On Stochastic Risk Ordering of Network Services for Proactive Security Management

Mohamed Amezziane
Department of Mathematical Sciences
Depaul University
Chicago, Illinois
mamezzia@depaul.edu

Ehab Al-Shaer and Muhammad Qasim Ali
Dept of Software and Information Systems
University of North Carolina Charlotte
Charlotte, North Carolina
{ealshaer, mali12}@uncc.edu

Abstract—Contemporary network services don't have any statistical ranking mechanism for proactive security management. Since the emerging threats are actively exploiting the vulnerabilities in network services to compromise the system, not much attention has been paid to rank these services based on their vulnerability history. We argue in this paper that a reliable mechanism could be used to rank these services based on their vulnerability history. Such ranking will be significantly helpful for proactive network security management to partition services and deploy security countermeasures. We propose a framework using stochastic order alternatives to statistically rank network services based on time intervals between exploits as reported by National Vulnerability Database (NVD). We show that Statistical techniques can be used to rank these services by modeling the related metrics. We validated our technique using products of known ranking, and presented some case studies to confirm our result on real network services.

I. INTRODUCTION

Since the organizations and individuals rely heavily on the internet connectivity for their daily needs, network services and/or softwares have become an important part of the infrastructure. Change in the network behavior and vulnerability exploitation is an immense threat to such an infrastructure [1]. Such readily-available network connectivity is inherently vulnerable to network attacks, thus security and/or stability of the network services has become an important consideration. Many of these services act as servers and they are remotely exploitable due to inherent vulnerabilities. Vulnerabilities can be successfully exploited with three elements: System susceptibility or flaw, attacker’s access to the flaw and attacker’s capability to exploit the flaw. Increased sophistication of current attacks shows the capability of attackers. Due to the inherent nature of the readily available networks, accessing the network is a feature available to users. Thus the main element which can be taken care by network security designers and engineers is system susceptibility.

 Attacks on network services running over large infrastructure have grown in number and sophistication over the years. Risk ranking of these services provides effective means for improving security decision support and optimizing network security design. In this paper we present analytical foundation and novel techniques for statistical ranking of network services based on vulnerability inter-arrival time reported by the National Vulnerability Databases (NVD) [2]. NVD is the US government official repository for standard based vulnerability management data. They keep track of each vulnerability, score it using standard metric (CVSS) [3] and publish it. NVD announces each vulnerability with ID, standard name of the software/system, weakness type, discovery/published date of the vulnerability and some other reference and description data. We used the weakness type and published date fields in our research. We only analyzed the vulnerabilities that are remotely exploitable. We also used inter-arrival times of vulnerabilities for our analysis.

We proposed two methods for statistical ranking. The first method is based on the hypothesis testing with ordered alternatives where we intend to test whether a given ordering of several products is valid or not, this is based on the observed characteristics of the vulnerabilities (severity score and inter-arrival time). The second method is based on estimated intensity function, evaluated at the time of occurrence of the last vulnerability. To avoid working under wrong assumptions, we work with distribution-free (or nonparametric) methods, which makes minimal to no assumptions about the underlying distributions of the observations. Nonparametric method for detecting order is the Jonckheere-Terpstra test (JT) [4]. This method does not require knowledge about the underlying distribution, nor does it call for equality of variance. Moreover, the distributions of the observations do not have to be continuous and hence it can be used to compare both discrete and continuous random variables.

Stochastic ordering and ranking has been widely used in many different disciplines. It is used by engineers to analyze the products and machines reliability, by biostatisticians in the pharmaceutical industry to compare treatments, by the medical industry to study patients survival [5]. To our knowledge, this work is the first to conduct such analysis using NVD data repository. In our previous work,[6] and [7], we used the vulnerability history for predicting the next potential vulnerability score and time using probabilistic predictive modeling. However, in this work our goal is to rank or order services using only inter-arrival and not the NVD scores. Moreover
statistical ranking has been done in other domains [8], [9] but no attention has been given so far to rank network services.

The rest of the paper is organized as follows. Section 2 discusses statistical models that fit vulnerable services; however Section 3 describes the ranking methodology used in this work. Moreover Section 4 shows the simulation of proposed methodology along-with the case studies from NVD. Lastly Section 5 concludes the work done.

II. Modeling Vulnerable Services as Repairable Systems

Services and products are repaired after each vulnerability. Such systems are called repairable by reliability engineers [10]. However non-repairable system should be totally replaced after a failure [10]. It is obvious that software/service falls into the repairable system category. Suppose $T_0$, $T_1$, ... are the times of failure of the system, where $T_0$ is zero and $X_1$, $X_2$, ... are the times between failures, i.e., $X_i$ is the time between the $i-1$th and $i$th failure for $i=1, 2, 3, ...$. In our study, the vulnerability occurrences are failure of the system. It is obvious due to the nature of the phenomenon, that $T_i$ and $X_i$ are positively valued continuous random variables. For a fixed $t$, we denote by $N(t)$, which is the count random variable that represents the number of failures in the time interval $(0, t)$.

If $X_i$’s tends to increase with time, the system becomes more reliable over time. In contrast to it, the system becomes more error prone if it decreases. One can alternatively study the properties of the stochastic process $N(t)$. In fact, $N(t)$ is related to the $T_i$ through the equation: $\Pr(N(t) = k) = \Pr(T_k < t)$, which states that the number of failures before time $t$ is $k$ is equivalent to the $k$th failure happens before time $t$. Hence, one can study the system reliability by fitting an appropriate model to $N(t)$. This is usually done by measuring the failure rate over time through the rate of occurrence of failure (ROCOF), also known as the process intensity function. It is defined as $v(t) = d/dt\{E[N(t)]\}$, where $E[Y]$ denotes the expected value of $Y$. Thus a good system has a decreasing ROCOF over time [11], [12].

Since occurrence of a failure is generally considered a rare event, Poisson processes ought to be a plausible choice to model the number of failures over time, assuming that the failures are independent. ROCOF on the other hand, cannot be assumed to be constant over time, therefore non-homogeneous Poisson processes (NHPP) are best suited. Its relevant statistical methodology is well developed and easy to apply as it can be seen from the probability function of a NHPP: $\Pr(N(t) = k) = \{M(t)/k!\}exp(-M(t))$, where $M(t) = \int_0^t v(u) du$. It can be observed that when $v(t) = v$ is constant over time the NHPP becomes a homogeneous Poisson process, thus the $X_i$ are independent and identically exponentially distributed with rate parameter $v$. If we consider NHPP with time dependent ROCOF $v(t)$, then the number of failures in the time interval $(t_1, t_2)$ has a Poisson distribution with mean $M(t_2) - M(t_1)$. If we can find a suitable parametric form for $v(t)$, we can get very flexible model for the failures of a repairable system. The most frequently used models for $v(t)$ are:

- The Exponential Law Process (ELP): $v_1(t) = \exp(\beta_0 + \beta_1 t)$, as the value of $\beta_1$ increases, the number of failures increases as well.
- The Power or Duane’s Law Process (PLP): $v_2(t) = \gamma^{1/t - 1}$. For a fixed $\gamma$, as the value of the shape parameter $\delta$ increases, the system becomes more exposed to failures. Similarly when we fix $\delta$ and increase the value of the scale parameter $\gamma$.

These models have the ability of modeling the reliability of both improving and deteriorating systems and have gained a wide acceptance in practice [10]. The general procedure for model selection should depend solely on the data. Using some risk criterion such as Bayes rule, maximum likelihood or minimum least-squares, the model parameters are estimated using some function of the observations $X_i$’s. The same set of data is used afterwards to validate the model through some goodness-of-fit tests. For a thorough description of the steps of model estimation and validation, the reader is referred to classical failure time analysis texts [11], [13] and [14].

III. Ranking Methodology

We use two ranking methods which are as follows: The first method is based on the hypothesis testing with ordered alternatives where we intend to test whether a given ordering of several products is valid or not, this is based on the observed characteristics of the vulnerabilities (severity score and time between vulnerability occurrences). The second method is based on the estimated intensity function, evaluated at the time of occurrence of the last vulnerability.

A. Hypothesis Testing with Ordered Alternatives

Due to the random nature of the severity scores and vulnerabilities inter-arrival time, we need to consider stochastic ordering among different services or products based on their susceptibilities to attacks. Stochastic ordering is defined among random variables through their corresponding cumulative distribution functions (cdf): a random variable $X$ is said to be stochastically smaller than a random variable $Y$, we denote $X \leq_{ST} Y$, if $F_X(x) < F_Y(x)$, where $F_X(.)$ and $F_Y(.)$ are the cdf’s of $X$ and $Y$ [5]. This can be interpreted as: $Y$ assumes larger values than $X$ does. For example, if we compare two products based on the time between occurrences of vulnerabilities, the better product would have larger durations between vulnerabilities.

Since the distribution of the characteristic of interest is usually unknown, we select a sample of random observations that are related to the characteristic of interest, and use their values to estimate the cdf with an empirical one and conduct tests of hypothesis to compare the distributions. The hypotheses in this case are:

- $H_0: F_x = F_y$
- $H_a: F_x < F_y$

When more than one sample are at hand and we want to check if the first population has larger observations than the
second, the second population has larger values than the third, etc., the hypotheses become:

\[ H_0 : F_1 = F_2 = \ldots = F_k \]
\[ H_a : F_i \leq F_2 \leq \ldots \leq F_k \]

This type of hypothesis is called Ordered Alternatives [15]. Notice that when \( F_i(.) \) is a location family, i.e., it can be written as \( F_i(x) = F(x - \mu_i) \) for all \( i \), the test becomes means ordering alternatives:

\[ H_0 : \mu_1 = \mu_2 = \ldots = \mu_k \]
\[ H_a : \mu_1 \leq \mu_2 \leq \ldots \mu_k \] with at least one inequality.

There are many statistical tests which can identify the order among the alternatives [16, 17]. Choosing which test to use depends on the assumptions. A frequent assumption is that the samples are selected from populations with distribution \( \{F_1, F_2, \ldots , F_k\} \) [4], [15], [16], and [17]. This method does not require minimal to no assumptions about the underlying distributions. A popular method for detecting order is the Jonckheere-Terpestra test (JT-test) [4], [15] or use a large sample approximation of T [15], [16]. The above procedure leads to an exact test, i.e., one that delivers an exact p-value with pre-determined level of significance. It can also be used to obtain the p-values of the Bartholomew test.

### B. Ranking Based on Testing with Ordered Alternatives

Based on the tests described in the previous section, i.e., the ordered mean alternatives (B-test) and stochastic ordered alternatives (JT-test), we can develop a ranking methodology of different softwares or services by considering all possible ordered alternatives.

Suppose that we have \( k \) services, \( S_1, S_2, \ldots , S_k \), that we would like to rank based on one of the vulnerability characteristics (severity score, time between occurrences). We conduct tests with ordered alternatives of the form:

\[ H_0 : F_{i_1} = F_{i_2} = \ldots = F_{i_k} \]
\[ H_a : F_{i_1} \leq F_{i_2} \leq \ldots \leq F_{i_k} \] with at least one strict inequality.

where \( \{i_1, i_2, \ldots , i_k\} = \{1, 2, \ldots , k\} \), in other words, set of the form \( \{i_1, i_2, \ldots , i_k\} \) where \( j \neq i_j \) are permutations of \( \{1, 2, \ldots , k\} \).

In all, we will be conducting \( k! \) tests of hypotheses, out of which, several will decide in favor of the alternative hypotheses. This is due to the fact that some \( F_i \) are not significantly different. For example, if \( F_1 < F_2 < F_3 = F_4 < F_5 \), then the following two tests will decide both in favor of the alternatives:

\[ H_a : F_1 = F_2 = F_3 = F_4 = F_5 \]
\[ H_a : F_1 \leq F_2 \leq F_3 \leq F_4 \leq F_5 \] with at least one strict inequality.

and

\[ H_a : F_1 = F_2 = F_3 = F_4 = F_5 \]
\[ H_a : F_1 \leq F_2 \leq F_4 \leq F_3 \leq F_5 \] with at least one strict inequality.

So both alternatives are correct and this is not a contradiction, since \( F_3 = F_4 \), and the inequalities in the alternative hypotheses are not strict.

This raises the problem of large number of concurrent alternatives, i.e., we have several ordering of the \( F_i \) all correct with probability of type I error that is equal to some preset value, \( \alpha \). In our analysis we will choose the level of significance \( \alpha \), depending on the number of products, to be no
more than 0.001. This means that the probability of deciding in favor of some false ordering of the products is no more than 0.001. Therefore to rank the services based on their concurrent orderings, we define some ranking rules:

1) **Ranking with the smallest p-value**: We consider the most significant alternative, i.e., the one with the most evidence against the null hypothesis. This alternative is identified as the one with the smallest p-value. Once this is done, we can assign ranks 1 to \( k \) to services based on the order of the corresponding distribution in the alternative with the smallest p-value. Notice that \( F_1 \leq F_2 \leq \ldots \leq F_k \), will not detect equality between two distributions. So, it would be left to the user to interpret if an inequality (\( \leq \)) is a strict one (\(<\)) or an equality. To fix this problem, we need to consider several alternatives and combine their information to distinguish between equalities and strict inequalities as we will show next.

2) **Ranking with measures of ranks central tendencies**: Let \( m \leq k! \) be the number of positive ordered alternatives. Since there are \( k \) products, we rank them between 1 and \( k \). We count the number of times a service \( S_i \) is in a position \( j \). Notice that for any service \( S_i, i = 1, \ldots, k \), the following relationship holds: \( \sum_{j=1}^{k} m_{ij} = m \). Hence, for a given service \( S_i \), we can assign relative frequencies, \( p_{ij} = m_{ij}/m \), to each of the positions that the service occupies. Based on the values, we can define measures of central tendencies of the rank of a service \( S_i \):

- The **Expected Rank**: the mean rank computed as the sum of the ranks weighted by the relative frequencies: 
  \[
  R_E(i) = \sum_{j=1}^{k} j p_{ij}.
  \]
- The **Median Rank**: the value of the ranks that has half of the ranks occupied by service \( S_i \) to its left and the other half to its right: 
  \[
  R_M(i) \text{ such that } \sum_{j=1}^{R_M(i)} p_{ij} \geq 0.5 \text{ and } \sum_{j=R_M(i)}^{k} p_{ij} \geq 0.5.
  \]

Based on the above formula of the ranks, we notice that \( R_M(i) \) and \( R_m(i) \) assume integer values between 1 and \( k \) while \( R_E(i) \) does not have to be an integer since the \( p_{ij} \) are rational numbers. This implies that if two products \( S_i \) and \( S_j \) behave similarly as far as the vulnerability exposure is concerned, we might have \( R_M(i) = R_M(j) \) and \( R_m(i) = R_m(j) \), while most likely \( R_E(i) \) and \( R_E(j) \) will be slightly different. For example, if the services are such that the distribution of their vulnerability’s characteristics are ordered as: \( F_1 < F_2 < F_3 = F_4 < F_5 \), we will obtain \( R_M(3) = R_M(4) \) and \( R_m(3) = R_m(4) \), while \( R_E(3) \) and \( R_E(4) \) will be very close to each other but slightly different. We recommend that users consider at least the two measures \( R_E(.) \) and \( R_M(.) \) and decide accordingly. We show later in the real data examples and simulations that the three different measures concur most of the time.

### C. Ranking Based on the Final ROCOF

Since there is no guarantee that the vulnerabilities occurrences are always independent, it wouldn’t be safe to assume that number of vulnerabilities over time is a stationary HPP. We work within the framework of the more inclusive NHPP, i.e., \( \text{Pr}(N(t) = k) = \{M(\nu(u))\} \text{exp}(\int_{0}^{t} \nu(u) du) \) where \( \nu(u) \) is not necessary a constant function. If \( \nu(t) \) is known to belong to one of the commonly used families, estimating its parameters will enable us to fully determine the function. Since this is seldom the case, it would be better to estimate \( \nu(t) \) non-parametrically. This can be done using different smoothing techniques. We choose the simplest and widely used kernel intensity estimator [22], [23], which defined, for the \( i^{th} \) sample, as:

\[
\hat{\nu}(t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \frac{1}{h} K \left( \frac{t - X_{i,t}}{h} \right)
\]

where the kernel \( K(u) \) is a continuous function that is symmetric about 0 and the smoothing parameter \( h \) is a sequence of numbers that decays slowly to 0 as the sample size increases.
Once the estimation is done for different systems, we can compare them based on their intensity function evaluated at the time of the last occurrence.

IV. EVALUATION

To evaluate the proposed ranking methodology, first we simulate few processes whose ordering or ranking is already known to show that the proposed framework is accurate. Next, we present a simulated example which shows the advantages of using nonparametric methods over parametric ones. Then, we apply the same framework to certain case studies of services whose data is available at NVD. We used the most common services or softwares which can be ordered intuitively. This shows that the framework proposed ranks those services correctly and can be used to rank any type of service using any metric(s).

A. Simulation for Stochastic Ranking Validation

Most of our data come from a NHPP, so it would be fit to simultaneously generate random variants from several such processes whose stochastic ordering are known and then apply our ranking methodologies to the simulated data in order to verify the proposed framework.

For our simulation study, we consider the PLP defined in section 2 with a ROCOF function $\nu_2(t) = \gamma \delta e^{t-1}$. Such choice is motivated by the flexibility of this model, due to its polynomial nature, and its ability to fit both fast and slow increasing, decreasing and constant failures rates as evidenced by its success in many applications [10], [12]. A service or a products with a large value of $\delta$ or a large value of $\gamma$ are exposed to more vulnerabilities than the ones with small values. Hence we generate 7 different samples, each corresponding to a product, first by fixing the scale parameter $\gamma$ and increasing values of $\delta$, and then by fixing the shape parameter $\delta$ and increasing values of $\gamma$. Products that are more susceptible to vulnerabilities (with large values of $\gamma$ or $\delta$) are given higher ranks: 7, 6, ... . We then apply the ordered alternatives ranking and the final ROCOF ranking methodologies.

The reason for choosing 25 observations is because it is considered a small sample in simulating statistical experiments and several of the real data that we collected form samples of sizes larger than 25. If the proposed ranking methodology works for this small sample then it would certainly performs better with larger samples since the probability of type II error decreases with the sample size. We choose 7 products for computational purposes since generating all permutations
requires running an exponentially complex algorithm.

Simulation was done using different values of $\gamma$ and $\delta$, whose values are mentioned in Table 1 and 2. It can be observed from both the tables that the proposed framework yields the accurate ranking (as compared to the known Real Rank), however it can be seen that for real rank 5 and 6 the expected rank (for both JT-based and Ordered Mean-based) is not accurate but if we take into account the mode rank and smallest p-value rank of JT-based, we can still rank them correctly. Moreover the proposed framework is still able to identify the most stable or less prone to attacks service, which is the essential purpose of the work to assist the designers and provide them the knowledge for available options in-terms of the reliability of a service.

B. Case Studies from NVD

In this section we discuss few case studies of services or softwares whose data is readily available at NVD, in-order to rank them and show the results obtained using the proposed framework. The lower rank value means the service or software is potentially more vulnerable, however higher rank means potentially less vulnerable. We showed the rankings using two methodologies which are JT-based and Ordered Mean-based. Details of these case studies are as under.

1) OS Example: Figure 1 shows the ranking of eight operating systems. It can be easily observed that Linux Distributions are potentially less vulnerable as compared to other OSes (Windows, MAC etc). It doesn't mean that they are not designed properly; however, they are potentially more prone to attacks due to their day to day use by majority of the people. We cannot conclude that Windows XP is more vulnerable than Debian Linux due to its design sophistication, since we are not taking into account the sophisticated design parameters in this study. However if design parameters are publicly available, this framework can be used to rank them. NVD database is not granular enough to provide the data that which module has caused the vulnerabilities to expose. Availability of component level vulnerability can be be helpful in designing the network.

2) HTTP Service Example: Figure 2 shows the ranking for Hypertext Transfer Protocol (HTTP). It is clear that apache is potentially more vulnerable than any other http server. This makes sense since it is the most popular or widely used http server available today. IIS (Internet Information Server) is almost alike the apache and popular as well. If we consider all the rankings shown in the figure, we can see that the results form two groups, in first group apache and IIS are there however rest of the http servers are in the other group.

3) Browser Example: Similarly we ranked the popular browsers whose ranking is shown in Figure 3. It also shows that most popular browsers exposes the vulnerability in short time intervals. Ranking yields that the Internet Explorer and Firefox are potentially more vulnerable. This is intuitive since they are widely used and popular browsers as well. Therefore the framework provides a statistical method to rank services in-order to design the network which is less prone to attacks.

V. CONCLUSION

Risk-based Ranking of network services is highly useful for not only recommending products but also for optimizing security design and policies based on quantitative justifications. The NVD by NIST hosts one of the largest databases that keep track of software vulnerability including severity score and inter-arrival times. The paper presents a new statistical framework to rank the services based on NVD data. Our results were rigorously validated using simulation and real-life data. We observed that services that are assigned the similar ranking class usually share common popularity level. Highly popular products might exhibit lower ranking. Please note that our objective is to introduce stochastic ranking for risk ordering and not necessarily identifying software risk factors.
REFERENCES


