Detecting network portscans through anomaly detection

Hyukjoon Kim Surrey Kim Michael A. Kouritzin Wei Sun

Random Knowledge Inc., Edmonton, Alberta, Canada

ABSTRACT

In this note, we consider the problem of detecting network portscans through the use of anomaly detection. First, we introduce some static tests for analyzing traffic rates. Then, we make use of two dynamic chi-square tests to detect anomalous packets. Further, we model network traffic as a marked point process and introduce a general portscan model. Simulation results for correct detects and false alarms are presented using this portscan model and the statistical tests.

Keywords: Network portscan, anomaly detection, z-test, chi-square test, marked point process

1. INTRODUCTION

Hackers often use a four step process for network intrusions: reconnaissance, exploitation, execution and clean up. Reconnaissance includes a method called portscanning, which is the assessment and identification of listening network services, used by the malicious Hacker to focus his/her attention on promising avenues. Scanning is performed by sending a sequence of probing packets to the target network ports and observing its responses. Inevitably, programs have exploitable weaknesses. Once a system's weakness is discovered, an intruder exploits this flaw to enter the system and executes his attack.

Since portscans is one of the most common precursors to network intrusion, detecting them is essential to ensure a maximum level of information security. Yet, the existing detection tools limit their focus on simple, predictable portscanning activities. Indeed, currently available intrusion detection systems are based on having a priori knowledge of attack methods and signatures. Such tools attempt to build upon an existing knowledge-base set of rules to determine whether potential attacks may be occurring.

In this note, we research detecting portscans from the viewpoint of anomaly detection. The idea is to use some statistical techniques to compute the departure of current network traffic from the normal traffic and classify these departure as anomalous. To this end we first introduce some static tests to analyze traffic rates and help detect non-stealthy portscans such as SYN floods. Then, we make use of two dynamic chi-square tests to detect slower portscans performed by a more patient hacker. The idea originates from Ref. 5, where the authors use a chi-square test with simple exponential smoothing to detect independent anomalous events in information systems. We find that this idea can be further developed to handle our portscan detection problem, where a smoothing vector is introduced to handle different traffic rates. We also replace the simple exponential smoothing with double exponential smoothing to handle trends in network traffic. Considering the dependency of chi-squares, we use windows with a threshold to help decrease false alarms. Moreover, we introduce the bootstrap method and randomization testing to handle the problems caused by small sample size and more even stealthier, slower portscans. If we further consider packet types, these tests would provide us with much more information about network anomaly.

The remainder of this note is organized as follows. In Sect. 2, we introduce the statistical tests for anomaly detection. In Sect. 3, we model network traffic as a marked point process and introduce a general portscan model. Moreover, we use a vertical portscan model to test the statistical methods for their performance through simulations.

Further author information: (Send correspondence to W.S.)
H.K.: E-mail: kieler@telusplanet.net, Telephone: (780)428-9218
S.K., M.A.K., W.S.: E-mail: {surrey.kim, michael.kouritzin, wei.sun}@randomknowledge.net, Telephone: (780)428-9218

Signal Processing, Sensor Fusion, and Target Recognition XIII, edited by Ivan Kadar,
Proceedings of SPIE Vol. 5429 (SPIE, Bellingham, WA, 2004)
0277-786X/04/$15 · doi: 10.1117/12.546127

2. STATIC AND DYNAMIC TESTS FOR ANOMALY DETECTION

We model packet traffic as a stochastic process with a state space S. For example, in relation to Ref. 4, S would be the four dimensional space consisting of the destination port, destination IP, source port, and source IP. Since the four-dimensional state space S is too large, we split it into the product of the destination space S_D and source space S_S. Notice that the traffic information from source port and source IP might be inaccurate because of many existing stealthy techniques such as idle scanning and source IP spoofing. In this note, we concentrate on detecting anomaly in S_D. To simplify notation, in the sequel, we set $S = S_D$ which consists of the destination port and destination IP.

2.1. Static Tests

Let S' be a subset of S. Here S' can be all the computers in a LAN connected with a port; all the 65535 ports associated with a server; or many points of port/IP combination. For concreteness, we let S' be all the computers connected with a particular port, e.g. port 23; however, the following static tests can be used for many other scenarios. Considering traffic rates, we may divide the computers into several classes such that all the computers in a class have similar levels of rate. Let C be a computer class and $t_0 = l_0\varepsilon$ for some constants $\varepsilon > 0$ and $l_0 \in I\!N$. Suppose that the packet traffic for a computer $IP_i \in C$ is a Poisson process with parameter γ_i. For a fixed $m_i \in I\!N$, we use Newton's method to solve the following equation

$$\sum_{j=0}^{m_i-1} \frac{(\gamma_i\tau_i)^j}{j!} e^{-\gamma_i\tau_i} = 0.99.$$

We denote by $N_i(0)$ the number of packets arriving at IP_i during the time interval $[t_0-\tau_i, t_0)$. If $N_i(0) > m-1$, we continue to observe the packet traffic for IP_i during the time intervals $[t_j, t_j+\tau_i)$, $t_j := t_0+j\varepsilon$, $0 \le j \le k_1-1$, where $k_1 \in I\!N$ is a suitable constant. Denote by $N_i(j+1)$ the number of packets arriving at IP_i during the time interval $[t_j, t_j+\tau_i)$, $0 \le j \le k_1-1$. If the cardinality of $\{0 \le j \le k_1-1 : N_i(j+1) > m_i-1\}$ is greater than some threshold ϱ_1, then we conclude that there are anomalous packets arriving at IP_i.

It is possible that $N_i(0) < \gamma_i\tau_i$ for each $IP_i \in C$ while a hacker still has an exploit for port 23. This is a horizontal scan, which is the common type of portscans at present. To detect this type of scans, we first select out all the computers satisfying $N_i(0) \ge \gamma_i\tau_i$. We denote this subclass of computers in C by C'. Suppose that the cardinality n of C' is not less than 30. We now use the central limit theorem and suggest the following z-test:

$$P\left(\frac{1}{\sqrt{q}}\sum_{i=1}^{q}\frac{N_i(0)-\gamma_i\tau_i}{\sqrt{\gamma_i\tau_i}} > 2.33\right) \sim 1\%.$$

We arrange C' in a decreasing order according to $\frac{N_i(0)-\gamma_i\tau_i}{\sqrt{\gamma_i\tau_i}}$ and do the above z-test $n-29$ times. For each $30 \le q \le n$, we use the first q computers to do the z-test. If we find that for some q,

$$\frac{1}{\sqrt{q}}\sum_{i=1}^{q}\frac{N_i(0)-\gamma_i\tau_i}{\sqrt{\gamma_i\tau_i}} > 2.33,$$

we continue to observe the traffic for $IP_i \in C'$, $1 \le i \le q$, during the time intervals $[t_j, t_j+\tau_i)$, $0 \le j \le k_2-1$, where $k_2 \in I\!N$ is a suitable constant. If the cardinality of

$$\left\{0 \le j \le k_2-1 : \frac{1}{\sqrt{q}}\sum_{i=1}^{q}\frac{N_i(j+1)-\gamma_i\tau_i}{\sqrt{\gamma_i\tau_i}} > 2.33\right\}$$

is greater than some threshold ϱ_2, then we conclude that there are anomalous packets arriving at IP_i.

Remark 1 We find that the following z-test sometimes provide us with additional information about network anomaly.

$$P\left(\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\frac{\frac{(N_i(0)-\gamma_i\tau_i)^2}{\gamma_i\tau_i}-1}{\sqrt{2+\frac{1}{\gamma_i\tau_i}}}\right| > 2.58\right) \sim 1\%.$$

2.2. Dynamic Tests

In this section, we introduce some dynamic tests for detecting network anomaly. For concreteness, we take detecting block scans as an example. Suppose that S' is an arbitrary subset of S and $T = N\varepsilon$ for some constants $\varepsilon > 0$ and $N \in \mathbb{N}$. We arrange S' in a finite sequence and use $|S'|$ to denote the number of points in S'. For each packet with destination port/IP $i \in S'$, we assign it a $|S'|$-dimensional vector with the i-th component being equal to 1 and the remaining components 0. For each time interval $[(k-1)\varepsilon, k\varepsilon)$ we denote by X_k the sum of all vectors associated with those packets visiting S' during the time interval $[(k-1)\varepsilon, k\varepsilon)$.

Let $\{X_k\}_{k=1}^N$ the vectors associated with a sequence of normal packets visiting S' during the time interval $[0, T)$. We use the following double exponential smoothing technique to get a new sequence $\{X_k^*\}_{k=1}^N$:

$$\begin{cases} X_k^* = \alpha \cdot X_k + (1-\alpha) \cdot (X_{k-1}^* + b_{k-1}), \\ b_k = \beta \cdot (X_k^* - X_{k-1}^*) + (1-\beta) \cdot b_{k-1}, \end{cases}$$

where $\alpha,\ \beta$ are two smoothing vectors with $0 \le \alpha_i,\ \beta_i \le 1$ for $1 \le i \le |S'|$ and $\cdot$ denotes componentwise multiplication. In this study, we initialize X_1^* to X_1 and $b_1 = \frac{X_{n_0} - X_1}{n_0}$ for some constant $n_0 \in \mathbb{N}$. Further, we use the following recursive formula to incrementally update $\{X_k^*\}_{k=1}^N$ after each time period ε:

$$X_k^{**} = \frac{1}{k}((k-1)X_{k-1}^{**} + X_k^*).$$

It is possible that some components of X_N^{**} are equal to 0, or equivalently, some points in S' are not visited in the training data. To avoid having a zero value for the denominators of equation (1) below when we apply the chi-square test, we replace 0 with 0.000001 for these components and denote the resulting vector by Y_N. Take another sequence of normal packets visiting S' during the time interval $[T, T + M\varepsilon)$, where $M \in \mathbb{N}$. We first use the above method to get a sequence of $|S'|$-dimensional vectors $\{Z_j\}_{j=1}^M$. Then, we use the following formula to compute chi-squares:

$$\chi_j^2 = \sum_{i=1}^{|S'|} \frac{(Z_j(i) - Y_N(i))^2}{Y_N(i)}, \quad 1 \le j \le M. \tag{1}$$

2.2.1. Dynamic test 1

If M is sufficiently large, we define the upper limit of chi-squares to be $\overline{\chi^2} + 4\sigma_{\chi^2}$, where

$$\overline{\chi^2} = \frac{1}{M}\sum_{j=1}^M \chi_j^2,$$

and

$$\sigma_{\chi^2} = \sqrt{\frac{1}{M-1}\sum_{j=1}^M (\chi_j^2 - \overline{\chi^2})^2}.$$

For a sequence of testing data, we first compute the associated chi-squares and then compare them with the above defined upper limit. If a chi-square is greater than the upper limit, we take it out and call it an outlier. To decrease false alarms, we further check the neighbors of each outlier. More precisely, we choose a suitable window size $s_1 \in \mathbb{N}$ and threshold $\varrho_1 \in \mathbb{N}$. For each outlier χ_i^2, we consider its left neighbors $\chi_i^2, \chi_{i-1}^2, \ldots, \chi_{i-s_1+1}^2$. If the number of outliers among these neighbors is greater than ϱ_1, we conclude that there are anomalous packets arriving.

Remark 2 If M is small, we suggest using the bootstrap method, as shown in Ref. 1, to find the confidence interval for the chi-squares. First, we define the empirical distribution function

$$F_M(x) = \frac{1}{M}\sum_{j=1}^M I(\chi_j^2 < x).$$

Then, we draw with replacement K new samples $\{(\chi^2_{1,k},\ldots,\chi^2_{M,k})\}_{k=1}^K$ from this empirical distribution. For $1 \le k \le K$, we define

$$\mu_k = \frac{1}{M}\sum_{j=1}^{M}\chi^2_{j,k},\ \sigma_k = \sqrt{\frac{1}{M-1}\sum_{j=1}^{M}(\chi^2_{j,k}-\mu_k)^2}.$$

We define $u_{99\%}$ to be the 99% upper quantile of $\{\frac{\mu_k-\overline{\chi^2}}{\sigma_k}\}_{k=1}^K$. The bootstrap improved confidence interval is then

$$[\overline{\chi^2}, \overline{\chi^2}+\sigma_{\chi^2}u_{99\%}].$$

2.2.2. Dynamic test 2

It is possible that χ^2_j is smaller than the upper limit or within the confidence interval (Remark 2) for each $1 \le j \le M$, while there are still anomalous packets. To handle this problem, we suggest further using the randomization test.

Let $L < M$ be a suitable window size and $\{\chi^2_j\}_{j=1}^J$ $(J > L)$ a sequence of chi-squares associated with the testing data. For each $0 \le i \le J-L$, we consider the subsequence of chi-squares $\{\chi^2_{1+i},\ldots,\chi^2_{L+i}\}$. We randomly select a subsequence of chi-squares of length L from $\{\chi^2_1,\ldots,\chi^2_M\}$ and denote it by $\{\chi^{2,n}_{1,i},\ldots,\chi^{2,n}_{L,i}\}$. We define

$$D_i = \frac{1}{L}\sum_{j=1}^{L}(\chi^2_{j+i}-\chi^{2,n}_{j,i}).$$

If $D_i > 0$ then we conduct the randomization test. The procedure is as follows. We first put the above $2L$ chi-squares together and randomly divide them into two groups of length L. Such a division is called an arrangement. Then, we compute the absolute value of difference between the two means of chi-squares associated with the two groups. Further, we randomly select 10,000 of the possible arrangements and compute the 10,000 differences. Finally, we determine the proportion p_i of these arrangements for which the corresponding absolute values of difference exceed D_i. If $p_i < 1\%$, we call the subsequence $d_i := \{\chi^2_{1+i},\ldots,\chi^2_{L+i}\}$ an outlier. Similar to the dynamic test 1, for each outlier d_i, we further check its neighbors. We choose a suitable window size $s_2 \in I\!N$ and threshold $\varrho_2 \in I\!N$. If the number of outliers among $\{d_i, d_{i-1},\ldots,d_{i-s_2+1}\}$ is greater than ϱ_2, we conclude that there are anomalous packets arriving.

We feel that the above dynamic chi-square tests will become more powerful if we make use of the packet type information. Following Ref. 3, we may develop a script to scan each tcpdump data file and extract the packet type information about the network traffic. For each TCP connection, we check

Class I (direction): 1. out-going, 2. in-coming, or 3. inter-LAN.

Class II (3-way handshake): 1. successful connection, 2. rejected connection, 3. attempted but not established connection, or 4. unwanted SYN acknowledgment received.

For a successful connection, we further check

Class A (how connection is terminated): 1. normal, 2. abort, or 3. half closed.

Class B (duration): 1. long duration, 2. medium duration, or 3. short duration.

Class C (flags): 1. normal, or 2. one of the recorded connection/termination errors.

Class D (ports): 1. well-known service, or 2. user application.

There are $3\times3+3\times3\times3\times2\times2 = 117$ packet types all together. We may define S to be the state space consisting of these 117 points and conduct the above chi-square tests.

3. MODELS AND SIMULATION RESULTS

3.1. General Model

We model packet traffic as a marked point process with marks in a space S. First, we use γ to specify the likelihood of different packets under normal traffic, giving a benchmark from which we can judge the anomalousness of a sequence of observed packets. Then, letting $Y_1(A,t)$ denote the number of packets having marks with values in $A \subset S$ that have been observed up to tome t under normal traffic conditions, we can write

$$Y_1(A,t) = \int_{A\times[0,\infty)\times[0,t]} 1_{[0,\gamma(u)]}(v)\xi_1(du \times dv \times ds),$$

where ξ_1 is a Poisson random measure on $S \times [0,\infty) \times [0,\infty)$ having mean measure $\nu \times l \times l$ with ν the counting measure on S and l the Lebesgue measure on the half real line.

We use the following counting-measure process to keep track of the intrusive port scan packets:

$$\eta(A,t) = \int_{A\times[0,\infty)\times[0,t]} 1_{[0,\lambda(u,\eta_{s-})]}(v)\xi_2(du \times dv \times ds),$$

where η_0 is some random counting measure with distribution μ_0 satisfying $E(\eta_0(S)) < \infty$ and ξ_2 is another Poisson random measure independent of ξ_1. We model the activity of a portscan as a randomly initiated and growing cluster counting the unfriendly network probes. There is an intensity $\lambda(u,0) = \gamma_0(u)$ that gives the rate at which a new cluster (i.e. portscan) is initiated by a packet with mark u. Once initiated, the scan adds another point u to the cluster η with intensity $\lambda(u,\eta)$.

3.2. Simulation Results

We let S be the state space consisting of the port class and IP class and use a matrix to denote S. For simulation tests performed, we group 299 ports into 5 classes and group 304 computers into 6 IP classes. Legitimate traffic rates γ and the maximum number R of malicious packets for vertical scans are shown below.

$$\gamma = \begin{pmatrix} 5.334500 & 2.930000 & 0.505000 & 1.010000 & 0.100000 & 0.340100 \\ 0.200000 & 1.301000 & 0.505000 & 0.100000 & 0.101000 & 1.930000 \\ 0.100000 & 0.701000 & 1.183320 & 0.401000 & 0.210000 & 0.200000 \\ 0.100000 & 0.100000 & 1.163000 & 0.100000 & 1.230000 & 0.100000 \\ 0.100000 & 0.100000 & 0.100000 & 3.236000 & 1.500000 & 0.640100 \end{pmatrix},$$

$$R = \begin{pmatrix} 15 & 15 & 15 & 15 & 15 & 15 \\ 31 & 31 & 31 & 31 & 31 & 31 \\ 63 & 63 & 63 & 63 & 63 & 63 \\ 63 & 63 & 63 & 63 & 63 & 63 \\ 127 & 127 & 127 & 127 & 127 & 127 \end{pmatrix}.$$

We let the initial distribution μ_0 be the uniform distribution on S. At the beginning, the scanner uses μ_0 to randomly select a point $u_{i_0 j_0} \in S$ and send his/her first packet. Once $u_{i_0 j_0}$ is selected, the scanner begins to use the following mechanism to scan the 299 ports for his/her target host j_0.

$$\lambda(u_{ij_0},\eta) = \begin{cases} 0, & \text{if } R_{ij_0} - |\eta_{ij_0}| = 0, \\ c, & \text{otherwise,} \end{cases}$$

where c is a constant.

3.2.1. Static tests

We use window sizes

$$\tau = \begin{pmatrix} 1.01756 & 1.19708 & 1.6302 & 1.26644 & 4.36045 & 1.28211 \\ 2.18023 & 1.37224 & 1.6302 & 4.36045 & 4.31728 & 1.20736 \\ 4.36045 & 1.17439 & 1.08095 & 1.08739 & 2.07641 & 2.18023 \\ 4.36045 & 4.36045 & 1.09983 & 4.36045 & 1.03992 & 4.36045 \\ 4.36045 & 4.36045 & 4.36045 & 1.08389 & 1.19019 & 1.28613 \end{pmatrix}$$

and packet count thresholds

$$m = \begin{pmatrix} 12 & 9 & 4 & 5 & 3 & 3 \\ 3 & 6 & 4 & 3 & 3 & 7 \\ 3 & 4 & 5 & 5 & 3 & 3 \\ 3 & 3 & 5 & 3 & 5 & 3 \\ 3 & 3 & 3 & 9 & 6 & 4 \end{pmatrix}.$$

Figures 1-4 are the true positives (true detects) and false positives (false alarms) results for 100 simulation runs. For each simulation run we have 10 malicious vertical scans to detect, starting at random times.

3.2.2. Dynamic tests

We compute X^{**} offline using $N = 20000$ and test both dynamic tests using $M = 10000$. We use smoothing vectors

$$\alpha = 1 - 10^{-\frac{3}{2}\gamma} = \begin{pmatrix} 1.000000 & 0.999960 & 0.825217 & 0.969451 & 0.292054 & 0.691077 \\ 0.498813 & 0.988819 & 0.825217 & 0.292054 & 0.294495 & 0.998726 \\ 0.292054 & 0.911182 & 0.983211 & 0.749677 & 0.515828 & 0.498813 \\ 0.292054 & 0.292054 & 0.981991 & 0.292054 & 0.985711 & 0.292054 \\ 0.292054 & 0.292054 & 0.292054 & 0.999986 & 0.994377 & 0.890390 \end{pmatrix}$$

and

$$\beta = 1 - 10^{-\frac{3}{4}\gamma} = \begin{pmatrix} 0.999900 & 0.993654 & 0.581929 & 0.825217 & 0.158605 & 0.444192 \\ 0.292054 & 0.894257 & 0.581929 & 0.158605 & 0.160057 & 0.964314 \\ 0.158605 & 0.701977 & 0.870428 & 0.499678 & 0.304175 & 0.292054 \\ 0.158605 & 0.158605 & 0.865801 & 0.158605 & 0.880464 & 0.158605 \\ 0.158605 & 0.158605 & 0.158605 & 0.996259 & 0.925011 & 0.668926 \end{pmatrix}.$$

For the dynamic test 2 we also use the window size $L = 50$. Figures 5-8 are the true positives (true detects) and false positives (false alarms) results for 100 simulation runs. For each simulation run we have 10 malicious vertical scans to detect, starting at random times.

REFERENCES

1. W. Härdle, and L. Simar, *Applied Multivariate Statistical Analysis*, http://www.quantlet.com/mdstat/scripts/mva/htmlbook/mvahtml.html, 2003.
2. S. Jha, M.A. Kouritzin, and T.G. Kurtz, "Detecting stealthy port scans, a filtering approach," *Preprint*, 2003.
3. W. Lee, and S. Stolfo, "Data Mining Approaches for Intrusion Detection," in *Proceedings of the Seventh USENIX Security Symposium (SECURITY'98)*, San Antonio, TX, 1998.
4. S. Staniford, J. Hoagland, and J. McAlerney: "Practical Automated Detection of Stealthy Portscans," *J. Comput. Sec.*, **10**, pp. 105-136, 2002.
5. N. Ye, and Q. Chen, "An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems," *Qual. Reliab. Engng. Int.*, **17**, pp. 105–112, 2001.

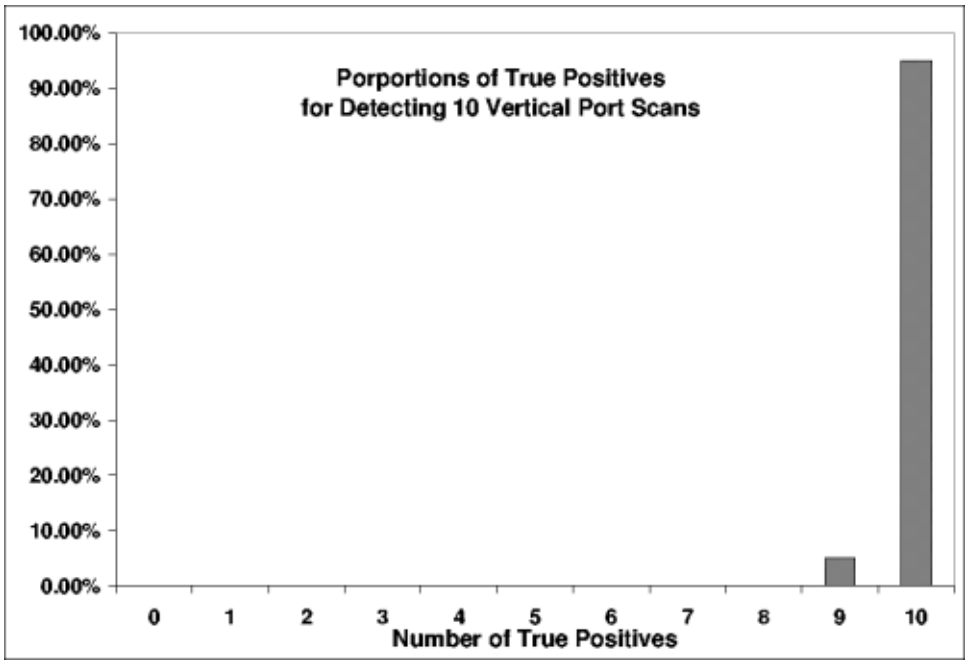

Figure 1. True detects for the static Poisson test

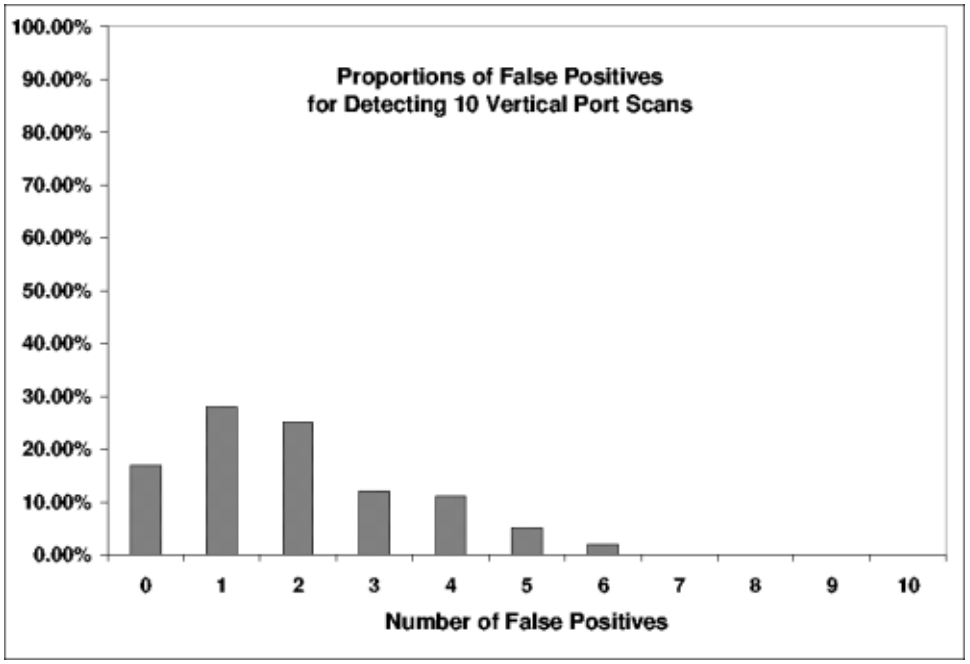

Figure 2. False alarms for the static Poisson test

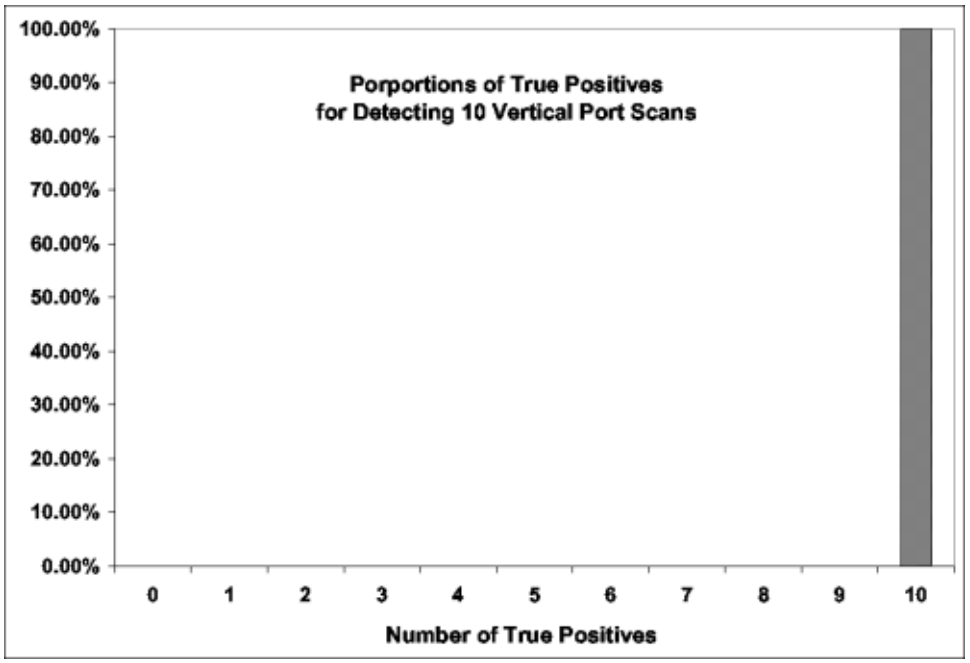

Figure 3. True detects for the static z-test

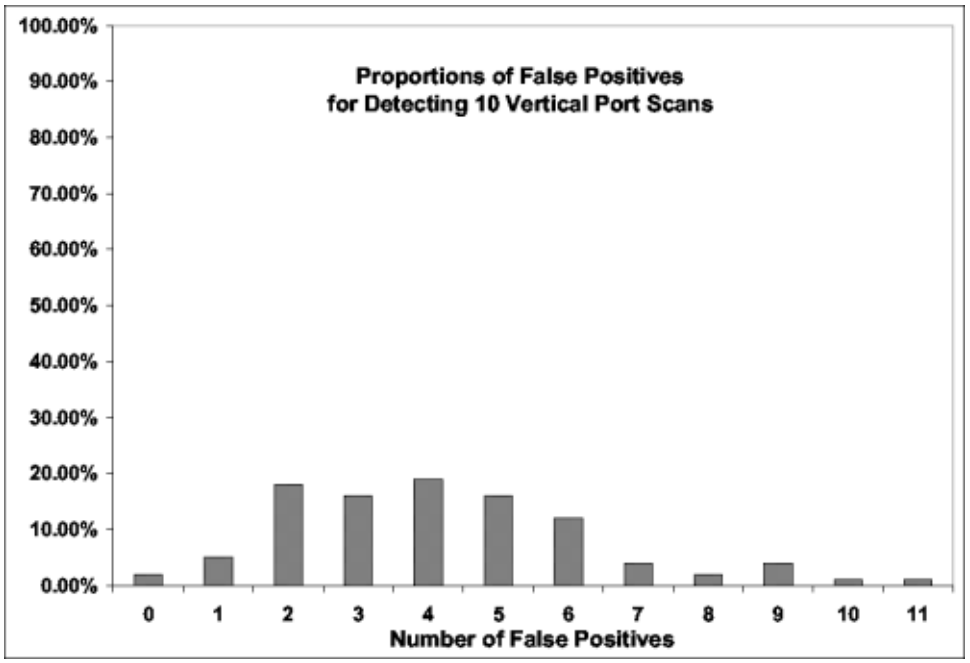

Figure 4. False alarms for the static z-test

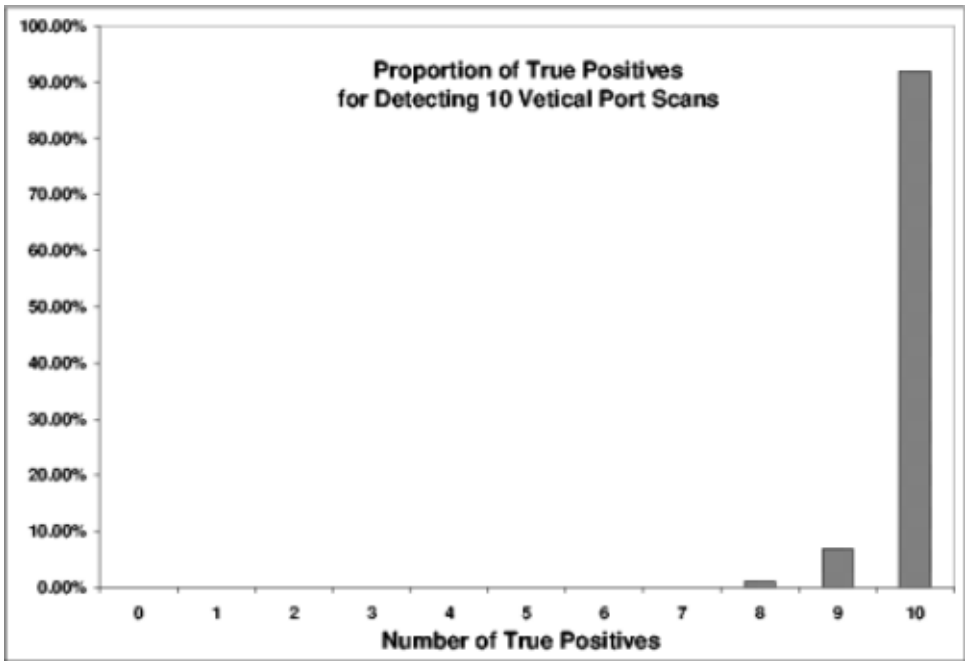

Figure 5. True detects for the dynamic test 1

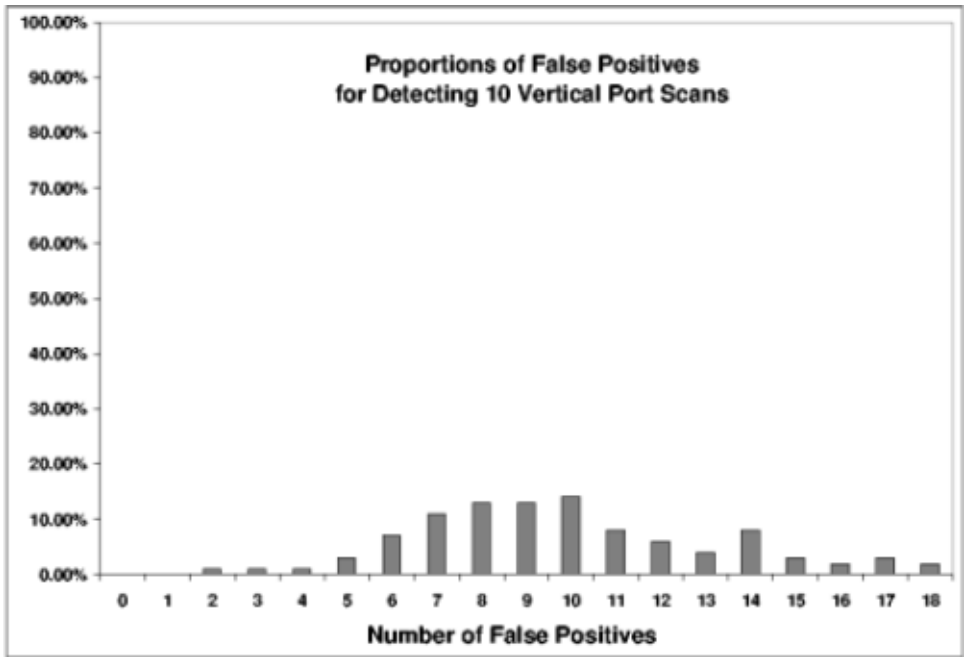

Figure 6. False alarms for the dynamic test 1

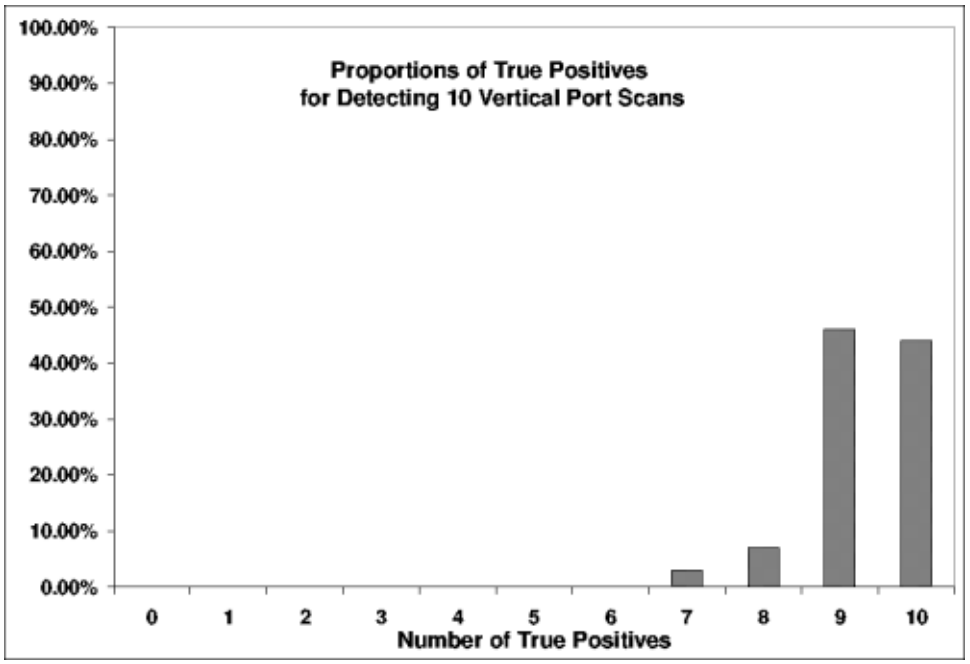

Figure 7. True detects for the dynamic test 2

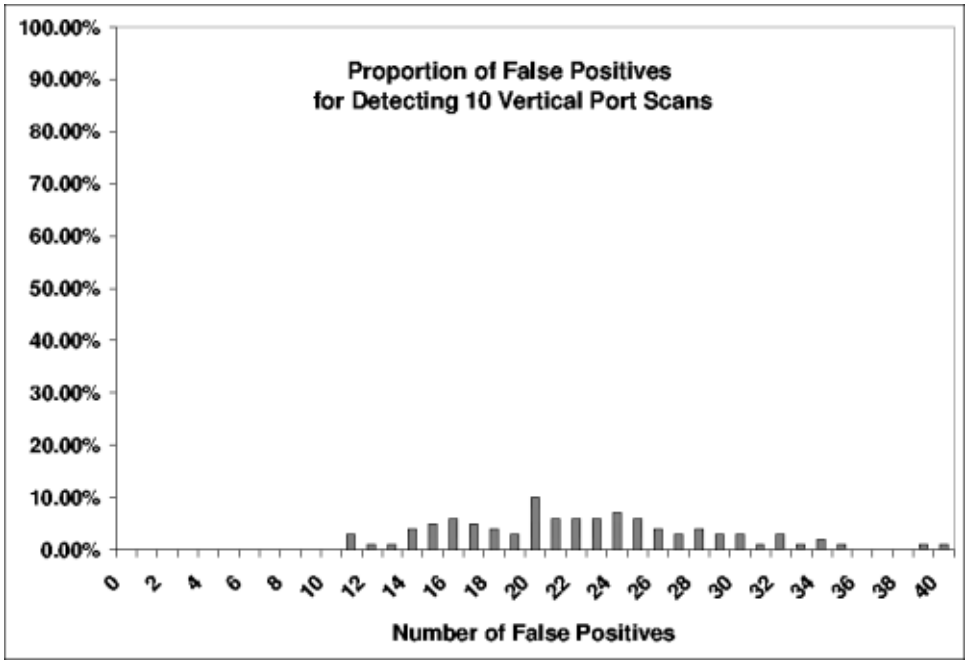

Figure 8. False alarms for the dynamic test 2