers (e.g., Windows Server 2008) and 238,614 (or 91.2%) workstations (e.g., Windows 7).

4.2 Information about malware alarms

Employees of the enterprise are required to utilize computer systems that are provided by it. Each such computer system is equipped with the anti-malware solution provided by Symantec (Symantec Endpoint Protection (SEP) [36]). When a malware is detected on a system by its local agent, the information about the intrusion is submitted to a central database. Consequently, this database logs information about most *known* cyber intrusions at any location within the enterprise (some intrusions are missed by SEP, cf. Section 6.3). Each intrusion event contains information about, for instance, 1) the unique identification string of the compromised system, 2) the IP of the compromised system, 3) the time of the event and 4) the detected malware.

The enterprise has stored data on intrusions detected by SEP daily (and without disruption) since the 15th of October 2009 - this research concerns all data from this occasion until the 10th of August, 2012. A total of 5,602,097 malware alarms associated with 63,658 unique computer systems were observed during the studied time period. Thus, 24.3% of all systems operational during the study had at least one malware incident. While this figure should be viewed with care, it should also serve as a *reasonably accurate* estimate.

The fraction of compromised servers and workstations, and their corresponding operating systems types (UNIX or Windows) could be estimated in the same manner. In total, 6 out of 23,144 servers (or 0.026%), 63,652 out of 238,614 workstations (or 26.7%), 3 out of 16,642 systems with UNIX (or 0.018%), and 63,655 out of 245,115 systems with Windows (or 26.0%), had at least a single malware incident. Thus, it can be concluded that servers and UNIX systems have significantly less malware incidents than workstations and Windows systems.

4.3 Extracting malware incidents from malware alarms

The 5.6 million alarms are however not to be confused with the actual number of malware occurring at the enterprise as a single malware typically raise several alarms by the Symantec anti-malware solution. This due to 1) the action taken by the anti-malware (e.g., first an event describing putting the malware into quarantine and then an event describing removal of the malware) and 2) that malware typically spreads to various locations of systems (each detected instance providing an event in the anti-malware log).

This means that alarms typically appear in clusters (or alone), with significant pauses between each set. Consequently, a reasonable method of extracting malware incidents from the alarms is to devise a threshold based on the amount of time that has passed after an alarm. Given a sufficient amount of time, the anti-malware (or incident handling personnel) should have been able to properly mitigate the threat (e.g., through removing the malware or formatting the system). However, this time threshold should not be set too high - given enough time, there is a likelihood of the same malware occurring again, through a new intrusion.

After statistical analysis and discussions with IT security personnel at the enterprise (who are employed to monitor and manage malware activity on a daily basis), a threshold of 1 calendar week was chosen. This threshold was chosen due to three reasons: 1) The scanning policy of SEP is set to scan once per week. Consequently, if SEP's runtime detection module fails to detect the execution of a malware, then this malware should (given that applicable signatures are available) be detected by SEP's scanning module within 1 week. 2) It sometimes (but not often) takes several days for SEP (and in some cases, IT personnel) to find and properly remove a malware. 3) It denotes a significant duration of time between alarms in the dataset - few unique malware incidents should thus be removed from the dataset and few malware incidents should be counted multiple times through this method. Or in other words, our data suggest that it is very unlikely that multiple malware incidents occur for the same system within the same week. The potential methodological issues with this method are discussed in Section 6.3. Also, it is only used to enable a subset of all results (cf. Section 4.6.1).

An example of how the extraction of malware incidents was performed can be seen in Figure 1, which depicts all malware alarms that occurred during the studied time period for a single computer system. A total of four unique malware (as denoted by SEP) were detected - `W32.Imaut.AA` (27 alarms), `W32.Imaut` (27 alarms), `W32.Pilleuz!gen1` (54 alarms) and `W32.SillyFDC` (92 alarms). Given the threshold of 1 week between malware alarms, there would be six unique malware incidents counted for the system - 2 incidents of `W32.Imaut.AA` and `W32.Imaut`, and a single incident of `W32.Pilleuz!gen1` and `W32.SillyFDC`. The curious regular lapses in time between April and June correspond to weekends.

Through this methodology, the original set of 5,602,097 alarms was narrowed down to 279,850 *unique* malware incidents. In other words, each malware incident spawned on average 20 alarms.

An overview of the detected malware and their properties can be seen in Table 2. Symantec categorizes malware according to its impact in terms of damage (*Damage*), effort required for removal (*Removal*) and propagation speed (*Distribution*). Damage and Distribution have the states *High, Medium* and *Low*, Removal has the states *Difficult, Moderate, Easy*. As each malware event contained the alias of the detected malware, the Symantec online encyclopedia [37] could be consulted to gather data regarding these properties. Some detected malware did however not have information available in the encyclopedia about these attributes - consequently, malware characteristics regarding such events could not be categorized. In total, approximately 73% of all malware events could be detailed regarding these aspects.

As can be seen in Table 2, a large portion of the classified malware incidents were of medium damage (71.32%), easy to remove (86.55%) and had a low propagation speed (90.41%).

TABLE 2: Characteristics of malware incidents according to the Symantec encyclopedia.

Attribute	State	Frequency	% of total
Damage	High	879	0.43%
	Medium	144719	71.32%
	Low	57325	28.25%
	No data	76927	-
Removal	Difficult	287	0.14%
	Moderate	27007	13.31%
	Easy	175629	86.55%
	No data	76927	-
Distribution	High	11201	5.52%
	Medium	8256	4.07%
	Low	183466	90.41%
	No data	76927	-

4.4 Extracting intrusions from malware alarms

Due to the complexity of today's malware market it is difficult for miscreants to effectively manage all aspects involved with spreading their applications (e.g., discovering vulnerabilities, creating exploits, constructing payloads, reaching out to target users and remaining undetected). Thus, a frequently used method for miscreants who wants to install applications on target hosts is to consult service providers who offer means for them to do so. The miscreants simply pay the service provider for the number of application installs that it wishes to receive and the service provider installs the malware through some type of distribution mechanism [38].

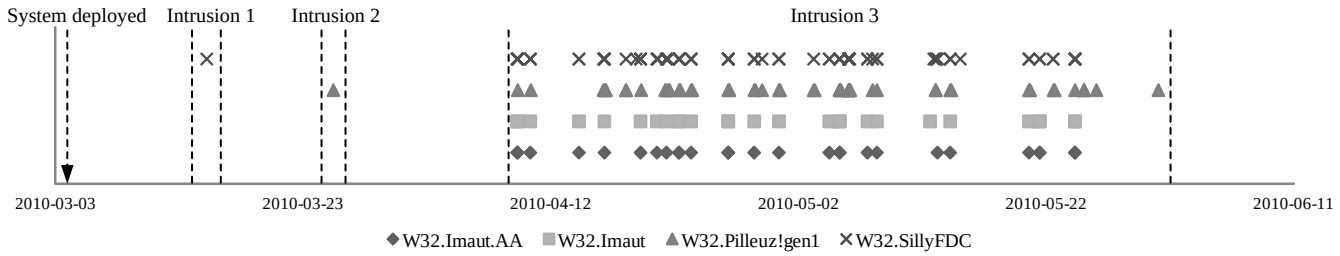


Fig. 1: All malware alarms for a system.

As the service provider is rewarded for the amount of installs, it is beneficial to install multiple malware simultaneously. In practice this means that a system sometimes has multiple unique malware detected within a short amount of time, all originating from the same intrusion, i.e., an event which has led to system compromise (e.g., usage of a compromised USB-drive, a drive-by-download, or execution of a malware appended to an e-mail).

Intrusions are measured by only counting the first malware alarm in each set of alarms that are detected less than a week apart. I.e., when there is one week or more without any additional alarms for the same system it means that an intrusion has been managed. This method is the same as what is employed to extract malware incidents from alarms (cf. Section 4.3), but without considering which malware that is in question. It is also employed due to the same reasoning as is discussed previously. The example system illustrated in Figure 1 receives three intrusions through the method (the first and second intrusion involving single malware incidents and the third intrusion resulting in four malware incidents).

Through this methodology, the 5.6 million alarms were narrowed down to 203,025 intrusions. I.e., 38% ($279,850 - 203,025 = 76,825$) of all malware incidents occurred simultaneously to other incidents, and could thus be the result of pay-per-install schemes [38].

4.5 Measuring the number of intrusions of a computer system

The first research question of the study concerns the number of intrusions of a computer system. To enable reliably answering this question, there is a need to 1) know how long all studied systems have been employed and 2) normalize the results to some appropriate time scale. To satisfy the first property, this research focuses on the set of systems (out of 63,658) which have reliable timestamps on their installation or removal and had been operational for at least a year during the studied time period. A total of 33,575 systems (with a total of 119,877 intrusions) fulfill this property. To satisfy the second property, the mean number of intrusions per system and year is measured and rounded off to the nearest integer.

4.6 Measuring time to compromise

The second research question concerns TTC. To answer this question, this study employs three measurement methods - the time between compromises, the time to first compromise and the overall time to compromise.

4.6.1 Time To First Compromise

The time to first compromise (TTFC) is measured as the time from the deployment of a computer system (denoted in the system registry) until the first malware alarm for that system. It can be mathematically denotes as follows: Pose that a system S is installed at a time T_I at the enterprise. Now pose that the first malware alarm for this system is registered at a time T_M. The $TTFC$ for S would thus be $TTFC_S = T_M - T_I$.

This metric circumvents any reliability issues that might the data extracted using the previously described 1 week threshold (cf. Section 6.3). However, the nature of the metric also means that not all systems can be studied - only computers that were installed after the 15th of October 2009 and operational at the time the register was extracted are suited for analysis (systems only have one timestamp corresponding to either removal or installation, cf. Section 4.2). A total of 5728 systems, with a mean operational time of 434 days, satisfy this property. None of these systems were servers or were running UNIX. Thus, the analysis of TTFC is regarding Windows workstations.

4.6.2 Time Between Compromises

An intrusion is assumed to have been managed when there has been 1 week or more without any additional malware alarms for a system. The time between compromises (TBC) is consequently measured as the time from the last malware alarm for an intrusion until the time of the first malware alarm for the next intrusion. The pseudocode in Algorithm 1 describes how TBC values are extracted from the 5.6 million malware alarms.

```
Data: Array of malware alarms, sorted according to system IDs and
      time-stamps
Result: Array of TBC values for different systems
Intrusions = [];
while Alarms are left to process do
    CurrentSystem = CurrentAlarm[0];
    CurrentTime = CurrentAlarm[1];
    if CurrentSystem == PreviousSystem then
        TBC = CurrentTime - PreviousTime;
        if TBC > 24*7 hours then
            Intrusions << [PreviousSystem, TBC];
        end
    end
    PreviousSystem = CurrentSystem;
    PreviousTime = CurrentTime;
end
```

Algorithm 1: Calculation of Time Between Compromises.

Relating back to Figure 1, there would be two samples for TBC (the time between intrusion 1 and 2, and 2 and 3). In

total, there are 139,363 samples available for analysis of TBC.

None of the UNIX systems had more than a single intrusion. Furthermore, of the 6 compromised servers, only 3 had more than a single intrusion, and in total 10 TBC samples. Thus, analysis with AIC is conducted on Windows workstations (a total of 139,353 TBC samples). The 10 server TBC samples ranged from 9.3 days to 153.4 days, with an arithmetic mean of 64.6 days.

4.6.3 Overall Time To Compromise

The overall time to compromise (TTC) is the TTFC and the TBC in combination for all systems that have such data. In other words, the TTFC and TBC samples for the 5728 systems mentioned in Section 4.6.1. In addition to the TTFC samples, these systems had a total of 7441 TBC samples. Thus, a total of 13,169 samples are available for analysis of overall TTC.

4.7 Measuring time to compromise over intrusions

The third research question of this study concerns whether time to compromise increases over the number of intrusions of a computer system. To analyze this property, there is a need to know both TTFC and TBC. Consequently, this dataset is the same as what is employed to measure overall TTC (13,169 samples) - but with an additional variable to track the order of intrusions of a computer system and enable statistical correlation analysis of how TTC behave as a function of intrusions.

5 Results

This chapter describes the results of the study and is divided in three sections (along the three research questions of the study).

5.1 Intrusions per system

AIC scores for the tested distributions (LN, N, and PO), given the 33,575 studied systems, are given in Table 3. The log-normal distribution is best fit, without any empirical support for the Poisson distribution or the normal distribution. Figure 2 shows QQ-plots for the tested statistical distributions. These support the conclusion that the log-normal is best fit for modeling cyber intrusions per year.

However, as is shown in the cumulative distribution functions (CDFs) in Figure 3, approximately 90% of all studied systems have 3 intrusions or less every year. While the QQ-plots show that the log-normal outperforms the normal and Poisson also in this range, they also suggest that both the normal and Poisson might be reasonably fit at modeling a large portion of the dataset. Thus, in cases where low prediction accuracy of systems with more than 3 intrusions per year is considered adequate, all three distributions might suffice.

Another interesting property visualized by the QQ-plots is whether a distribution over- or underestimates the actual number of intrusions per year. Overestimating the number of intrusions, i.e., providing conservative estimates, is typically recommended for security practices as it allows some room for error. As can be seen, the log-normal distribution provides conservative estimates, while the normal and Poisson give more liberal estimates.

TABLE 3: AIC scores for intrusions as a function of systems. The numbers within brackets denote the empirical support for different distributions of intrusions per year. $\Delta_i < 2$ gives substantial support, $4 < \Delta_i < 7$ considerably less, and $\Delta_i > 10$ essentially none [31].

Dataset	Samples	PO (Δ_{PO})	N (Δ_N)	LN (Δ_{LN})
Intrusions	33575	92896 (16404)	103845 (27353)	76491 (0)

Maximum likelihood parameter estimates for all tested distributions regarding intrusions as a function of computer systems can be found in Table 4. There is a mean of two intrusions per year.

TABLE 4: Maximum likelihood parameter estimates of intrusions as a function of systems.

Dataset	λ (PO)	μ (N)	σ (N)	μ (LN)	σ (LN)
Intrusions	0.496	2.015	1.608	0.492	0.597

5.2 Time to compromise

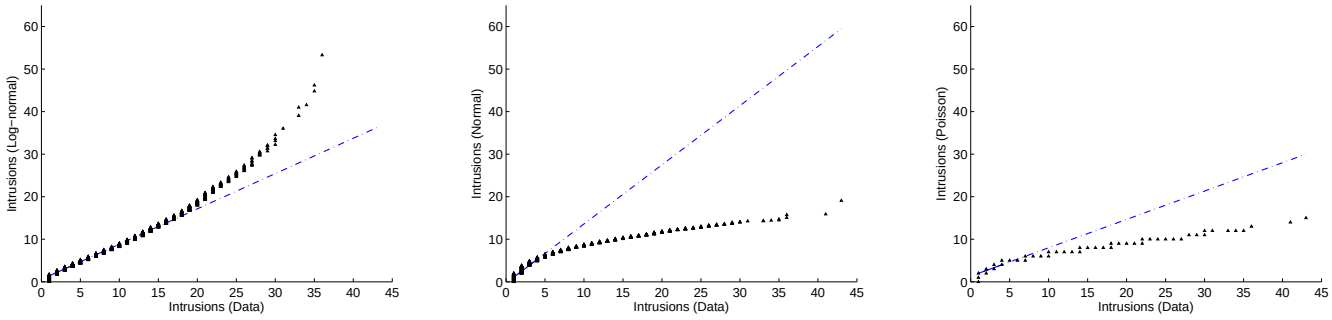
5.2.1 Time To First Compromise

AIC scores for the tested distributions regarding TTFC (EXP, GAM, LN, PAR and WBL) are given in Table 5. The Pareto distribution is best fit, with Weibull receiving minor support, and Gamma, log-normal and exponential receiving no support. The QQ-plots in Figure 4 support the conclusion that Pareto is best fit (with log-normal being the clearly worst fit). Figure 6a shows CDFs for TTFC along with the tested statistical distributions. As can be seen, approximately 90% of all intrusions requires 400 days or less - a range which all distributions except the log-normal are reasonably fit at modeling (Figure 4f shows the fitness of the log-normal in this range). It can however be argued that intrusions of systems that are difficult to compromise are the most interesting to accurately model; the reason for their relative security could be, for instance, storage of sensitive information. Given this viewpoint, only the Pareto might suffice.

Finally, Pareto gives (on overall) conservative estimates on the amount of days until system compromise; the remaining distributions give more liberal estimates.

5.2.2 Time Between Compromises and overall Time To Compromise

AIC scores for the tested distributions regarding TBC and overall TTC are given in Table 5. The log-normal distribution is best fit for both datasets, with essentially no empirical support for any of the other tested distributions. Figure 5 shows QQ-plots for TBC. As the QQ-plots for TTC are very similar to those of TBC (with the same conclusions), they are not detailed in the paper. On overall the QQ-plots support the conclusion that log-normal is best fit. However, the relative fit of the distributions vary significantly across the number of days required to compromise a system.



(a) Log-normal (scale 45/60). (b) Normal (scale 45/60). (c) Poisson (scale 45/60).

Fig. 2: QQ-plots for intrusions per year.

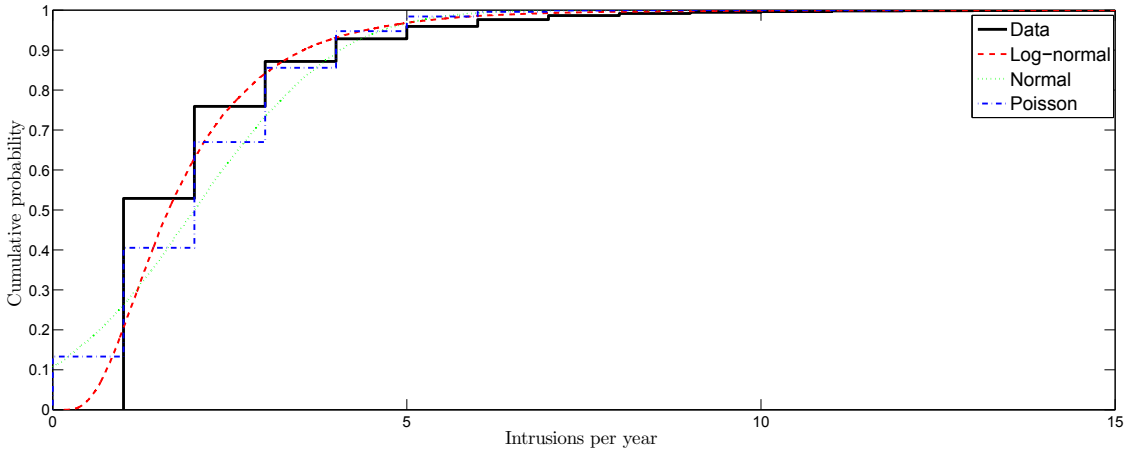


Fig. 3: Cumulative distribution functions of intrusions with fitted statistical distributions.

The exponential, gamma, and Weibull distribution are reasonable at modeling intrusions which require less than 100 days; however, they are poor at modeling intrusions that require more time to be successful. The log-normal and Pareto model the dataset well and rather well, respectively, up to approximately 600 days, but terrible at any higher values. Figure 6b and 6c shows CDFs for the datasets along with the tested statistical distributions. As can be seen, approximately 90% of all intrusions for TBC require less than 180 days. The goodness of fit of the exponential distribution in this range is illustrated in Figure 5f. As a large portion of the data (where days > 80) is not well modeled, the exponential distribution might be a poor choice for modeling TBC and overall TTC.

All tested distributions provide (on overall) conservative estimates for intrusions requiring up to approximately 600 days - from this point on the log-normal and Pareto turn to provide very liberal estimates.

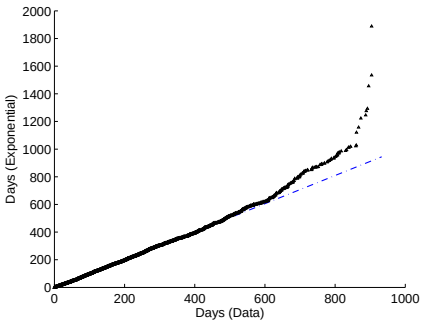
The small irregularity around 60 days in Figure 6b and 6c was studied in depth to identify the reason(s) behind this phenomenon; however, there were no apparent patterns that could be deduced. For instance, the 60-day intrusions occurred at seemingly random points in time and there were no dominant malware or locations. Furthermore, the enterprise itself had no explanation. Thus, to our best knowledge this irregularity is due to random variation.

5.2.3 *Maximum likelihood parameter estimates*

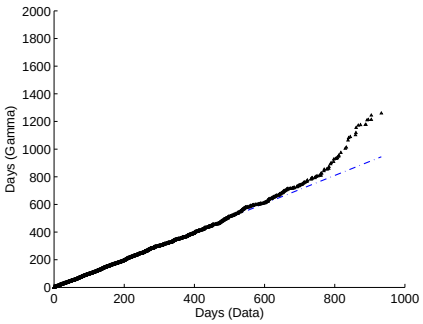
Maximum likelihood parameter estimates for all tested distributions regarding intrusions as a function of time can be found in Table 6. The (arithmetic) mean TTFC is 171 days, the mean TBC is 75 days, and the mean overall TTC is 108 days. These estimates can be of use to researchers and practitioners who have yet to gather such data.

5.3 Time to compromise as a function of intrusions

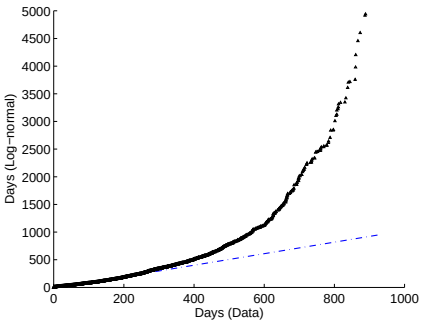
The third research question analyzes the hypothesis "*Time to compromise is positively correlated to the number of intrusions of a computer system*" (*H1*). Spearman correlation analysis [29] is employed to analyze this property. The Spearman correlation is a non-parametric test that only requires that the individual observations can be ranked into two ordered series [29] - a property fulfilled by the dataset. The significance of the results is estimated using chi-square hypothesis testing. The null hypothesis of the hypothesis test, *H0*, is that the correlation coefficient between the tested random variables is zero. The boundary associated with rejecting the null hypothesis is described using probability, p. A commonly used level of significance is $\alpha = 0.05$, so that if $p < 0.05$, then it



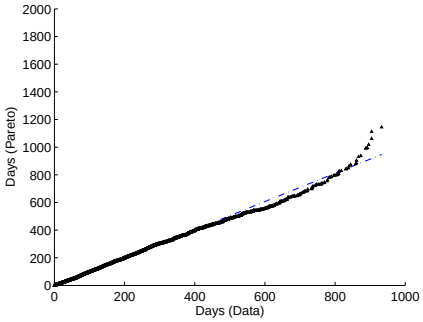
(a) Exponential (scale 1000/2000).



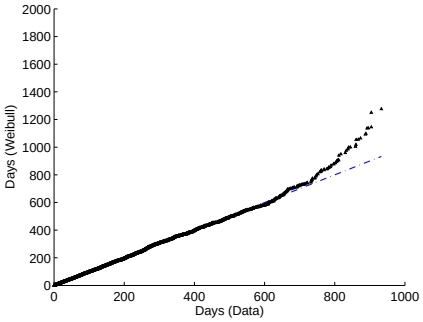
(b) Gamma (scale 1000/2000).



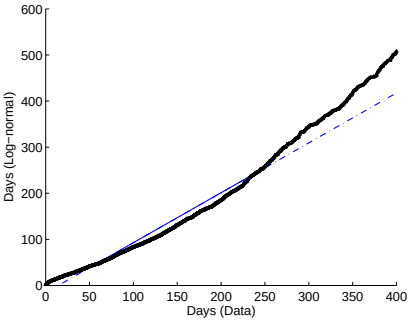
(c) Log-normal (scale 1000/5000).



(d) Pareto (scale 1000/2000).

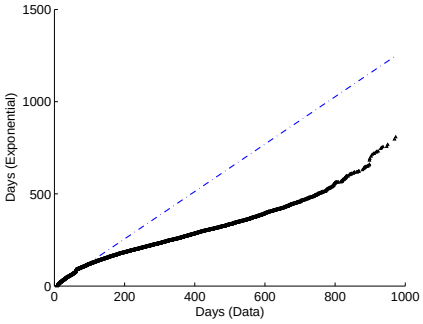


(e) Weibull (scale 1000/2000).

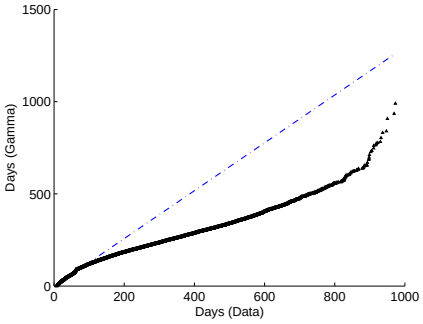


(f) Log-normal, 90% data (scale 400/600).

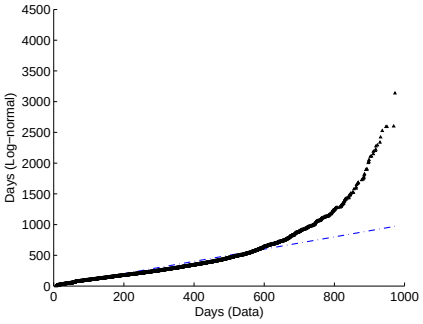
Fig. 4: QQ-plots for Time To First Compromise (measured in days).



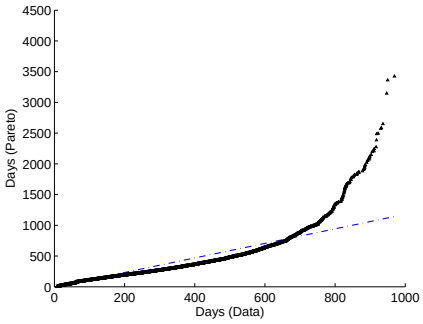
(a) Exponential (scale 1000/1500).



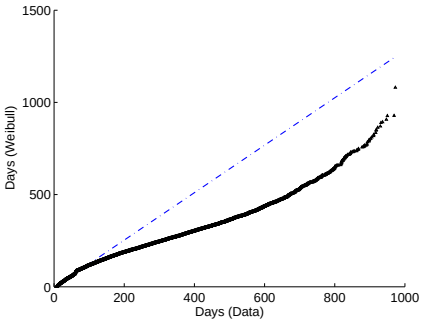
(b) Gamma (scale 1000/1500).



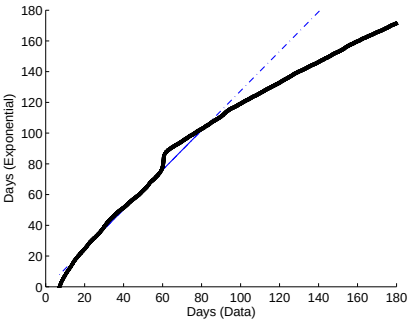
(c) Log-normal (scale 1000/4500).



(d) Pareto (scale 1000/4500).



(e) Weibull (scale 1000/1500).



(f) Exponential, 90% data (scale 180/180).

Fig. 5: QQ-plots for Time Between Compromises (measured in days).

TABLE 5: AIC scores for distributions of time to compromise. The numbers within brackets denote the empirical support for different distributions of intrusions per year. $\Delta_i < 2$ gives substantial support, $4 < \Delta_i < 7$ considerably less, and $\Delta_i > 10$ essentially none [31].

Dataset	Samples	EXP (Δ_{EXP})	LN (Δ_{LN})	WBL (Δ_{WBL})	GAM (Δ_{GAM})	PAR (Δ_{PAR})	Best fit
TTFC	5728	70390 (39)	71675 (1323)	70362 (10)	70369 (18)	70352 (0)	PAR
TBC	139353	1481946 (32474)	1449471 (0)	1479671 (30199)	1481853 (32382)	1470374 (20903)	LN
TTC	13169	149855 (1176)	148678 (0)	149428 (750)	149648 (969)	149107 (428)	LN

TABLE 6: Maximum likelihood parameter estimates of time to compromise.

Dataset	λ (EXP)	μ (LN)	σ (LN)	a (WBL)	c (WBL)	p (GAM)	a (GAM)	α (PAR)	x_m (PAR)
TTFC	0.0058	4.616	1.247	175.357	1.060	1.084	158.222	-0.0905	187.018
TBC	0.0133	3.719	1.065	71.272	0.914	0.968	77.439	0.267	55.031
TTC	0.0092	4.005	1.247	101.224	0.877	0.859	126.611	0.275	80.267

TABLE 7: Correlation scores for H1.

Dataset	Samples	Correlation	p
TTC	13169	-0.458	< 0.0001
TBC	7441	-0.131	< 0.0001

implies that the null hypothesis shall be rejected [29]. In other words, there is less than 5% likelihood of error if trusting in the stated hypothesis. On the other hand, if $p \geq 0.05$, then the stated hypothesis shall be rejected.

An overview of the arithmetic mean TTC values for the studied intrusions can be seen in Figure 7. In Figure 7, the horizontal axis concerns the number of compromises of the studied computer systems; the vertical axis concerns the mean TTFC, or the mean TBC. For instance, the mean time between the first and second compromise of a system (TBC1) is 67 days and the time between the fourth and fifth compromise of a system (TBC4) is 55 days. Very few systems had 9 or more intrusions during the three studied years. Thus, the TTC values for such intrusions are put in the same dataset to allow a proper sample size (TBC9-TBC28).

On overall, it seems that TTC decreases along the number of intrusions of a system. However, the TTFC is on overall significantly larger than any TBC - something which could have an effect on the correlation scores. I.e., if including all data then the correlation score could indicate that TTC decreases over the number of intrusions, even though this only is the case when comparing TTFC to TBC. Thus, the statistical analysis in this section studies H1 from a perspective of 1) all TTC samples and 2) only TBC samples.

An overview of the results can be seen in Table 7. The correlation scores for both TTC and TBC are significant and negative (the opposite of what was expected), suggesting that the time required to compromise a computer system actually drops along with the number of intrusions of the system.

6 Discussion

This chapter is divided in three sections. Section 6.1 discusses the results from the goodness of fit tests (RQ1 and RQ2) and Section 6.2 discusses the results regarding RQ3. Finally, Section 6.3 and Section 6.4 critically examines the reliability and validity of the results.

6.1 A hypothesized intrusion model

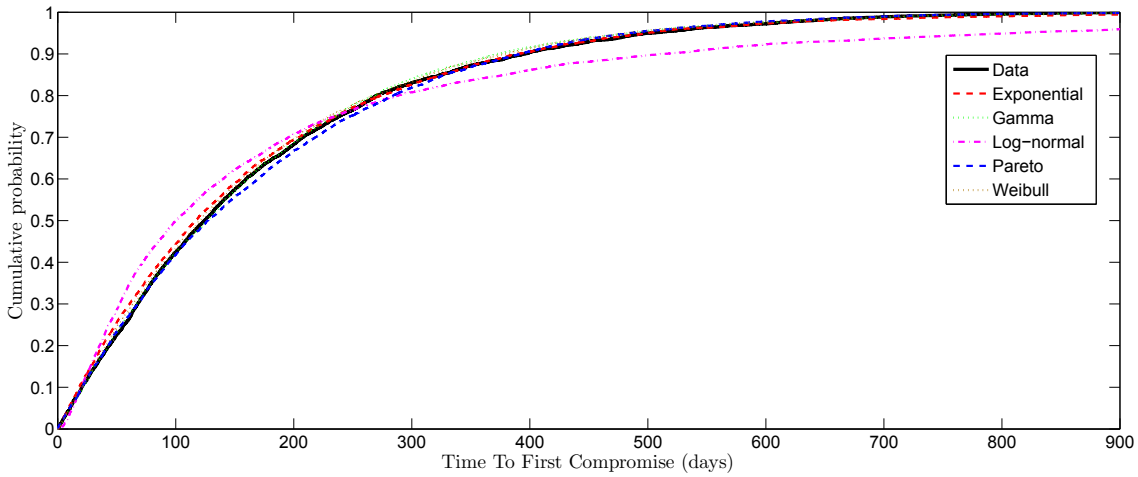
The Pareto distribution is often recognized based on the Pareto principle, commonly referred to as the ”80-20 rule”. I.e., 20% of all samples carry 80% of the total weight. During this study, the Pareto distribution was found to best fit the time to the first compromise of a computer system. That is, a small set of the studied computer systems had a significantly higher time to first intrusion than the rest. From the perspective of the Pareto principle - if one would compromise a set of systems, then 20% of these systems would cover 80% of the total effort. Notable is that the actual distribution in the dataset was closer to ”50-20” - 20% of the systems covered 50% of the effort.

To further examine the validity of the Pareto distribution, this research studies the goodness of fit for the five tested distributions on the dataset presented by Holm et al. [16] (a cyber defense exercise involving targeted attacks by professional penetration testers). The AIC scores for this dataset are -86.4 (EXP), -86.0 (GAM), -64.8 (LN), -88.6 (PAR) and -87.7 (WBL). In other words, the study by [16] supports the results from the present study - the time to first intrusion of a computer system is best modeled by a Pareto distribution.

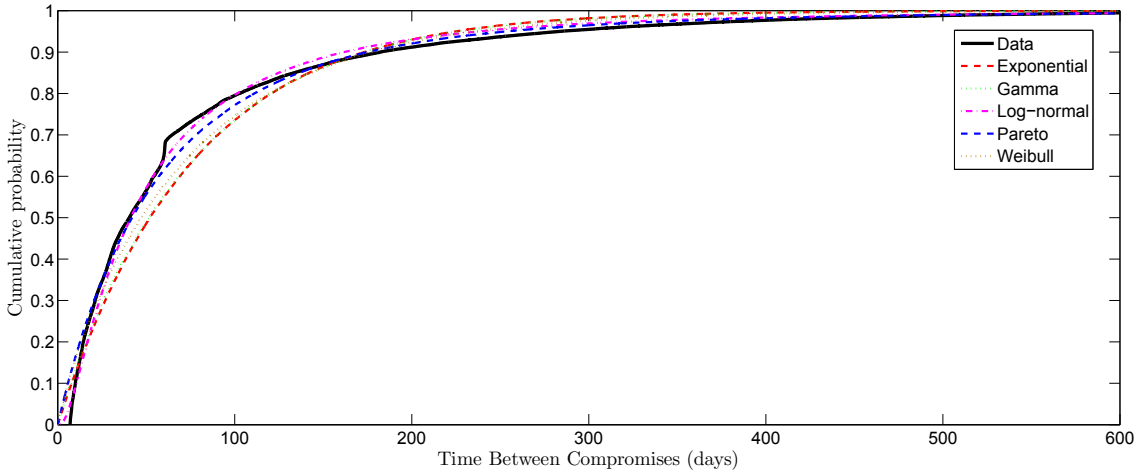
Once a system has been successfully compromised, an entire new situation unfolds in terms of effort for recapturing it. As it has previously been compromised, it is known to the criminal underworld. It might also contain backdoor functionality from the previous intrusion that was not detected by the anti-malware. Furthermore, next to all studied malware (a total of 1920 different malware types were detected during the study) have the functionality to spread through USB drives. If the anti-malware fails to detect (or clean) malware present on a USB drive, then this will provide a new intrusion opportunity each time the drive is inserted into the system.

Consequently, it is not very surprising that the time between intrusions follow a different statistical distribution than the time to the first intrusion of a system - i.e., the log-normal rather than the Pareto.

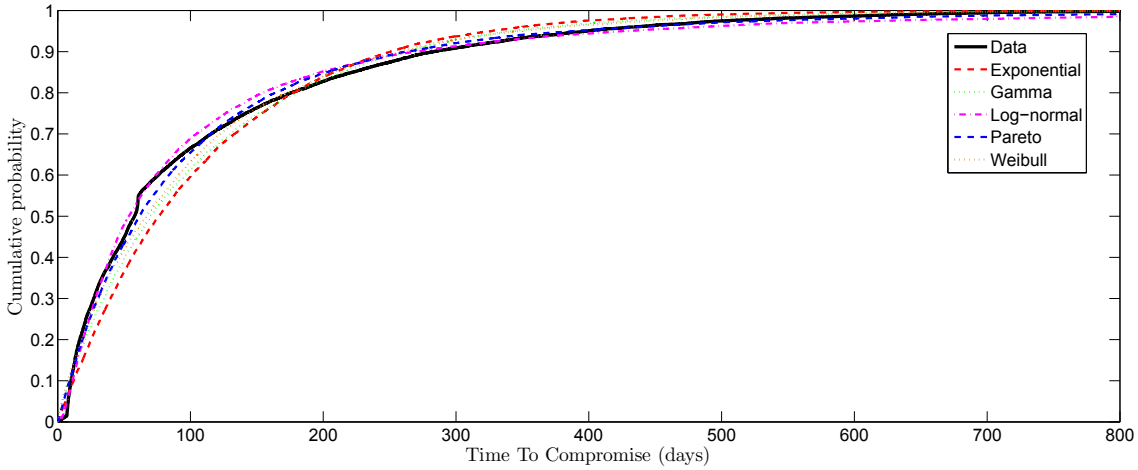
The probability density function of the log-normal distribution is initially rapidly increasing until it reaches its peak, and then slowly decreasing until it reaches zero. In other words, the probability of successful intrusion significantly increases during the first brief period of time, and then slowly decreases during a significantly longer period of time.



(a) Time To First Compromise.



(b) Time Between Compromises.



(c) Overall Time To Compromise.

Fig. 6: Cumulative distribution functions of TTFC, TBC and TTC with fitted statistical distributions.

The rationale behind this is that systems (or the employees' managing them) that previously have been compromised, and still are vulnerable to cyber attacks, probably are quickly reclaimed. However, if the time increases and no intrusion has

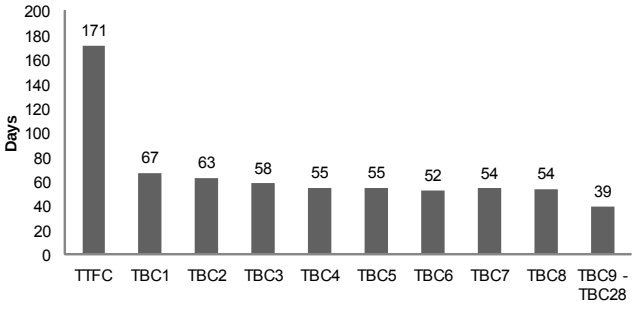


Fig. 7: Mean time to compromise values over intrusions.

occurred, then this would imply that the system is relatively secure (the previous threat has been completely managed), and it is to be expected that significant effort will be required to reclaim the system.

Also, there is reason to believe that this model not only is applicable for large-scale attacks, but also for targeted attacks: a vulnerable system will likely require little effort for a targeted attacker to compromise - but if the system is not quickly compromised, then it can be expected to have few (or none) easy-to-exploit vulnerabilities, and thus require significant time to compromise.

A key concept for the log-normal distribution is the central limit theorem, which denotes that the probability distribution of a variable is the product of many random variables (none of which dominates the result) [39]. Consequently, the observations from this study hints towards the variety of factors influencing the time between cyber intrusions in fact could be independent of one another. This is certainly a valuable insight to both researchers and practitioners.

The log-normal was also best fit to model the overall TTC (the combined TTFC and TBC estimates). This is simply the outcome of each studied system having several intrusions and thus more TBC than TTFC samples - valuable knowledge to a practitioner who wishes to calculate the overall number of intrusions of a computer system, but without considering both the Pareto and log-normal distribution.

Finally, the results from this study points to the Poisson process model for modeling cyber intrusions as a significantly worse fit than the Pareto and log-normal distributions. Future research should reflect upon these results when deciding upon distribution alternatives.

6.2 Time to compromise decrease with the number of intrusions

The third research question concerns whether TTC increases with the number of intrusions of a computer system. In other words, does a compromised system become less vulnerable when having been subjected to an intrusion? The results from this study hint towards the opposite - a system becomes more vulnerable along with the number of intrusions of that system. This study can hypothesize four possible reasons behind this result:

1) *Once a system is broken, it can no longer be trusted.*

One common viewpoint within cyber security is that a system once compromised cannot be trusted. That is, if a malware has been executed on a system then the software employed on that system can no longer be trusted - including intrusion detection mechanisms such as anti-malware solutions. During the present study it is unknown when computers were formatted as a result of an intrusion. Cyber security personnel at the enterprise however indicated that this seldom is the case. Consequently, each additional intrusion (assuming that the system is not formatted) should decrease its dependability.

2) *The threat management process fails to address significant sources of threats.*

Another reason could be poor threat management. Oftentimes, problematic malware are managed by external personnel who use remote login to remove threats. Consequently, threats that persist on mediums outside of the system might not be properly cleaned (e.g., malware on USB drives).

3) *Security awareness is only temporarily increased as a result of a cyber intrusion.*

A common hypothesis is that individuals who fall victim to crime becomes more security aware. However, the results from this study points towards the opposite. One reason behind this could be that the increased awareness is not permanent. In fact, previous research has shown that the effect might be rather temporary [40].

4) *Computer users seldom apply software security updates.*

A fourth reason could be that many computer users fail to implement critical security updates of the web browser and its resources [41], [42], perhaps due to being unaware of that malware often exploit such flaws. A missed update will not decrease the vulnerability of the corresponding system and given the assumption that patches rarely are deployed, this suggests that a system will become increasingly more vulnerable during its life-cycle.

6.3 Reliability

There are several claims that can be made against the reliability of results presented in this paper. This section describes these claims.

Anti-malware solutions have false positives and false negatives. A significant issue for this study is that anti-malware solutions fail to detect a large portion of all intrusions that occur and report intrusions that actually are non-malicious events. However, this is the case for *any* study on the topic of intrusions as all intrusion detection systems struggle with false positives and false negatives.

Anti-malware alarms correspond to detection, not infection. It is not certain that the time of detection equals the time of infection. However, as there is no reason to believe that this error is more prevalent for one intrusion than another, its effects should be randomly distributed over the dataset and thus not affect the analysis results. Furthermore, the time of infection will per definition always be a problem for these types of studies - no matter what forensics tool that is employed it will still be a matter of detection (compared to infection).

Seven days are used as a threshold for denoting when one intrusion has been managed. To enable extrating malware incidents and intrusions from the 5,602,097 malware alarms, this research utilized a time-based threshold of seven days. I.e., if seven or more days have passed without any additional alarms, then the intrusion is treated as mitigated. While it effectively enabled cleaning the data, there are other possible explanations for pauses in time than mitigation. For example, a computer system might be shut down by an employee that goes on vacation in the middle of an anti-malware scan that provides alarms, and then started a month later when the employee gets back, giving more alarms for the same incident. If this was a serious reliability threat, then the mean time between intrusions should be close to 7 days. However, it is almost ten times larger than this amount. Thus, it is (most likely) a small reliability threat and above all a *useful* threshold - without such a threshold it would not be possible to answer RQ1 or measure TBC.

What if the anti-malware solution was set up in different ways on different systems? It would indeed have been a significant issue if each anti-malware agent was set up in a different way (e.g., regarding scanning policy and update frequency). Fortunately, the configuration of each instance of the utilized malware solution is locked to an enterprise-wide standard profile. Consequently, all agents should follow the same policies.

Calendar time and not working time is measured. Time to compromise models such as [9], [10] assume that TTC is the *working time* (i.e., active work) by an attacker to compromise an asset. This is also how TTC was empirically modeled by Jonsson and Olovsson [11] - as the group working time by students (cf. Section 2). Thus, the results from the present research need be viewed in the light of calendar time and not be confused with working time; the best-fit distribution of calendar time might not be the same as for working time. That said, it is impossible to conduct studies using real world data on this topic without consulting calendar time - enterprises naturally cannot observe the actual time spent by attackers.

6.4 Validity

There are several claims that can be made against the validity of results presented in this paper. This section describes these claims.

The characteristics of the attackers are unknown. As for calendar time, the skill and mindset of the attackers are unknown to this study. Research utilizing controlled environments such as [11], [16] have the advantage that it can capture (and control) properties related to the attacker. However, in turn this could have adverse effect on the behavior of the attacker and the realism of the laboratory environment. Clearly, no issues regarding realism exist for the present study.

Most observed intrusions are likely the result of automated malware platforms. Most malware that occur are not the result of attacks targeted against specific individuals or enterprises, or specially crafted to penetrate a certain solution. Rather, they are a result of automated malware platforms such as drive-by-downloads [43], phishing bots [44] and computer worms [45]. While the author believe that the Pareto and log-normal distributions might be appropriate also for targeted attacks (cf. Section 6.1), there is little empirical evidence to justify this claim. Thus, this hypothesis should be interpreted with care until it has been properly studied - especially as the only related empirical study on the topic found TTC to follow the exponential distribution [11].

The results might only be valid for the characteristics of the studied enterprise. A registry contained information about all types of hardware and operating systems (and their versions) employed in the enterprise (cf. Section 4.1). Hundreds of different hardware models and operating systems located in more than 130 countries are present in the dataset. Consequently, the studied enterprise has a vast variety of computer systems and personnel, and the results should be interpreted thereafter.

7 Conclusions and Future Work

This paper studied properties of cyber intrusions through three research questions: *"What statistical distribution is best fit to model the number of intrusions of a computer system?"* (RQ1), *"What statistical distribution is best fit to model the time-to-compromise a computer system?"* (RQ2) and *"Does time to compromise increase for each successful intrusion?"* (RQ3).

Regarding RQ1, the results show that the log-normal distribution is best suited for modeling the number of intrusions of a computer system. The results are slightly different for RQ2, where the Pareto distribution is best fit for modeling the time to first intrusion and the log-normal distribution for modeling the time between intrusions.

Regarding RQ3, the results show that TTC actually decreases as a function of intrusions. A few key factors behind this result could be that 1) current intrusion detection mechanisms are not adequate, 2) the root cause of an intrusion sometimes is not mitigated, 3) the increase of security awareness is merely temporary and 4) computer users lack awareness on the risk of browser exploits. Future research would benefit from further studying these variables.

Important results from this research are also the parameter estimates and the number of compromised systems during the studied period of time, not only on overall, but for workstations, servers, UNIX and Windows systems. These estimates can be seen as a starting point for an enterprise that has yet to gather such data.

Finally, while this research concerns a large body of real-world data, there are many questions that are left unanswered. In particular, this research is not able to answer *why* some computers have tens of intrusions, while others have none. This is certainly an area which requires a great deal of more research, preferably utilizing real-world data such as what is analyzed in the present study. A valuable insight from the present research into this topic is that the log-normal distribution showed such a good degree of fit. This implies that the variables influencing successful compromise of a computer system in fact are independent - a property that should make valid research on the topic a great deal simpler.

REFERENCES

[1] B. Schroeder and G. Gibson, "A large-scale study of failures in high-performance computing systems," *Dependable and Secure Computing, IEEE Transactions on*, vol. 7, no. 4, pp. 337–351, 2010.

[2] D. Nurmi, J. Brevik, and R. Wolski, "Modeling machine availability in enterprise and wide-area distributed computing environments," in *Euro-Par 2005 Parallel Processing*, ser. Lecture Notes in Computer Science, J. Cunha and P. Medeiros, Eds. Springer Berlin / Heidelberg, 2005, vol. 3648, pp. 612–612, 10.1007/11549468_50. [Online]. Available: http://dx.doi.org/10.1007/11549468_50

[3] T. Heath, R. Martin, and T. Nguyen, "Improving cluster availability using workstation validation," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1, pp. 217–227, 2002.

[4] D. Nicol, W. Sanders, and K. Trivedi, "Model-based evaluation: From dependability to security," *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 48–65, 2004.

[5] J. Conrad, "Analyzing the risks of information security investments with monte-carlo simulations," in *Fourth Workshop on the Economics of Information Security*, 2005, pp. 2–3.

[6] B. Madan, K. Goševa-Popstojanova, K. Vaidyanathan, and K. Trivedi, "A method for modeling and quantifying the security attributes of intrusion tolerant systems," *Performance Evaluation*, vol. 56, no. 1, pp. 167–186, 2004.

[7] J. Ryan, T. Mazzuchi, D. Ryan, J. Lopez de la Cruz, and R. Cooke, "Quantifying information security risks using expert judgment elicitation," *Computers & Operations Research*, vol. 39, no. 4, pp. 774–784, 2012.

[8] N. Schneidewind, "Cyber security prediction models," *The R & M Engineering Journal, American Society for Quality*, 2005.

[9] M. McQueen, W. Boyer, M. Flynn, and G. Beitel, "Time-to-compromise model for cyber risk reduction estimation," *Quality of Protection*, pp. 49–64, 2006.

[10] D. Leversage and E. James, "Estimating a system's mean time-to-compromise," *Security & Privacy, IEEE*, vol. 6, no. 1, pp. 52–60, 2008.

[11] E. Jonsson and T. Olovsson, "A quantitative model of the security intrusion process based on attacker behavior," *Software Engineering, IEEE Transactions on*, vol. 23, no. 4, pp. 235–245, 1997.

[12] H. Okamura, T. Dohi, and S. Osaki, "Software reliability growth model with normal distribution and its parameter estimation," in *Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), 2011 International Conference on*. IEEE, 2011, pp. 411–416.

[13] P. Kapur, H. Pham, S. Anand, and K. Yadav, "A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation," *Reliability, IEEE Transactions on*, no. 99, pp. 1–1, 2011.

[14] J. Zheng, "Predicting software reliability with neural network ensembles," *Expert systems with applications*, vol. 36, no. 2, pp. 2116–2122, 2009.

[15] C. Harteis, J. Bauer, and H. Gruber, "The culture of learning from mistakes: How employees handle mistakes in everyday work," *International Journal of Educational Research*, vol. 47, no. 4, pp. 223–231, 2008.

[16] H. Holm, M. Ekstedt, and D. Andersson, "Empirical analysis of system-level vulnerability metrics through actual attacks," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, no. 6, pp. 825 –837, nov.-dec. 2012.

[17] B. Schneier, *Secrets and lies: digital security in a networked world*. Wiley, 2011.

[18] B. Fischhoff, P. Slovic, and S. Lichtenstein, "Fault trees: Sensitivity of estimated failure probabilities to problem representation." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 4, no. 2, p. 330, 1978.

[19] R. Ortalo, Y. Deswarte, and M. Kaâniche, "Experimenting with quantitative evaluation tools for monitoring operational security," *Software Engineering, IEEE Transactions on*, vol. 25, no. 5, pp. 633–650, 1999.

[20] D. Long, A. Muir, and R. Golding, "A longitudinal survey of internet host reliability," in *Reliable Distributed Systems, 1995. Proceedings., 14th Symposium on*. IEEE, 1995, pp. 2–9.

[21] S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L. Cranor, J. Hong, and E. Nunge, "Anti-phishing phil: The design and evaluation of a game that teaches people not to fall for phish," in *Proceedings of the 3rd symposium on Usable privacy and security*. ACM, 2007, pp. 88–99.

[22] J. Downs, M. Holbrook, and L. Cranor, "Behavioral response to phishing risk," 2007.

[23] R. Shaw, C. Chen, A. Harris, and H. Huang, "The impact of information richness on information security awareness training effectiveness," *Computers & Education*, vol. 52, no. 1, pp. 92–100, 2009.

[24] J. Stanton, K. Stam, P. Mastrangelo, and J. Jolton, "Analysis of end user security behaviors," *Computers & Security*, vol. 24, no. 2, pp. 124–133, 2005.

[25] M. Workman, "A test of interventions for security threats from social engineering," *Information Management & Computer Security*, vol. 16, no. 5, pp. 463–483, 2008.

[26] K. Cai, D. Hu, C. Bai, H. Hu, and T. Jing, "Does software reliability growth behavior follow a non-homogeneous poisson process," *Information and Software Technology*, vol. 50, no. 12, pp. 1232–1247, 2008.

[27] M. Vineyard, K. Amoako-Gyampah, and J. Meredith, "Failure rate distributions for flexible manufacturing systems: An empirical study," *European journal of operational research*, vol. 116, no. 1, pp. 139–155, 1999.

[28] J. Plank and W. Elwasif, "Experimental assessment of workstation failures and their impact on checkpointing systems," in *Fault-Tolerant Computing, 1998. Digest of Papers. Twenty-Eighth Annual International Symposium on*. IEEE, 1998, pp. 48–57.

[29] R. Warner, *Applied statistics: From bivariate through multivariate techniques*. Sage Publications, Incorporated, 2007.

[30] H. Akaike, "Factor analysis and AIC," *Psychometrika*, vol. 52, 1987.

[31] K. Burnham and D. Anderson, *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Verlag, 2002.

[32] S. Gokhale and R. Mullen, "A multiplicative model of software defect repair times," *Empirical Software Engineering*, vol. 15, pp. 296–319, 2010, 10.1007/s10664-009-9115-y. [Online]. Available: http://dx.doi.org/10.1007/s10664-009-9115-y

[33] I. Myung, "Tutorial on maximum likelihood estimation," *Journal of Mathematical Psychology*, vol. 47, no. 1, pp. 90–100, 2003.

[34] S. Konishi and G. Kitagawa, *Information criteria and statistical modeling*. Springer Verlag, 2008.

[35] Y. Sakamoto, M. Ishiguro, and G. Kitagawa, *Akaike information criterion statistics*. KTK Scientific Publishers, 1986.

[36] Symantec, "Symantec Enterprise Protection," Available on http://www.symantec.com/protection-suite-enterprise-edition, accessed September 10, 2012.

[37] ——, "Symantec Online Encyclopedia," Available on http://www.symantec.com/security_response, accessed September 14, 2012.

[38] J. Caballero, C. Grier, C. Kreibich, and V. Paxson, "Measuring pay-per-install: The commoditization of malware distribution," in *Proceedings of USENIX Security*, 2011.

[39] R. Mullen, "The lognormal distribution of software failure rates: origin and evidence," in *Software Reliability Engineering, 1998. Proceedings. The Ninth International Symposium on*. IEEE, 1998, pp. 124–133.

[40] R. Dodge and A. Ferguson, "Using phishing for user email security awareness," *Security and Privacy in Dynamic Environments*, pp. 454–459, 2006.

[41] D. Barroso, "Botnets-the silent threat," *European Network and Information Security Agency (ENISA)*, vol. 15, p. 171, 2007.

[42] S. Stamm, Z. Ramzan, and M. Jakobsson, "Drive-by pharming," *Information and Communications Security*, pp. 495–506, 2007.

[43] N. Provos, P. Mavrommatis, M. Rajab, and F. Monrose, "All your iframes point to us," in *Proceedings of the 17th conference on Security symposium*. USENIX Association, 2008, pp. 1–15.

[44] J. Milletary and C. Center, "Technical trends in phishing attacks," *Retrieved December*, vol. 1, p. 2007, 2005.

[45] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, "A taxonomy of computer worms," in *Proceedings of the 2003 ACM workshop on Rapid malcode*. ACM, 2003, pp. 11–18.

Hannes Holm is a PhD student at the department of Industrial Information and Control Systems at the Royal Institute of Technology (KTH) in Stockholm, Sweden. He received his MSc degree in management engineering at Luleå University of Technology during 2010. His research interests include enterprise security architecture and cyber security regarding critical infrastructure control systems.