Quality Engineering

ISSN: 0898-2112 (Print) 1532-4222 (Online) Journal homepage: https://www.tandfonline.com/loi/lqen20

Control charting methods for autocorrelated cyber vulnerability data

Anthony Afful-Dadzie & Theodore T. Allen

To cite this article: Anthony Afful-Dadzie & Theodore T. Allen (2016) Control charting methods for autocorrelated cyber vulnerability data, Quality Engineering, 28:3, 313-328, DOI: 10.1080/08982112.2015.1125926

To link to this article: https://doi.org/10.1080/08982112.2015.1125926

Published online: 31 Mar 2016.

QUALITY ENGINEERING
2016, VOL. 28, NO. 3, 313–328
http://dx.doi.org/10.1080/08982112.2015.1125926

Taylor & Francis
Taylor & Francis Group

Control charting methods for autocorrelated cyber vulnerability data

Anthony Afful-Dadzie[a] and Theodore T. Allen[b]

[a]Business School, University of Ghana, Accra, Ghana; [b]Integrated Systems Engineering, The Ohio State University, Columbus, Ohio

ABSTRACT
Control charting cyber vulnerabilities is challenging because the same vulnerabilities can remain from period to period. Also, hosts (personal computers, servers, printers, etc.) are often scanned infrequently and can be unavailable during scanning. To address these challenges, control charting of the period-to-period demerits per host using a hybrid moving centerline residual-based and adjusted demerit (MCRAD) chart is proposed. The intent is to direct limited administrator resources to unusual cases when automatic patching is insufficient. The proposed chart is shown to offer superior average run length performance compared with three alternative methods from the literature. The methods are illustrated using three datasets.

KEYWORDS
autocorrelation; average run length (ARL); control charts; EWMA control charts; statistical control

Introduction

Cyber attacks are on the increase and many organizations are losing substantial amounts of money as a result. A study of the financial impact, customer turnover, and actions taken by 51 companies in the United States concluded that, on average, the cost of a successful attack in 2010 increased to $7.2 million, up 7% from $6.8 million in 2009 (Ponemon Institute 2011). Cyber vulnerabilities are ways that hosts such as personal computers, servers, and printers can be exploited. Examples of vulnerabilities include: weak passwords, weak authentication processes, unsupported operating systems, information disclosures, and the use of software with known exploitable bugs. Reportedly, over 90% of successful attacks exploit known vulnerabilities for which a patch exists but has not been applied by the system administrators (Legard 2002). Therefore, while new technology to identify and patch vulnerabilities is important, securing and focusing human resources to eliminate known vulnerabilities is also important.

The objective of this article is to propose control charting methods for cyber vulnerabilities to direct the attention of system administrators to unusual occurrences that correspond to assignable causes that they can address. As noted in Afful-Dadzie and Allen (2014), a substantial fraction of vulnerabilities are repaired each month by automatic patching without local intervention. Typically, only a tiny fraction of vulnerabilities are repaired manually because of automatic patching and limited resources. As a result, it may be of interest for administrators to intervene only when there is something unusual occurring (i.e., an assignable cause) or, alternatively, a major threat is clear (e.g., an on-going attack). Therefore, this article focuses on a statistical process control approach designed to signal the presence of assignable causes.

Previous authors have developed monitoring techniques relating to cyber vulnerabilities. Yet, some have used data that is not available in vulnerability reports. For example, Dowdy (2012) discusses the challenges in integrating data from many sources to summarize risks. Abedin et al. (2006) also use traffic volumes as part of a comprehensive network evaluation approach. Further, Abedin et al. (2006) introduce exponential functions in their formulations which potentially complicate the interpretation. Others authors have based their metrics on forecasted quantities without invoking the concepts from statistical process control (Ahmed et al., 2008). In this article, a relatively simple monitoring technique based on readily available data and statistical process control is proposed.

Cyber vulnerability data are often provided monthly with reference to the Common Vulnerability Scoring

CONTACT Theodore T. Allen allen.515@osu.edu Integrated Systems Engineering, The Ohio State University, 1971 Neil Avenue – 210 Baker Systems, Columbus, OH 43210.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/lqen.

System (CVSS) described in Mell et al. (2007). CVSS scores range from 0.0, meaning no vulnerability, to 10.0, indicating that the program evaluating the system (scanner) is in a position to take over the host system. Common scanning technology divides vulnerabilities into categories based on the CVSS score: low (0.0–3.9), medium (4.0–6.9), high (7.0–9.9), and critical (10.0). A given host could have multiple vulnerabilities, e.g., 2 mediums and 1 critical. Therefore, the situation is somewhat analogous to manufacturing with nonconformity counts of different levels of severity. Weightings of these counts or demerits and the associated demerit charting techniques are potentially relevant in this case (e.g., see Nembhard and Nembhard 2000). Also, because of the infrequent (often monthly) nature of relevant data, charting without subgroups or "individuals" control charting of demerits is relevant.

However, unlike in manufacturing, the hosts (or units) with the vulnerabilities (or nonconformities) are not shipped each period. Instead, these hosts might be personal computers which are used for multiple months and might likely have the same vulnerabilities for an extended period. On-going "patching" eliminates a fraction of the vulnerabilities each month, but far fewer than 100%. The accumulation of vulnerabilities almost unavoidably induces autocorrelation or correlation in period-to-period nonconformity counts. Autocorrelation is a major issue related to control chart performance (Alwan and Roberts 1988; Montgomery and Mastrangelo 1991; Runger and Willemain 1995; Loredo et al. 2002; Nembhard and Nembhard 2000).

An additional complication is that local vulnerabilities are influenced by external causes including continual discoveries of new vulnerabilities for the software in use. These phenomena could cause a constant increase in vulnerability counts over time on many systems (Alhazmi and Malaiya 2005).

Charting based on autoregressive (AR) moving average modeling promises to eliminate the adverse effects of autocorrelation and trending because the model residuals are generally uncorrelated and detrended (Montgomery and Mastrangelo 1991; Runger and Willemain 1995). Perhaps the simplest of the relevant schemes is based on the first-order autoregressive or AR(1) model. Authors have noted the ability of such approaches for addressing autocorrelation as well as underlying trends (Runger and Willemain 1995). Another relevant approach is moving centerline demerit (MCD) charts which offer the advantage that the charted quantity is intuitive, i.e., it is the demerits per unit (Nembhard and Nembhard 2000). Yet, Runger and Willemain (1995) noted the poor average run length performance of residual charts given their diminished capacity to identify shifts after the first subgroup following the shift (Runger and Willemain 1995). MCD charts are also based on residuals and can be expected to have similar performance. This deficiency motivates two new charts that are proposed in this article. These are "adjusted demerit" (AD) and hybrid moving centerline residual-based demerit and adjusted demerit (MCRDAD) charts for monitoring cyber vulnerabilities. An average run length (ARL) comparison is also described to confirm the benefits of the proposed methods.

The remainder of this article is organized as follows. First, alternative statistical process control (SPC) charts relevant to cyber vulnerability data are described. Because of the repeat nature of cyber vulnerabilities, the focus is on procedures specifically addressing autocorrelated data. The reviewed procedures include moving centerline demerit (MCD) charts from Nembhard and Nembhard (2000) and moving centerline charts based on AR(1) residuals. Issues with residual-based charting are used to motivate the proposed adjusted demerit (AD) and moving centerline residual-based and adjusted demerit (MCRAD) methods. The average run lengths (ARLs) of the alternative methods are then compared. Next, the application of the proposed methods is illustrated using three cyber vulnerability datasets from different organizations. Finally, conclusions are presented and opportunities for future work are described.

Statistical process control charting

In this section, four alternative methods are described. As mentioned previously, the carryover of vulnerabilities from one period to the next causes a high degree of autocorrelation in related vulnerability data. Therefore, the focus here is on methods specifically addressing autocorrelation rather than general techniques such as exponentially weighted moving average (EWMA) charts. Also, the charting methods can be applied both retrospectively as an analysis technique and also built into scanning software for active monitoring.

The first alternative method explored here is moving centerline demerit (MCD) charting from Nembhard

and Nembhard (2000). The second is a trivial combination of the residual charting methods from Runger and Willemain (1995) and the moving centerline concept from Nembhard and Nembhard (2000). Next, an adjusted demerit chart and a hybrid moving centerline residual-based and adjusted demerit (MCRAD) methods are proposed. The motivations for the proposed methods relate to the objectives of improved average run length performance and interpretability.

Residual-based charts

In general, the residuals of a defensible time series model are approximately independent, and identically distributed from a normal distribution assuming that the process is under control (for example, when there are no shifts). The properties of the residuals can be evaluated using autocorrelation function (ACF) and partial autocorrelation function (PACF) residual plots. The charting of residuals from time series models such as AR(1) was described in Runger and Willemain (1995). For the AR(1) processes, the model prediction can be written:

$$\hat{y}_i = \mu + \varphi y_{i-1} \qquad [1]$$

and the model residual is simply:

$$\hat{\varepsilon}_i = y_i - \hat{y}_i. \qquad [2]$$

where y_i is the dependent variable (or the demerit per unit in our model) at time period i, $\hat{\varepsilon}_i$ is a white noise with zero mean and constant variance, and μ and $-1 < \varphi < 1$ are constants to be determined. The symbol "^" denotes the estimated or predicted value based on the data, y_i for time period $i = 1, \ldots, p$. In a residual chart, the charted quantity is $\hat{\varepsilon}_i$ in (2).

Nembhard and Nembhard (2000) examined charts based on residuals in Eq. [2] and proposed two modifications. First, they argued that charting of residuals is not intuitive for decision-makers in that they are generally more interested in the process mean than the model residuals. Instead of residual charting, they proposed using a moving centerline based on model predictions and moving limits based on the standard deviation of the residuals. Their proposed approach is in accord with the insights in Alwan and Roberts (1988), who had argued that residual charting was insufficient, while offering the simplicity of a single chart. The Nembhard and Nembhard (2000) moving centerline approach is functionally identical to residual charting in that the charts would deliver the same out-of-control signals in identical situations and yet the charted quantity is the demerits per unit. Second, Nembhard and Nembhard (2000) argued that time series modeling might be too complicated for many possible users and exponentially weighted moving average (EWMA) offers similar predictions with only a single adjustable parameter, λ. Therefore, they based the centerline (CL_i) of their moving centerline demerit (MCD) chart on the following EWMA formula:

$$CL_i = \hat{y}_{i+1} = \lambda y_i + (1 - \lambda)\,\hat{y}_{i-1} \qquad [3]$$

where λ is the weight given to the most recent weighted value and must satisfy $0 < \lambda \leq 1$

Then, the MCD upper control limit UCL_{i+1}, and lower control limits LCL_{i+1} are:

$$\begin{aligned} UCL_{i+1} &= \hat{y}_i + M\hat{\sigma} \\ LCL_{i+1} &= \hat{y}_i - M\hat{\sigma} \end{aligned} \qquad [4]$$

where M is a potentially adjustable parameter given in Nembhard and Nembhard (2000), and usually $M = 3.0$. The parameter $\hat{\sigma}$ is the standard deviation for the one step ahead prediction errors $e = y_i - \hat{y}_i$, which are independent and uncorrelated with mean of zero. Nembhard and Nembhard (2000) proposed two procedures for estimating λ and $\hat{\sigma}$. The first of which is used for illustration and involves selecting λ to minimize the sum of squared residuals and $\hat{\sigma}$ as the root mean squared residual.

A trivial variant of the MCD charts is to simply base the predictions on the time series model in Eq. [1] instead of the EWMA model in Eq. [3]. This approach offers the benefit of MCD charts in that the charted quantity is the intuitive demerits per unit. Also, the predictions are based on the likely more accurate time series models instead of the EWMA model. The proposed variant is referred to as moving center-line residual-based demerit (MCRD) charts. The MCRD is slightly different than a residual chart because unlike the residual chart the MCRD will adjust the lower limit to zero in situations when the calculated lower control limit is negative.

As mentioned previously, MCD and MCRD charts are approximately equivalent to residual charts in the signals generated. Also, Runger and Willemain (1995) documented the average run-length (ARL) properties of residuals charts with two notable findings. First, residual charts offer run-length performance that may

be considered poor based on the tables provided by Runger and Willemain (1995) compared with alternatives for cases without autocorrelation and EWMA charts. Second, the poor performance relates to the fact that residual charts offer a relatively high probability of generating an out-of-control signal in the first subgroup after a shift. After the first subgroup, the chance of detecting the shift is greatly diminished. In the next section, two charting techniques are proposed with the objective of offering improved ARL performance compared with MCD and MCRD charting procedures.

Adjusted demerit charts

Standard demerit charts are generally considered to be inapplicable to cases involving significant autocorrelation (e.g., see Montgomery, 2012). These charts are based on the assumption of independently Poisson distributed demerits. While the Poisson distribution seems approximately appropriate for weighted vulnerability counts, the assumption of independence from period to period does not apply because of the significant autocorrelation. In what follows, we first present the standard demerit control chart model, point out its limitations to charting demerits per unit of cyber vulnerability data, and proposed an adjusted demerit control chart for overcoming such limitations.

The standard demerit control chart formulas from Dodge (1928) are derived as follows. Let d_i be the weighted total number of demerits in period i, n_i be the sample size, and c_{ik}, be the number of class k nonconformities, $k = 1, 2, \ldots, m$. If w_k is the weight of nonconformity class k, the weighted demerits d_i, and the demerit per unit D_i (which is the charted quantity and referred to in this article as demerit per host) in period i are:

$$d_i = \sum_{k=1}^{m} w_k c_{ik}$$

and

$$D_i = \frac{d_i}{n_i}. \quad [5]$$

The average number of demerits, $\bar{D}_k$, across all the p periods, for nonconformity class k is:

$$\bar{D}_k = \frac{\sum_{i=1}^{p} c_{ik}}{\sum_{i=1}^{p} n_i}. \quad [6]$$

Then, the center line (CL) of the demerit control chart is:

$$CL = \sum_{k=1}^{m} w_k \bar{D}_k. \quad [7]$$

The upper and lower control limits for period i are:

$$UCL_i = CL + M\hat{\sigma}_i$$

and

$$LCL_i = \max\left[\left(CL - M\hat{\sigma}_i\right), 0\right] \quad [8]$$

where

$$\hat{\sigma}_i = \sqrt{\frac{\sum_{k=1}^{m} w_k^2 \bar{D}_k}{n_i}}, \quad [9]$$

and where M is a potentially adjustable parameter which, in standard demerit charts, is 3.0.

The standard demerit chart given above is likely to foster high false alarm rates if applied to charting cyber vulnerabilities for the following reasons. The estimated standard deviation in Eq. [9] is based on the assumption that the demerit counts of different levels of severity are uncorrelated. For the cyber vulnerabilities in the case studies shown later, at least two counts of vulnerabilities are significantly correlated for all three of the organizations considered. In the presentation here, the anonymous organizations are assigned labels corresponding to their size, so that organization #1 had the most hosts. For example, the correlation between the high and critical counts for organization #1 in Table 1 is 0.97 which is significant with a p-value less than 0.001. Also, Dodge and Romig (1928) assumed that the charted quantities (demerits per host), y_i, exhibit no autocorrelation if the system is under statistical control. As noted in Table 4 later, the autocorrelation coefficients are significant for all three organizations.

It was the violations of assumptions of control charts that motivated new methods such as those in Runger and Willemain (1995) and Nembhard and Nembard (2000). Runger and Willemain (1995) evaluated residual charts and determined that applying individuals control charts (e.g., see Montgomery, 2012) to batched

Table 1. The estimated AR(1) parameters for the three organizations (cases).

	Case 1	Case 2	Case 3
Coefficient ($\hat{\varphi}$)	0.920	0.697	0.646
Mean ($\hat{\mu} = \frac{\mu}{1-\varphi}$)	3.178	5.490	1.999
Sigma ($\hat{\sigma}$)	1.110	1.569	1.381

observations offered relatively desirable average run lengths. Then, instead of charting the demerits per host for each period, y_i, one would chart the average of m subgroups. Runger and Willemain (1995) recommended batch sizes to reduce the autocorrelations to less than 0.1. For cases such as organization #1 with autocorrelation coefficients greater than 0.98, the recommended batch size was $m = 58$. With each period lasting a single month, there would be a single subgroup every 4.8 years, which is impractical for cyber vulnerability charting.

With the goal of providing desirable average run length performance with an intuitive charted quantity, the following adjusted demerit charting procedure is proposed.

Step 1: Apply time series modeling from Box and Jenkins (1994) to develop a time series model of the demerits per host. For example, Table 1 shows the coefficients for the AR(1) models derived in the case studies.

Step 2: Obtain a value of M in Eq. [7] such that the average run length (ARL) with the process in control achieves a desired value, e.g., ARL(in control) = 200.0. The value of M can be determined using simulation based on the model derived in Step 1. Apply charting to the demerits per host using the derived M values.

The above adjusted demerit procedure is facilitated by modern computing. Using this procedure, there is no assumption about the autocorrelation or cross correlation other than that it can be modeled appropriately in Step 1.

Moving centerline residual-based and adjusted demerit charts

An alternative approach is to chart the demerits per host using limits from both moving centerline residual-based (MCR) and adjusted demerit (AD) charts. If the demerits per host cross any of the control limits, an out-of-control signal is generated by the derived hybrid moving centerline residual-based and adjusted demerit (MCRAD) chart. Let M_1 refer to the parameter in Eq. [4] associated with MCD limits and M_2 to refer to the parameter in Eq. [7] associated with AD limits. As a default and for simplicity, we set $M_1 = 3.0$ and then find M_2 using a two-step procedure similar to the one for determining adjusted demerit chart limits. A user might seek even greater average run length performance by optimizing simultaneously over M_1 and M_2. It is also possible, the desired in-control average run length cannot be attained using the default value of $M_1 = 3.0$. Then, both M_1 and M_2 should be adjusted simultaneously to achieve desired in-control average run length with, again, the in control model being the estimated time series model.

Comparison of average run lengths

In this section, the four charting procedures are compared using average run lengths (ARLs). While ARL calculations are skewed by rare long run lengths, we include them to provide a direct comparison with previous research on charts for autocorrelated data. The derived ARL values are based on a simulated demerit per host data from an autoregressive model. The four charting procedures to be compared are: moving centerline demerit (MCD) from Nembhard and Nembhard (2000), moving centerline residual-based demerit (MCRD) which is an extension of residual charts from Runger and Willemain (1995), adjusted demerit (AD), and moving centerline residual and adjusted demerit (MCRAD) charts. The ARL values are estimated using 20,000 simulations in which the shift (δ) occurs on the first subgroup with the initial subgroup being subgroup zero following the procedure in Runger and Willemain (1995). Therefore, all the ARL estimates have standard deviations less than 1% (0.007 × standard deviation) of the estimated ARL values making virtually all comparisons significant simultaneously. Therefore also, after the first subgroup all responses derive from Eq. [10] with δ (in increment of 0.5) added. In each case, the simulated demerits per host derived from the standard AR(1) model of the demerits per host (y_i) with a single lag can be written for period i:

$$y_i = \mu + \varphi y_{i-1} + \varepsilon_i, \qquad [10]$$

where the ε_i are assumed to be independent identically distributed (IID) N(0, σ^2). The coefficients, μ and φ, can be estimated through least squares regression using a lag variable, which is available in standard software under the time series menus.

Table 1 contains the three sets of parameters needed for simulating the demerit per host data. These were obtained from the three case study datasets described later. The related ARL results are shown in Tables 2–4, where values under $M = 3$ are presented to show the

Table 2. Average run length values for an AR(1) process with estimated parameters $\varphi = 0.920$, $\mu = 3.178$, and $\sigma = 1.110$ based on data from Case 1.

	MCD		MCRD		AD		MCRAD
δ/σ	$M = 3.0$	$M = 2.85$	$M = 3.0$	$M = 2.725$	$M = 3.0$	$M = 20.9$	$M_2 = 22.79$
0.0	314.21	200.37	484.59	199.43	2.20	200.00	199.90
0.5	297.47	189.37	448.29	183.27	1.97	144.24	151.98
1.0	278.44	177.10	401.20	165.40	1.61	104.56	114.53
1.5	256.47	161.23	358.64	142.48	1.33	77.22	84.21
2.0	223.98	134.84	294.63	114.13	1.13	56.59	58.77
2.5	176.05	105.49	220.37	82.18	1.04	41.92	38.01
3.0	125.81	70.60	146.87	51.32	1.01	31.46	22.29
3.5	74.77	39.78	83.10	27.08	1.00	22.83	11.88
4.0	39.31	19.72	39.48	11.32	1.00	16.29	5.51

Table 3. Average run length values for an AR(1) process with estimated parameters $\varphi = 0.697$, $\mu = 5.490$, and $\sigma = 1.569$ for Case 2.

	MCD		MCRD		AD		MCRAD
δ/σ	$M = 3.0$	$M = 3.65$	$M = 3.0$	$M = 2.77$	$M = 3.0$	$M = 7.06$	$M_2 = 7.51$
0.0	48.47	199.21	436.14	199.98	5.90	199.31	200.09
0.5	46.83	192.58	352.51	167.69	4.91	88.81	105.17
1.0	44.11	186.23	244.80	124.08	3.60	43.31	53.64
1.5	38.86	176.32	164.14	83.05	2.46	23.47	28.52
2.0	32.13	158.91	101.09	50.35	1.64	13.31	15.61
2.5	22.41	126.80	55.81	28.24	1.26	7.80	8.48
3.0	13.72	90.74	28.06	14.37	1.09	4.67	4.61
3.5	7.14	56.66	12.68	6.42	1.02	2.81	2.55
4.0	3.42	27.57	5.39	2.83	1.01	1.82	1.61

relatively arbitrary performance levels if the standard choices are used. For example, the $M = 3$ in-control run lengths for the demerit charts are generally so short that false alarms would make their application prohibitively expensive.

For the MCRAD chart, the simulations involve values of $M_1 = 3.0$ in Eq. [4] and M_2 in Eq. [8] that generate ARL in-control (with no assignable causes active, $\delta/\sigma = 0.0$) values approximately equal to 200. In all cases where the ARL in-control value is approximately 200, the ARL values for the MCD chart exceed that of the MCRD chart. This is explained by the fact that, in using the same AR(1) internalized within the MCRD chart, an advantage is conferred to the MCRD chart. In other words, the MCRD charting method is designed to directly address the test cases such that its residuals are IID N(0, σ^2). Similarly, the ARL for the MCD and MCRD exceed that for the AD and MCRAD charts. The exception is for the largest shifts ($\delta/\sigma = 4.0$) under the assumptions in Table 2. Then, the MCRD chart offers a lower average run length than the AD chart. Yet, the MCRAD chart dominates the MCD and MCRD charts in all cases. Therefore, it is concluded that the use of MCD or MCRD charting in the context of cyber vulnerabilities is generally inadvisable since AD and MCRAD methods offer generally superior ARL performance. This assumes that the ability to perform time series modeling and simulation is within the capabilities of the practitioners. The authors have excel-based software available upon request for generating the M

Table 4. Average run length values for an AR(1) process with estimated parameters $\varphi = 0.646$, $\mu = 1.999$, and $\sigma = 1.381$ for Case 3.

	MCD		MCRD		AD		MCRAD
δ/σ	$M = 3.0$	$M = 2.69$	$M = 3.0$	$M = 2.605$	$M = 3.0$	$M = 2.405$	$M_2 = 2.49$
0.0	547.33	200.20	721.67	200.78	697.97	199.80	203.67
0.5	405.50	150.20	411.65	123.32	239.46	84.00	88.87
1.0	320.52	121.89	235.88	76.36	97.53	39.28	41.69
1.5	263.57	98.60	137.92	45.32	44.38	20.09	21.60
2.0	218.58	80.79	78.02	25.71	22.36	10.99	11.51
2.5	174.25	58.00	40.68	13.78	11.77	6.38	6.29
3.0	124.83	38.79	20.07	6.55	6.97	3.79	3.56
3.5	78.21	21.53	9.02	3.14	4.20	2.36	2.16
4.0	39.37	9.82	3.74	1.72	2.52	1.57	1.43

or M_2 values required by the AD and MCRAD charts, respectively.

Further, the AD chart dominates the MCRAD chart for small shifts ($\delta/\sigma < 3.0$) while the MCRAD dominates for large shifts ($\delta/\sigma > 3.0$). This corroborates Runger and Willemain (1995). Residual-based charts offer a relatively high probability of identifying a large shift on the first subgroup, but adjusted demerit charts offer improved detection probabilities in other cases. The differences are larger for the assumptions in Table 2 which is attributed here to the higher degree of autocorrelation. Runger and Willemain (1995) also found larger differences in ARL values among alternative charts when the degree of autocorrelation was relatively high. The MCRAD charts are recommended for cyber vulnerability modeling because we feel that the ability to quickly detect large shifts is likely relevant in applications. Yet, the AD charts also offer relative simplicity and competitive ARL performance making them a viable alternative.

Case studies

In this section, the case studies that motivated our research are described. The section begins with the steps taken to prepare the data for attribute charting and the report from the local system administrator about assignable causes. Next, the applications of Box-Jenkin's time series modeling are then described. Finally, results illustrate possible insights gained using the proposed adjusted demerit (AD) and moving centerline residual-based and adjusted demerit (MCRAD) charting procedures.

Vulnerability data preprocessing

The organizations under study had (altogether) 498 hosts over a 28-month period using data from the monthly Nessus scans. Nessus is a vulnerability scanning software developed by Tenable Network Security, widely regarded as a world leader in vulnerability scanning. One of the main challenges during a scan is that, if a host is turned off or its firewall is turned on, it would not appear in the final vulnerability report even if it had vulnerabilities.

The steps to generate attribute data were as follows.

Step 1. Identify all distinct vulnerabilities across all hosts and all 28 months. For example, host 1 might have vulnerability 23 (out-of-date operating system) and vulnerability 35 (weak password) while host 2 might have vulnerability 35 only. Combining results from all three of our case studies results in 183 distinct vulnerabilities.

Step 2. List the specific hosts known to have each of the observed vulnerabilities in each month. Table 5 shows a portion of this listing. The numbers in the table are the CVSS scores for the specific vulnerabilities. If an item is blank it implies that either the host did not have the vulnerability or the host was unavailable during the scanning period. It was found that only 36 of the 498 hosts exhibited any vulnerability during the 28 months. Therefore, the vulnerabilities were concentrated on approximately 7% of the hosts.

Step 3. Tabulate the counts of low, medium, high, and critical vulnerabilities on all hosts for each month. Table 6 shows the counts of vulnerabilities of different levels of severity for two of the hosts. The instances in which hosts were unavailable are marked with borders and bolding.

Step 4. Impute the missing data using the sample averages of the counts from the months before and after each instance (possibly including multiple month gaps), i.e., mean-based imputation was applied (Enders 2010). For missing data in the first or last months, counts were inserted from the closest month in time for which there were data. Note that such imputations are likely necessary as missing data in vulnerability datasets is the common result of hosts being unavailable during the system scans. The results are shown in Table 6 in bold font.

Step 5. Tabulate the total number of sampled hosts (n_i) successfully scanned in each period i and the total counts (c_{ik}) for severity levels $k = 1, \ldots 4$ for low, medium, high, and critical vulnerabilities.

Step 6. Calculate the demerits (D_i) per period i using:

$$D_i = \sum_{k=1}^{4} w_k c_{ik} \qquad [11]$$

with weights $w_1 = 2.0$, $w_2 = 5.5$, $w_3 = 8.5$, and $w_4 = 10$, for low, medium, high, and critical, respectively. These weights are determined with reference to the common vulnerability scoring system (CVSS; Mell et al. 2007). Also, the demerits per host, y_i, were derived using $y_i = D_i = d_i/n_i$. The resulting data are shown in Table 7 for

Table 5. Excerpt of the data of vulnerabilities and their CVSS scores for a single month.

Host#	month	Vul 1	Vul 2	Vul 3	Vul 4	Vul 5	Vul 6	Vul 7	…	Vul 183
1	3	5	5						…	
2	3			4.3					…	
3	3		5		2.6	2.6			…	
4	3						5.1	2.6	…	
5	3								…	
6	3						5.1	2.6	…	
7	3								…	
8	3								…	
9	3								…	
10	3								…	
11	3								…	
12	3								…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
498	3								…	2.6

the largest of the three organizations. Tables A1 and A2 in the appendix contain similar information for the other organizations.

The choice of mean-based imputation in Step 4 was motivated by our inspection of the available data for the 36 hosts for which there were vulnerabilities. Table A3 in the appendix provides the CVSS score for the most severe vulnerability on each host by CVSS score. The data indicate a high degree of constancy in the vulnerabilities on the hosts. The 462 hosts not shown in Table A3 are believed to have had no known vulnerabilities during the entire 28 months.

The system administrator was also interviewed for organization #1, which was the largest of the three. The administrator reported taking 16 manual actions during the 28 month period following direct requests from the host users. These included manually applying patches for hosts with automatic patching turned off, identifying false positives (vulnerabilities reported in the Nessus scan with little chance of causing actual intrusions), and changing host permissions. The actions also included the resolution of three then ongoing cyber-attacks. This was accomplished through removing host permissions and vulnerable software manually. There was no awareness of any actions taken with respect to hosts in organizations #2 and #3 during the 28 months.

Despite the actions taken by the administrator to patch a select number of hosts, the administrator perceived only a single unusual occurrence or assignable cause during the 28 months. The remaining variation was perceived to be typical or associated with common causes only. The assignable cause occurred during month 19 and began to affect counts on month 20. During month 19, there was a major organizational change and the administrator lost responsibility for approximately 200 hosts. This change included none of the hosts having vulnerabilities in Table A4.

Table 6. Vulnerability counts for two hosts with imputed data bolded.

	Host 1				**Host 2**			
Month	Low	Medium	High	Critical	Low	Medium	High	Critical
1	0	1	0	0	0	2	2	0
2	0	1	0	0	0	1	2	0
3	0	1	0	0	**0**	**1.5**	**2**	**0**
4	**0**	**1**	**0**	**0**	0	2	2	0
5	0	1	0	0	0	1	2	0
6	0	1	0	0	0	2	2	0
7	0	1	0	0	0	2	2	0
8	0	2	0	0	0	2	2	0
9	0	2	0	0	0	2	2	0
10	0	1	0	0	0	2	2	0
11	0	2	0	0	**0**	**3**	**2**	**0**
12	**0**	**1.5**	**0**	**0**	0	4	2	0
13	0	1	0	0	0	4	2	0
14	0	1	0	1	0	3	2	0
15	0	1	0	1	0	2	2	0
16	0	1	0	1	0	2	2	0
17	0	1	0	1	0	3	3	0

Table 7. Tabulation of the hosts scanned successfully and numbers of vulnerabilities of different levels of severity for Case 1. Also included are demerit and demerit per host data.

		Number of Vulnerabilities					
Month	n_i	Low	Medium	High	Critical	d_i	$y_i = D_i$
1	199	12	33	8	0	273.5	1.374372
2	201	12	30	8	0	257.0	1.278607
3	207	10	30	7	0	244.5	1.181159
4	241	10	33	9	0	278.0	1.153527
5	246	13	33	9	0	284.0	1.154472
6	219	19	41	8	0	331.5	1.513699
7	235	24	45	7	0	355.0	1.510638
8	244	21	46	9	0	371.5	1.522541
9	247	16	42	9	0	339.5	1.374494
10	237	15	46	7	0	342.5	1.445148
11	231	15	41	9	0	332.0	1.437229
12	205	14	47	9	0	363.0	1.770732
13	228	19	54	9	0	411.5	1.804825
14	247	18	71	12	1	538.5	2.180162
15	244	13	47	12	1	396.5	1.625000
16	243	18	63	16	1	528.5	2.174897
17	241	19	80	21	8	736.5	3.056017
18	217	18	84	25	8	790.5	3.642857
19	222	22	93	26	8	856.5	3.858108
20	129	26	118	26	7	992.0	7.689922
21	130	14	99	25	8	865.0	6.653846
22	130	20	123	27	8	1026.0	7.892308
23	128	19	130	29	8	1079.5	8.433594
24	136	23	99	21	6	829.0	6.095588
25	114	23	103	21	6	851.0	7.464912
26	92	21	91	17	3	717.0	7.793478
27	110	22	88	15	2	675.5	6.140909
28	85	18	48	7	0	359.5	4.229412

Time series models and autocorrelation

All of the charting methods consider here involve two-step approaches. In the first step, a procedure such as standard time-series modeling (Box and Jenkins 1994) is applied (MCRD, AD, and MCRAD). If the time-series model provides a good fit, the residuals are uncorrelated, approximately normally distributed, and provide useful inputs for charting.

In this section, the application of time series models to the data from the three organizations (three cases) is described. By examining the autocorrelation function (ACF) and partial autocorrelation function (PACF) for the demerits per host for the three cases (Table 7; Tables A1 and A2), it was determined that AR models offer an appropriate choice for all 3 cases. Figure 1 shows the ACF and PACF for organization #1 (case 1). The results for the 3 cases indicate that AR(1) models are good fit for the vulnerability data at hand. These choices are confirmed by studying the ACF and PACF plots of the AR(1) model residuals. In all three cases, the residuals show no evidence of autocorrelation and normal probability plots (not shown) indicate approximate normality (with the exception of the outlier associated with an assignable cause described above). The residuals for the case #1 data are shown in Figure 2. The corresponding ACF and PACF plots are excluded for cases #2 and #3 since they appear similar to those of case #1.

The relevance of AR(1) models for cyber vulnerability demerits is likely a general phenomenon because it relates to the carryover of the same vulnerabilities and associated demerits from period to period. With the complete data represented by Table 5, it was possible to identify vulnerabilities that appeared in one month but not the next month. Some of the missing vulnerabilities were presumably the result of the host being turned off or its firewall turned on. Yet, by assuming that all of the appearing and disappearing vulnerabilities were patched, an upper estimate of the average patching rate is obtained. For example, for organization #1 966 total vulnerabilities (not distinct) were identified over 28 months and a total of 265 instances in which vulnerabilities were present one month and absent the next. Therefore, the upper bound on the average monthly patching rate is determined as $100\% \times 265/966 = 27.4\%$. Table A4 in the Appendix contains the upper bound percentages of vulnerabilities that were patched from one month to the next for organization #1.

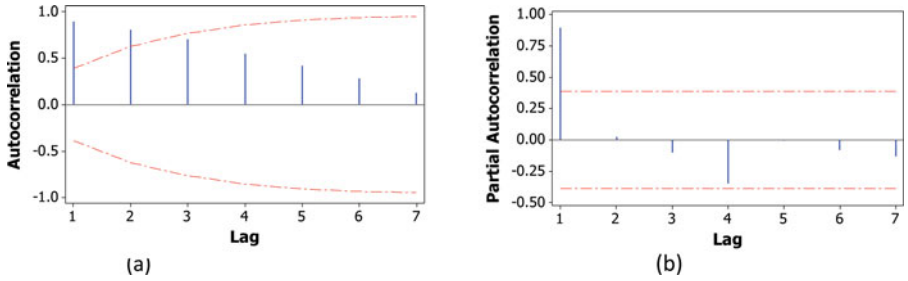

Figure 1. Case #1: Demerits per host (a) autocorrelation function and (b) partial autocorrelation function.

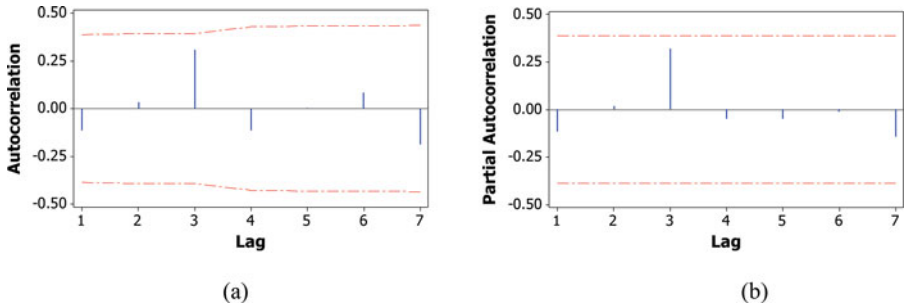

Figure 2. Case #1: AR(1) residuals (a) autocorrelation function and (b) partial autocorrelation function.

The AR model form was given in Eq. [10]. While the same AR(1) model form in Eq. [10] applied to all three cases, the degree of autocorrelation represented by the coefficient (φ) and the mean (μ) and standard deviation (σ) varied. Table 1 summarizes the coefficients for the three cases from AR(1) modeling. Therefore, organization #1 had tens of carry-over vulnerabilities from period to period (high φ) while organization #2 had many demerits per host (over 5) overall (high μ). Organization #3 had relatively lower carryover and good quality levels (low φ and low μ).

The application of the proposed methods

Upon consultation with the relevant system administrator and inspection of the data in Table 5, an assignable cause relating to an unusual shift in responsibility was identified. Through a re-organization, approximately 200 hosts were shifted to a different organization in period 20. Therefore, this detection can indeed be considered an assignable cause. The system administrator commented that no other occurrences during the 28 months seemed unusual.

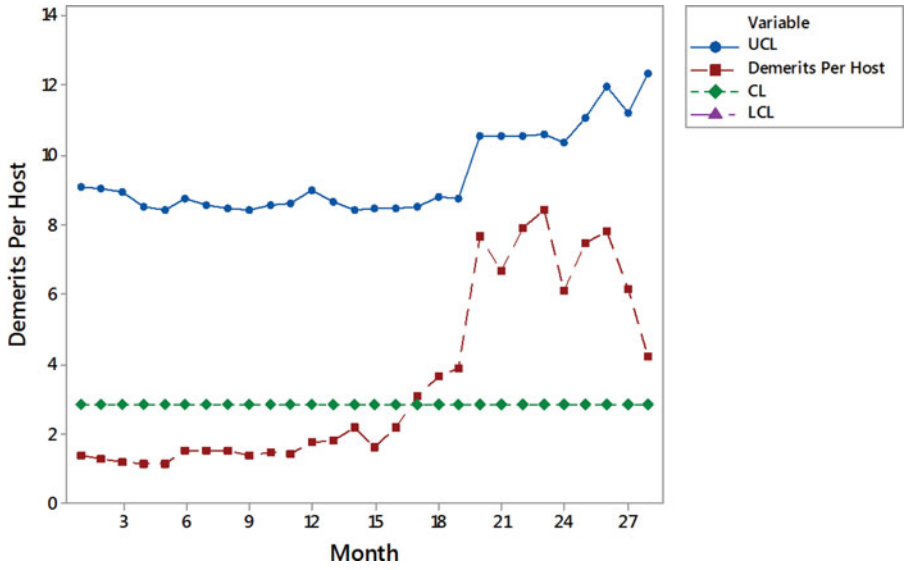

Figure 3. Adjusted demerit (AD) chart for the data from organization #1.

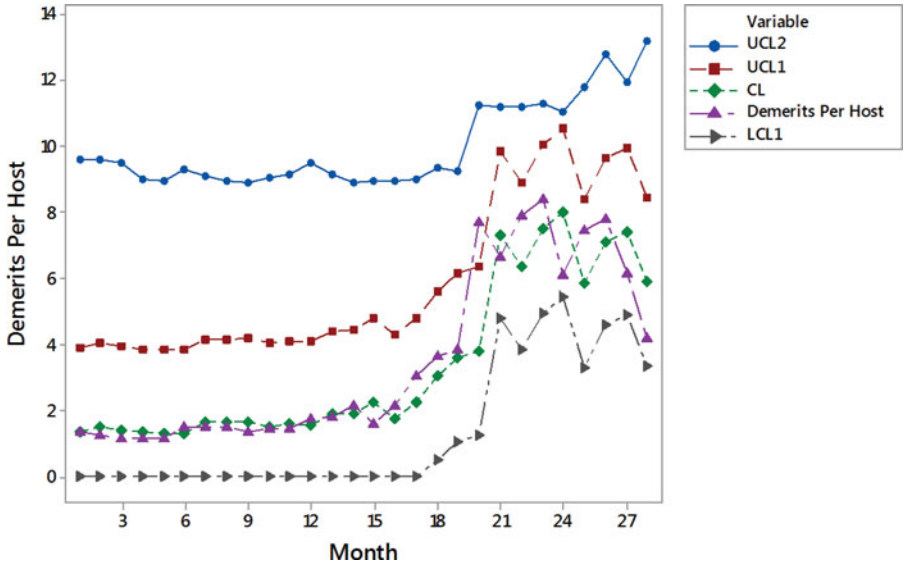

Figure 4. MCRAD chart for the data from organization #1 with $M_1 = 3.0$ and $M_2 = 22.79$.

In applying the adjusted demerit (AD) charting, the derived values of M for the three organizations and data sets are $M = 20.90$, 7.06, and 2.405, respectively. Note that the value 20.90 is much larger than 3.0 because of the relatively high degree of autocorrelation for organization #1. The adjusted demerit chart for the data from organization #1 in Table 7 is given in Figure 3. It is noteworthy, perhaps, that the adjusted demerit chart failed to identify the period 20 shift that both the MCD and MCRD charts (not shown) identified. It is conjectured that this failure highlights the relative strength of residual-based charts relating to immediate identification of shifts in the underlying process. Yet, the adjusted demerit chart has the potential advantage of being better able to identify causes in periods following a shift. The MCRAD chart combines the strengths of residual-based and adjusted demerit charts.

Based on data in Table 7 from organization #1, the value $M_2 = 22.9$ was used to achieve an approximate in-control average run length equal to 200. The derived MCRAD control chart is given in Figure 4. The chart generates the desired signal on subgroup 20 relating to the assignable shift of 200 hosts that were moved outside the relevant organization. The absence of a lower control limit from the adjusted demerit-related limits is due to the value $M_2 = 22.79$. Larger values of M_2 may generally be expected if the degree of autocorrelation is high. The value $\varphi = 0.920$ (Table 1) associated with

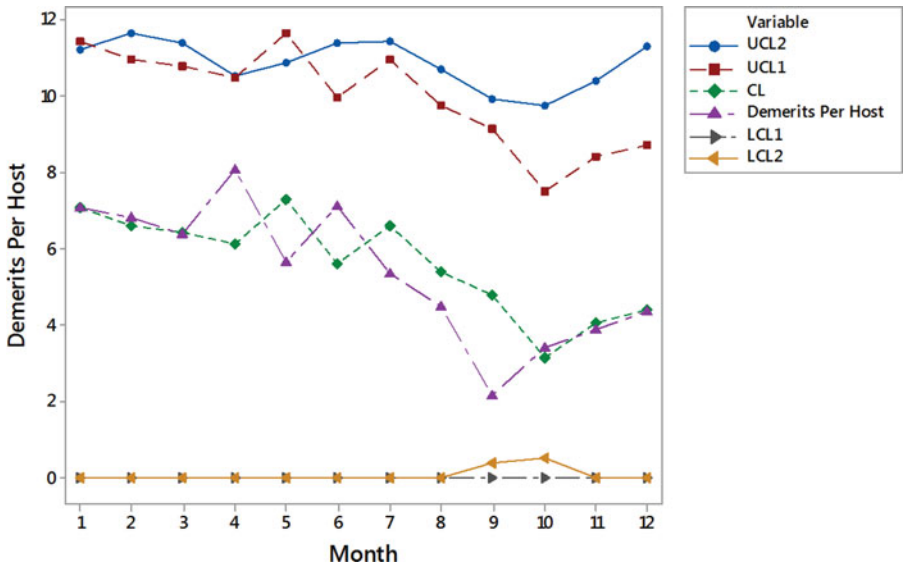

Figure 5. MCRAD chart for the data from organization #2 with $M_1 = 3.0$, $M_2 = 7.51$.

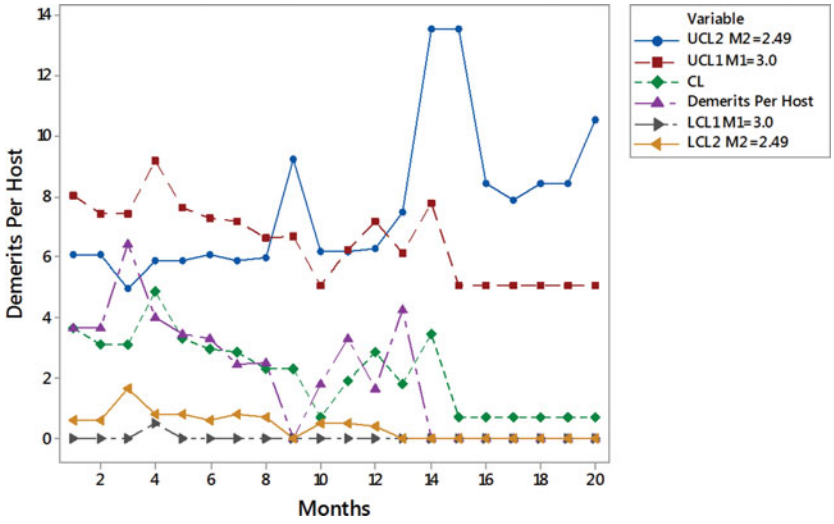

Figure 6. MCRAD chart for the data from organization #3 with $M_1 = 3.0$ and $M_2 = 2.49$.

the organization #1 dataset indicates a relatively high carryover of vulnerabilities from period to period.

The MCRAD charts for the data from organization #2 (Table A1) and organization #3 (Table A2) are given in Figures 5 and 6, respectively. Concerning organization #2, the chart in Figure 5 shows no out-of-control signals. For organization #3, the chart in Figure 6 shows an out-of-control signal in month three. This is as a result of the demerit per host for month 3 exceeding the residual-based chart control limit. From the perspective of the authors, this signal appears to be a false alarm.

Conclusions

This article addresses the important problem of monitoring cyber vulnerabilities using statistical process control (SPC) methods. The problem is important because of the high and growing threat level associated with cyber-attacks and the widespread use of personal computers and other hosts. A process is proposed to convert vulnerability data into demerits per host based on the common vulnerability scoring system (Mell et al. 2007). The application of standard time series models to cyber vulnerability data from three organizations is then described. The conclusion is that AR models with a single lag, i.e., AR(1) processes accurately model the three datasets and are possibly relevant for many other vulnerability modeling problems. The motivation for this choice relates to the carryover of the same unpatched vulnerabilities from one period to the next. Since the hosts are monitored instead of parts, one does not have new units each period.

Application of two residual-based methods taken from the literature is then investigated. The application involves moving centerline demerit (MCD) charting from Nembhard and Nembhard (2000) and a slight extension of the residual charts from Runger and Willemain (1995). The MCD chart offers the advantage of charting the relatively intuitive demerits per host instead of residuals which motivated the extension to create moving centerline residual-based demerit (MCRD) charts. The proposed adjusted demerit (AD) and hybrid moving centerline residual-based and adjusted demerit (MCRAD) charting methods are based on using simulation to determine the control limits. Average run length (ARL) comparisons were based on assumptions relevant to the three case studies. From this it is concluded that the proposed AD and MCRAD offer improved ARL performance compared with MCD and MCRD charts. The MCRAD charts in particular are recommended as a dashboard for monitoring cyber vulnerabilities. Also, the concepts of AD and MCRAD charts have applicability beyond cyber vulnerabilities and demerit charts to many charting situations involving autocorrelation.

About the authors

Anthony Afful-Dadzie is a lecturer at the University of Ghana Business School. He received his Ph.D. from The Ohio State University in Industrial & Systems Engineering and his MPhil from Cambridge. His research and teaching interests include quality engineering, cyber security, and economic models.

Theodore T. Allen is an associate professor of Integrated Systems Engineering at The Ohio State University. He is a fellow of the American Society for Quality and the author of over 50 peer reviewed publications including 2 textbooks. His research interests focus on optimization with parametric uncertainty including optimal experimental design and cyber security maintenance plan design.

Funding

This work was partially supported by National Science Foundation (NSF) grant #1409214.

References

Abedin, M., S. Nessa, E. Al-Shaer, and L. Khan. 2006. Vulnerability analysis for evaluating quality of protection of security policies. *Proceedings of the 2nd ACM Workshop on Quality of Protection*, Alexandria, Virginia, pp. 49–52.

Ahmed, M. S., E. Al-Shaer, and L. Khan. 2008. A novel quantitative approach for measuring network security. *Proceedings of the 27th IEEE INFOCOM 2008 Mini-Conference*, Phoenix, Arizona, pp. 1957–1965.

Alwan, L. C., and H. V. Roberts. 1988. Time series modeling for statistical process control. *Journal of Business and Economic Statistics* 6(1): 87–95.

Box, G.E.P., G.M. Jenkins, and Reinsel, G. C. 1994. *Time series analysis: forecasting and control*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall.

Cox, D. R. 1961. Prediction by exponentially weighted moving averages and related methods. *Journal of the Royal Statistical Society. Series B* 23(2): 414–422.

Dodge, H. F. 1928. A method of rating a manufactured product. *Bell System Technical Journal* 7: 350–368.

Dowdy, J. 2012. The cybersecurity threat to US growth and prosperity. In *Securing cyber space: a new domain for national security*, eds. N. Burns and J. Price, Washington, DC: Aspen Institute.

Enders, C. K. 2010. *Applied missing data analysis*, New York: Guildford Press.

Loredo, E. N., D. Jearkpaporn, and C. M. Borror. 2002. Model-based control chart for autoregressive and correlated data. *Quality and Reliability Engineering International* 18: 489–496.

Montgomery, D. C. 2012. *Introduction to statistical quality control*. 7th ed. Hoboken, NJ: Wiley.

Mell, P., K. Scarfone, and S. Romanosky. 2007. CVSS: A complete guide to the common vulnerability scoring system version 2.0, In *Forum of Incident Response and Security Teams*.

Montgomery, D. C., and Mastrangelo, C. M. 1991. Some statistical process control methods for autocorrelated data. *Journal of Quality Technology* 23 (3): 179–193.

Nembhard, D. A., and H.B. Nembhard. 2000. A demerit control chart for autocorrelated data. *Quality Engineering* 13 (2): 179–190.

Runger, G. C., and T. R. Willemain. 1995. Model-based and model independent control of autocorrelated processes. *Journal of Quality Technology* 27: 283–292.

Ponemon, L. 2010. Fifth Annual US Cost of Data Breach Study: Understanding Financial Impact, Customer Turnover and Preventive Solutions, Traverse City, MI: Ponemon Institute.

Appendix

This appendix includes additional data about vulnerabilities from our case studies. Table A1 describes the demerits for organization #2 and Table A2 describes the demerits for organization #3. Table A3 describes the score for the highest vulnerability on each host for the 36 hosts which had vulnerabilities (out of 498) for organization #1. Table A4 provides data on the worldwide known vulnerability counts and local patching percentages during the 28 month period for organization #1.

Table A1. Tabulation of the hosts scanned successfully and numbers of vulnerabilities of different levels of severity for organization #2. Also included are demerit sums based on the counts.

Month	Number of Hosts	Low	Medium	High	Critical	Demerits	Demerits Per Host
1	40	26	39	2	0	283.5	7.09
2	35	23	32	2	0	239.0	6.83
3	38	21	35	1	0	243.0	6.39
4	51	27	38	1	14	411.5	8.07
5	45	32	33	1	0	254.0	5.64
6	38	29	37	1	0	270.0	7.11
7	37	23	26	1	0	197.5	5.34
8	48	24	29	1	0	216.0	4.50
9	65	13	19	1	0	139.0	2.14
10	69	46	23	2	0	235.5	3.41
11	53	37	21	2	0	206.5	3.90
12	39	19	21	2	0	170.5	4.37

Table A2. Tabulation of the hosts scanned successfully and numbers of vulnerabilities of different levels of severity for organization #3. Also included are demerit sums based on the counts.

Month	Number of Hosts	Low	Medium	High	Critical	Demerits	Demerits Per Host
1	14	5	6	1	0	51.5	3.68
2	14	5	6	1	0	51.5	3.68
3	38	21	35	1	0	243.0	6.39
4	16	6	8	1	0	64.5	4.03
5	16	7	6	1	0	55.5	3.47
6	14	8	4	1	0	46.5	3.32
7	16	10	2	1	0	39.5	2.47
8	15	9	2	1	0	37.5	2.50
9	3	0	0	0	0	0.0	0.00
10	13	2	2	1	0	23.5	1.81
11	13	9	3	1	0	43.0	3.31
12	12	3	1	1	0	20.0	1.67
13	6	3	2	1	0	25.5	4.25
14	1	0	0	0	0	0.0	0.00
15	1	0	0	0	0	0.0	0.00
16	4	0	0	0	0	0.0	0.00
17	5	0	0	0	0	0.0	0.00
18	4	0	0	0	0	0.0	0.00
19	4	0	0	0	0	0.0	0.00
20	2	0	0	0	0	0.0	0.00

Table A3. Complete listing of CVSS scores for the most severe vulnerabilities on each host over the 28 months. Hosts with no vulnerabilities were omitted from the list. Empty cells are missing data.

	Host Number																																	
#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	5	2.6	4.3	5		5.1	6.2	5			7.5		7.5		7.6		5	5		7.5				5	9.3				5.1	2.6				6.4		
2	5		4.3	5		5	6.2	5.1							4.3		5	5		7.5				5	9.3					2.6				6.4		
3							6.2	5			7.5	7.5					5	5		7.5				5	9.3					2.6				6.4		
4			5.1					5			7.5			5			4.3	4.3		7.5				5	9.3			5.1		2.6				6.4	2.6	
5	5		5			5.1		5.1			7.5		7.5				4.3	4.3		7.5				5	6.2			5.1		2.6				6.4		
6	5							5.1		5.1							4.3	4.3		2.6				5	9.3			5.1		2.6				6.4		
7	5		5					5		5.1	7.5			5			4.3	4.3		7.5				5	9.3			5		2.6				6.4		
8	5							5			7.5	7.5					4.3	4.3		7.5				5	9.3			2.6	6.8	2.6				5		
9			4.3	5				5			7.5	5	7.5	5			4.3	4.3		7.5				5	9.3			6.4		2.6						
10			5	5		5.1				2.6		7.5					4.3	4.3		7.5				5	9.3			2.6	6.4	2.6						
11		5		5			7.5					7.5					4.3	4.3		7.5					9.3			2.6	6.4	2.6				2.6		
12	5		4.3	5							7.5	7.5	7.5	5			4.3	4.3		7.5					9.3			2.6	6.4	2.6				2.6		
13			10	5				5		2.6	7.5	7.5	7.5	5	4.3		4.3		7.5	7.5		7.5	7.5		9.3			2.6	6.4	2.6				2.6		
14			10	5				2.6			7.5		7.5				4.3	4.3	7.5	7.5		7.5			9.3			2.6	4.3	2.6				2.6		
15			10	5		5.1					7.5	7.5	7.5	7.5			4.3	4.3	7.5	5			7.5		9.3	7.5			7.5	2.6			5	5		
16			10	5				5.1			7.5	10	5	10			4.3	4.3	10	10		10	10						10	2.6			5	5		
17			10	5	5			5	2.6		7.5	10		5		4	4.3	4.3	10	10	5.1						10	2.6	10	2.6	7.5			5		
18			10		6.4				2.6		7.5						4.3	4.3		10					9.3		10	2.6	10		7.5	6.4		6.4		
19			10		6.4			6.4	7.5		7.5						4.3	4.3		10					9.3			2.6	7.5	2.6	7.5			6.4		
20																																				
21			10	7.5	7.5	6.4								10			4.3	4.3	10	10	5.1		10				10		7.5	2.6	7.5		5	6.4		
22			10	7.5	7.5			6.4						10			4.3	4.3	10	10	5.1		10				10		7.5		7.5		5	6.4		
23			4.3		6.4			6.4	6.4			10		7.5			4.3	4.3	10	10		10	10				10		5		7.5		5	2.6		
24			4.3		6.4			6.4	6.4			10		7.5			4.3		10	10		10	10				10		4.3				5	2.6		
25			4.3		6.4							10		7.5			4.3		4.3	10		10	4.3				4.3		4.3		7.5		5	2.6		
26			4.3		6.4			6.4																												
27																	4.3			5									4.3		7.5			2.6		
28			5	6.4	6.4			6.4						7.8			7.8			2.6								5	7.8		7.8			2.6		4.3

Table A4. Monthly cumulative number of worldwide vulnerabilities and the local percentage of new vulnerabilities that is patched each month, i.e., the number of vulnerabilities present in one month scan but missing in the next month scan divided by the total number of vulnerabilities in the first month.

Month	Cumulative Count of Distinct, Known Vulnerabilities Worldwide	Vulnerability Patching Percentage
1	27,503	3.23
2	28,080	12.90
3	28,583	41.38
4	28,847	47.83
5	29,140	18.75
6	29,487	23.81
7	29,728	12.90
8	29,972	16.67
9	30,433	45.71
10	30,792	4.55
11	31,200	35.48
12	31,593	36.36
13	31,873	11.76
14	32,235	50.00
15	32,505	66.67
16	32,875	13.33
17	33195	10.26
18	33516	34.62
19	34,003	27.66
20	34,381	20.00
21	34,628	10.81
22	34,940	13.24
23	35,286	38.89
24	35,601	9.09
25	35,879	66.04
26	36,195	48.00
27	36,584	18.18
28	37,082	37.04