Slow-Paced Persistent Network Attacks Analysis and Detection Using Spectrum Analysis

Li Ming Chen, Shun-Wen Hsiao, Meng Chang Chen, and Wanjiun Liao,

***Abstract*—A slow-paced persistent attack, such as slow worm or bot, can bewilder the detection system by slowing down their attack. Detecting such attacks based on traditional anomaly detection techniques may yield high false alarm rates. In this paper, we frame our problem as detecting slow-paced persistent attacks from a time series obtained from network trace. We focus on time series spectrum analysis to identify peculiar spectral patterns that may represent the occurrence of a persistent activity in the time domain. We propose a method to adaptively detect slow-paced persistent attacks in a time series and evaluate the proposed method by conducting experiments using both synthesized traffic and real-world traffic. The results show that the proposed method is capable of detecting slow-paced persistent attacks even in a noisy environment mixed with legitimate traffic.**

***Index Terms*—Network security, persistent activity, slow-paced attack, spectrum analysis, time series.**

I. Introduction

A persistent network activity actuates periodically for a certain period, and leaves a footprint of a series of data packets. Those activities include many daily normal operations, such as network management operations and remote database synchronization, as well as malicious attacks. Traditional attack detection [1], [2] usually assumes that the malicious attack (e.g., fast-spreading worms) generates more network traffic than legitimate events (i.e., normal non-attack network activities), and thus focuses on monitoring significant volume deviations from the normal traffic. In this paper, we use the spectrum analysis technique to analyze network trace and raise alert of potential slow-paced persistent network attacks (called *slow-paced attack* for short), which does not expose significant behavioral deviations that the traditional detection approaches cannot detect or may falsely misjudge.

Slow-paced attacks are programmed to perform a sequence of network events, such as port scan or "phone home" of a recruited bot, to jointly carry out the designated task. In order to bewilder the detection system, slow-paced attack increases the period between two events so called *temporally sparse*. The difficulty of detection problem resides in distinguishing such slow-paced attack events from increasingly complex and enormous Internet traffic. Typically, a slow-paced attack may consist of probe, compromise and propagate, command and control, and perform malicious tasks. Some of them (e.g., compromise and propagate) generate transient activities that will not affect the overall spectral pattern, while some activities (e.g., perform malicious tasks) may be bursty which can be easily observed in the time domain. In this paper we only focus on detecting those persistent activities generated by malware, instead of investigating all other malware behaviors.

Specifically, we frame our problem as detecting slow-paced persistent attacks from a time series obtained from network traffic. We focus on time series spectrum analysis to identify peculiar spectral patterns that may represent the occurrence of a persistent activity in the time domain. We argue that although the correlation among these slow-paced events is temporally weak and difficult to discover in the time domain, their regularity is preserved in the corresponding time series and may be observed in the frequency domain. Further, operating on a time series is time efficient and resource conserving, which allows us to analyze long-term network traffic. Compared with frequency-based approaches adopted in other research fields, such as harmonic analysis in signal processing or heart rate analysis in medicine, they usually have a clear target and single data source in a less noisy observation environment. In the slow-paced attack detection, however, we must deal with the concerns that (1) attack is not known beforehand, and (2) each monitoring environment has their own time-varying traffic patterns. The unknown target and complicated environment make the detection process even more difficult.

Based on the mathematical model proposed in this paper, we examine the spectral properties of both legitimate and slow-paced attack time series. We find that human triggered legitimate events tend to generate a flat spectrum and the average spectrum magnitude decreases when the duration of the observation is increased. In contrast, the persistent attack events generate a sequence of peaks in the frequency domain, which are not affected by the length of the time series. Based on the observation, we propose a method to adaptively detect slow-paced attack in a time series. We evaluate the proposed method by conducting experiments using both synthesized traffic and real-world traffic. The results show that the proposed method is capable of detecting slow-paced attacks even in a noisy environment mixed with legitimate traffic. The contributions of this work can be summarized in the following points.

- This work analyzes the characteristics of slow-paced attacks and legitimate traffic in the frequency domain.

Manuscript received Month Date, 2014; revised Month Date, 2014; accepted Month Date, 2014. Date of publication Month Date, Year.

L. M. Chen was with National Taiwan University, Taipei, 106 Taiwan. He is now with the Institute of Information Science, Academia Sinica, Taipei, 115 Taiwan (e-mail: leeming@iis.sinica.edu.tw).

S.-W. Hsiao is with the Research Center for IT Innovation, Academia Sinica, Taipei, 115 Taiwan (e-mail: hsiaom@iis.sinica.edu.tw).

M. C. Chen is with the Institute of Information Science, Academia Sinica, Taipei, 115 Taiwan (e-mail: mcc@iis.sinica.edu.tw).

W. Liao is with the Department of Electrical Engineering, National Taiwan University, Taipei, 106 Taiwan (e-mail: wjliao@cc.ee.ntu.edu.tw).

- The proposed frequency-based method can effectively detect slow-paced attacks, which are difficult to be observed in the time domain.
- The proposed method is evaluated using both synthesized dataset and real-world traffic traces, and the advantages and limitations of the proposed method are fully discussed.

The remainder of this paper is organized as follows. Section II discusses related work and Section III provides an example of a slow-paced attack and the background of spectrum analysis. In Section IV, we examine the spectral properties of both legitimate and persistent attack time series. We describe the problem and present our attack detection method in Section V. We evaluate the proposed method based on synthesized traffic and real-world traffic in Section VI. Section VII discusses the limitation of the proposed method and concludes the paper.

II. Related Work

In literature, many detection approaches also focus on investigating stealthy attacks. Staniford *et al.* [3] propose a stealthy portscan detection mechanism based on spatial contact properties. Our approach focuses on time series spectrum analysis which requires no prior information about contact peers and connection status. In [4], Sekar *et al.* propose a multi-resolution approach for worm detection, including slow worms. Their approach needs to track the contact peers of individual hosts for different destination ports in a large time window and may incur false positives (FPs) when facing a crowded network with complex traffic patterns (e.g., P2P). For botnet detection, Giroire *et al.* [5] apply a similar concept to identify stealthy C&C (command and control) traffic and detect bots in a monitored network. Similarly, they need to measure the temporal persistence for individual destination atoms on each end-host. However, nowadays, bot master may use fast-flux techniques to hide such persistent communications. In [6], Akujobi *et al.* propose an integrated detection approach for both fast and slow scanning worms. Their approach does not consider the impacts of legitimate traffic. Gao *et al.* [7] especially focus on online detection of stealthy spreaders in a high-speed network. This approach may have the same limitation as to the multi-resolution approach in [3] since it also needs to track distinct communication pairs in a predefined time interval and may result in FPs when dealing with a slow-paced attack. In [8], an entropy-based detection scheme for varying scan rate (VSR) worm is introduced. VSR worm detection is beyond the scope that our frequency-based detection method can deal with, because the regularity of attack behavior may no longer exist. However, our approach is general enough to detect other recurring phenomenon in network security, such as bots phone home, which cannot be detected by an attack-target distribution-based approach. In [9], Chen *et al.* propose a scalable forensics mechanism to identify the origin of a stealthy self-propagating attack. They focus on spatial and temporal contact activities to reduce the traffic volume and back track the attack, while our work focus on analyzing the spectral properties of the attack.

Some literatures also focus on the frequency domain analysis for some security issues. In [10], Barford *et al.* utilize wavelet filters to analyze the characteristics of ambient and anomalous traffic. Zhou *et al.* [11] use Fourier transform to mine frequency patterns from various network attacks, such DoS, portsweep, and dictionary attacks. Our approach can further recognize the suspicious patterns and identify the potential recurring rates. In [12], Kim *et al.* inspect the frequency characteristic of scanning worms and design a real-time detection algorithm to filter out the attacks. Their approach, however, considers the detection algorithm and is suitable in a well-managed operation environment in that some required parameters are available, such as the update margin. Our approach concentrates on identifying the persistent activities generated by a single or multiple slow-paced attacks.

III. Background

This section provides background knowledge of this work. We first give an example of slow-paced persistent attack. We then introduce the basic spectrum analysis theory, the discrete Fourier transform (DFT), including some practical issues when using DFT on time series spectrum analysis.

A. Stealthy and Slow-Paced Attack

Conficker [13], [14] is a computer worm and also a botnet discovered in November 2008. It has up to five variants and it mainly exploits vulnerability in the NetBIOS server service on Windows computers. In the execution flow of Conficker mentioned in [14], after completing its initial operation, Conficker enters an infinite loop to perform its preprogrammed tasks. Conficker uses this mechanism to gather or exchange information from the master servers that are controlled by the attacker. For example, Conficker A, one of the Conficker's variant, generates a list of domain names and attempts an HTTP connection to try to download the most recent executable code from each domain. This task is repeated every three hours to make itself updated and avoid being detected by signature matching. When a computer was infected by Conficker, such a long sleeping period makes Conficker stealthy that its communications are submerged within tremendous normal Internet traffic. Note that this persistent activity is only part of all the activities of Conficker. Similar to other malware, Conficker is also designed to propagate and cause damages to the system or network.

B. Discrete Fourier Transform

In this work, we adopt DFT as the spectrum analysis tool for understanding the frequency content of a connection attempt time series. The DFT maps a finite-length discrete time-domain signal into an equal-length sequence of frequency-domain samples, which is also called the spectrum. Given a time-domain signal x with length L (i.e., $x = \{x_t | t = 0, 1, \dots, L-1\}$), the converted spectrum X which consists of a sequence of samples X_f at frequency f can be computed as (1), where $\mathcal{F}()_f$ is a symbol used to denote the transform, $0 \le f \le L-1$, and the coefficient $1/L$ is a conventional normalization factor [15].

$$X_f = \mathcal{F}(x)_f = \tfrac{1}{L}\textstyle\sum_{t=0}^{L-1} x_t e^{-j2\pi ft/L}, \qquad (1)$$

We usually use shifted DFT to represent a spectrum in both positive and negative frequencies. Since these two portions of the spectrum are symmetrical at frequency 0, the spectrum analysis in this work will only consider the spectrum of non-negative frequencies. DFT could be computed in $O(L \log L)$ by using the fast Fourier transform (FFT) algorithm, such that the spectrum analysis is efficient. DFT has the property of linearity. For an aggregate signal $x = y + z$, the converted spectrum can be represented as $X = Y + Z$, where Y and Z are the spectrums of signals y and z respectively. However, the output of DFT (e.g., X_f) is in general a complex number. For the purpose of comparison, we usually take the absolute value of a complex number (denoted as $\mid X_f \mid$) to depict the magnitude at a frequency in a spectrum.

C. Practical Issues of Time Series Spectrum Analysis

In the below, we discuss some practical issues such as accuracy and performance when using DFT on time series spectrum analysis.

According to the sampling theorem, signal frequencies higher than the Nyquist frequency, which is defined as half the sampling frequency of the signal, will fold back into lower frequencies; that causes the *aliasing* problem. Fortunately, in this work we mainly focus on detecting low rate events, such that we are not worried about losing high frequency events. Even when high frequency events are lost, we are still aware of their impact due to the aliasing effect.

Leakage is a phenomenon where the energy at a frequency in a spectrum leaks to other adjacent frequencies. The reason is DFT usually requires that the input signal be a nonzero distribution of compact support and assumes that the support area can represent the periodicity of the signal [16]. However, as the periodicity of the input signal is unknown in advance, leakage problem is inevitable, and in this work, the proposed detection method will deal with the leakage problem.

A final and general concern about the accuracy involves the *picket fence effect* (a.k.a. *resolution bias error*), which is due to the discrete spectrum nature of DFT. If a frequency does not match the discrete points of a spectrum, its magnitude will be falsely measured.

For the performance issue, the FFT algorithm runs in linearithmic time according to the length of the time series. Practically, we create a time series with a shorter length while cover the same duration of the data by using a larger time bin. However, a binned time series has a coarse-grained temporal resolution, such that it also loses the capability to observe rapidly changing events. On the other hand, a binned time series may tolerate some frequency fluctuations when a regular event suffer disturbance due to various noises, such as variable transmission delay of packet. The impact of using time bin in time series spectrum analysis will be examined and discussed in our experiments (see Section VI).

IV. Spectral Property Analysis

In this section, we first use mathematical models to examine the spectral properties of both legitimate and persistent attack time series. These preliminary observations give us hints to explore the problem of slow-paced persistent network activities detection. We define an event as a *connection attempt* of one host communicates with another host, regardless of success or failure. Each event is timestamped with the connection start time. We construct a time series to record the *connection statistics* (i.e., total number of outbound connection attempts of all hosts in the monitored network domain per time interval) through the monitoring period. We now present the time series model for generating persistent attack events and legitimate events and examine their spectral properties.

A. Spectral Properties of Persistent Attack Events

1) Single Attacker: In this subsection, we discuss the case of single attacker who generates persistent attack events. Based on our definition of time series, the persistent events may be traced in the time series with identical interval. Therefore, we borrow the concept of impulse train [15] to model a persistent attack with attack period T that incurs an attack event for each time period T as a T-periodic time series $z = \{z_t\}$ of length L as follows:

$$z_t = \textstyle\sum_{k=0}^{R-1} \delta(t - kT), \tag{2}$$

where R is number of impulses during the observation (e.g., $R = \lfloor L/T \rfloor$) and δ is a delta function which returns 1 when the inner equation is zero, otherwise returns 0. We use δ to indicate a connection attempt triggered by an infected host at a specific time. In the ideal case, the spectrum of an infinite impulse train with a period T forms an impulse train with a frequency interval $1/T$ (for every two adjacent impulses in the spectrum).

When L is a multiple of T (i.e., $\mathrm{mod}(L, T) = 0$), the time series records the repeated attack activities during the observation. Based on DFT, we can know that the converted spectrum Z_f is

$$\begin{aligned} Z_f &= \tfrac{1}{L} \textstyle\sum_{t=0}^{L-1} (\sum_{k=0}^{L/T-1} \delta(t - kT)) e^{-j2\pi ft/L} \\ &= \tfrac{1}{L} \textstyle\sum_{k=0}^{L/T-1} \sum_{t=0}^{L-1} \delta(t - kT) e^{-j2\pi ft/L} \\ &= \tfrac{1}{L} \textstyle\sum_{k=0}^{L/T-1} e^{-j2\pi fkT/L}. \end{aligned} \tag{3}$$

Equation (3) is derived due to the shifting property of the delta function. If f is a multiple of L/T, Z_f is $1/T$ since $e^{j2\pi n} = 1$ where $n \in \mathbb{Z}$; otherwise, Z_f is 0 due to the summation property of the root of unity. Therefore, the converted spectrum will have T impulses with frequency interval L/T and magnitude $1/T$, which forms an impulse train in the frequency domain.

However, when L is not a multiple of T (i.e., $\bmod\ (L, T) \neq 0$), the spectrum spreads out to a series of needle-like structure due to the *leakage* problem, while the peaks of these needles still preserve certain periodicity. In this case, using a longer time series (i.e., a large L) can help increase the resolution in the frequency domain and reduce the effect of leakage.

In the below, we use some examples to demonstrate the spectral properties of a single attacker. We use $\mathrm{PA}(T, L, b)$ to represent a T-periodic attack time series with length L measured by bin size b, where PA is the abbreviation of persistent activity.

Fig. 1(a) shows the spectrum of $\mathrm{PA}(10, 100, 1)$. We can see that in this case the impulses in the spectrum have

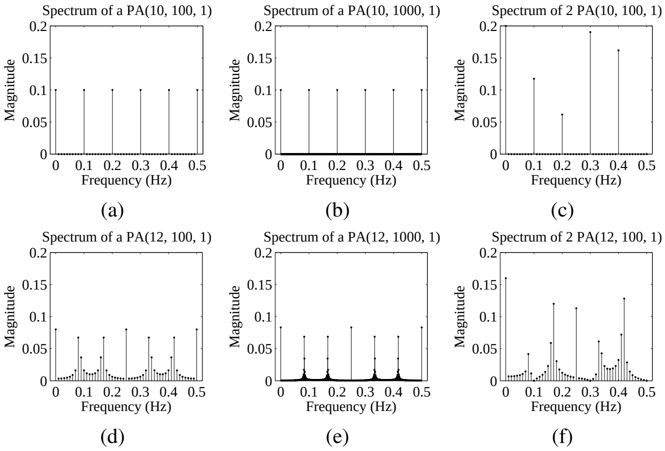


Fig. 1. Demonstration of the spectrums of persistent activities.

magnitude 0.1 and their frequency intervals are 0.1 Hz. In Fig. 1(b), we increase the length of the time series to 1000 and we can see that the converted spectrum presents the same spectral properties as Fig. 1(a) with a finer frequency resolution (i.e., 0.001 Hz). Fig. 1(d) shows the spectrum of $\mathrm{PA}(12, 100, 1)$. Due to the leakage problem, the spectrum has several peaks instead of impulses. The peak intervals (defined as the differences of two adjacent peaks) are close to $1/12$ $(0.08\overline{3})$ Hz. Since, the frequency unit in x-axis in Fig. 1(d) is 0.01 Hz, the converted spectrum depicts an imperfect impulse train due to the picket fence effect and the leakage problem. We can see that the peaks are actually located at frequency indices $\{0, 8, 17, 25, 33, 42, 50\}$ on the x-axis. Note that their intervals are not identical. In Fig. 1(e), we can see that using a longer time series can help increase the frequency resolution in the spectrum and somewhat ease the leakage problem. Overall, we find that the longer the time series is, the better the spectral properties of a single attacker in this case match the properties of the impulse train model.

2) Multiple Attackers: Considering the scenario that there are more than one attack source with same attack period in the network, their integrated time series contains multiple attack time series. We denote P T-periodic attack time series by $\{z^0, z^1, \ldots, z^{P-1}\}$ and their initial delays by $\{\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{P-1}\}$ where $\varepsilon_0 = 0$ and $\varepsilon_{i|i\neq 0} < T$. In signal processing, signal shifting in the time domain corresponds to modulation in the frequency domain. The converted spectrum of the integrated time series is as:

$$\begin{aligned}&\mathcal{F}(\{z^0 + z^1 + \ldots + z^{P-1}\})_f\\ &= \mathcal{F}(\{z_t + z_{t-\varepsilon_1} + \ldots + z_{t-\varepsilon_{P-1}}\})_f\\ &= (1 + e^{-j2\pi f\varepsilon_1/L} + \ldots + e^{-j2\pi f\varepsilon_{P-1}/L})Z_f, \qquad (4)\end{aligned}$$

where $Z_f = \mathcal{F}(z^0)_f$. Accordingly, we can know that the converted spectrum still has a periodic pattern while the magnitudes of these peaks are determined by the modulation effect. In Fig. 1(c) and 1(f), we show the converted spectrum of the integrated time series of two attack sources. Their initial delays are randomly generated. Note the initial delays of the two time series affect the shapes of the spectrum.

As a special case, the integrated time series can become a new periodic time series if their initial delays $\{\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{P-1}\}$ form an arithmetic progression with common difference T/P. In this case, (4) can be derived as

$$\begin{aligned}&\mathcal{F}(\{z_t + z_{t-\varepsilon_1} + \ldots + z_{t-\varepsilon_{P-1}}\})_f\\ &= (1 + e^{-j2\pi fT/(PL)} + \ldots + e^{-j2\pi f(P-1)T/(PL)})Z_f\\ &= \textstyle\sum_{q=0}^{P-1} e^{-j2\pi fqT/(PL)} Z_f. \qquad (5)\end{aligned}$$

Consider (3) and (5), we can infer that in a converted spectrum the magnitude is PZ_f at f when f is a multiple of PL/T, and 0 for other cases of f, if the leakage problem is ignored. Hence, the converted spectrum is equal to the spectrum of a *T/P*-periodic time series.

In a quick summary, if an attack time series consists of several persistent activities with identical period, its spectrum will contain an impulse train-like pattern. The frequency intervals of the peaks in the spectrum are nearly identical and present the periodicity of the persistent activities in the original time series. These peak intervals may not be identical due to the picket fence effect and the leakage problem. The number of attack sources (each one generates a persistent activity) affects the magnitudes of the peaks proportionally, while the initial delays of these interleaved persistent activities affect the spectrum sinusoidally. The length of an attack time series is not the main cause that changes the magnitudes and frequency intervals of the peaks, while a longer time series increases the frequency resolution of the spectrum and eases the picket fence effect and leakage problem to certain level. Finally, if the recurring events delay or fluctuate in the time domain, it may generate more complicated spectral pattern in the frequency domain. We will address this problem in Section VI.

B. Spectral Properties of Legitimate Events

In the early years, researchers usually used Poisson process to model network activities, such as TCP connection arrivals, due to its strong theoretical properties. However, Paxson *et al.* [17] discovered that applications such as TELNET that generates a single TCP connection per user session are well modeled by Poisson processes, whereas applications such as HTTP and X11 which may generate multiple connections per user session are not. Actually, Crovella *et al.* [18] demonstrated and explained why web traffic (i.e., HTTP) exhibits behavior that is consistent with self-similar traffic model. Feldmann [19] also showed that the TCP connection arrival process shows self-similar behavior and that TCP connection interarrival times are statistically better modeled by distributions with heavy tails, especially the Weibull distribution, than traditional models. In [20], Nuzman *et al.* demonstrated that in a local area network user session arrivals are still Poisson distributed and within an individual user session, the number of connections and mean interarrival times are biPareto distributed and the interarrivals of these connections are Weibull distributed. In recent years, the arrivals of user-initiated application sessions, such as web or P2P, were also investigated as Poisson distributed [12], [21]. Even in a wireless network, Lee *et al.* [22] discover that the marginal distribution of the TCP connection interarrival times is piecewise Weibull distributed.

In spite of extensive investigation on the TCP connection arrivals, few papers have discussed and modeled the arrival patterns of UDP flows. Although UDP is connectionless, some intrusion detection systems or firewalls (e.g., Netfilter/iptables) can somehow recognize "UDP connections" for better understanding and distinguishing network activities. In this paper, we consider both outgoing TCP and UDP traffics observed in a monitored network for our time series spectrum analysis.

In the following, we first investigate the scenario that a time series generated by individual host is independent and identical distributed (IID); therefore, so is the aggregate time series of all hosts in the whole network. Then, we investigate non-IID time series. We adopt the two-level model [20] to model real-world traffic and study their spectral properties.

1) IID Time Series: For an IID time series, we propose to use statistical-based approach to analyze its average spectral properties. That is, we study the mean and variance of the magnitude at each frequency of the converted spectrum.

Let's denote the time series of legitimate traffic of whole network by y_t, the expected value of a legitimate spectrum (denoted by Y_f) can be computed as follows:

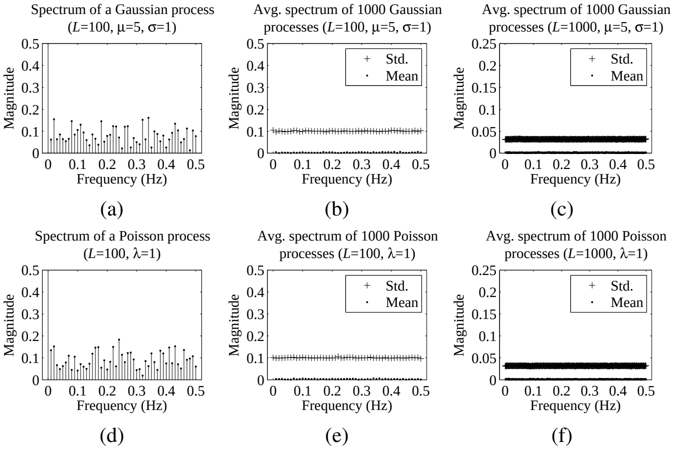
$$\begin{aligned}\mathbf{E}[Y_f] &= \mathbf{E}[\tfrac{1}{L}\textstyle\sum_{t=0}^{L-1} y_t e^{-j2\pi ft/L}] \\ &= \tfrac{1}{L}\textstyle\sum_{t=0}^{L-1}\mathbf{E}[y_t]e^{-j2\pi ft/L} &(6)\\ &= m(\tfrac{1}{L}\textstyle\sum_{t=0}^{L-1} e^{-j2\pi ft/L}) &(7)\\ &= m\delta(f), &(8)\end{aligned}$$

where (6) is derived because the time series is IID and in (7) we further denote $\mathbf{E}[y_t]$ by m to show that it is a time-independent random variable which is only affected by the probability distributions of this aggregate time series. In (8), the $\delta(f)$ is derived that the magnitude at frequency 0 reflects the mean connection statistics of the legitimate time series (which corresponds to the definition of DFT), while the magnitudes at the other frequencies have zero mean. It can be reasoned by the IID legitimate traffic generated by stochastic processes does not tend to concentrate at any particular frequency. The randomness of legitimate traffic is due to the dynamic of user operations which is not caused by any recurring phenomenon.

Similarly, the variance of Y_f can be computed as follows:

$$\begin{aligned}\mathbf{V}\mathrm{ar}(Y_f) &= \mathbf{V}\mathrm{ar}(\tfrac{1}{L}\textstyle\sum_{t=0}^{L-1} y_t e^{-j2\pi ft/L}) \\ &= \tfrac{1}{L^2}\textstyle\sum_{t}^{L-1}\mathbf{V}\mathrm{ar}(y_t)e^{-j4\pi ft/L} &(9)\\ &= v(\tfrac{1}{L^2}\textstyle\sum_{t=0}^{L-1} e^{-j4\pi ft/L} &(10)\\ &= \frac{v}{L},\end{aligned}$$

where (9) is derived because the time series is IID letting us ignore the covariance term when calculating the variance for the legitimate spectrum. Also, in (10), we denote $\mathbf{V}\mathrm{ar}(y_t)$ by v to show that it is also a time-independent random variable which represents the variance of connection statistics of an aggregate time series at each time unit. We find that the variance of Y_f is proportional to the variance of the time series, while degrades according to the increases in the length of the time series. From the above results, we know that in average the spectrum of an IID time series tends to be a flat spectrum.



(a) (b) (c)

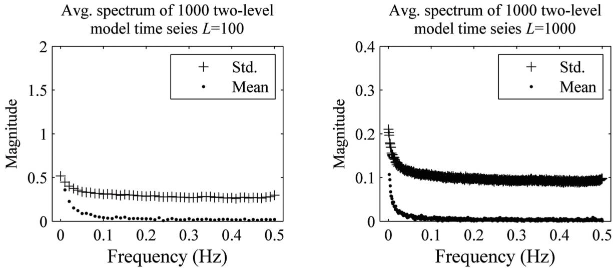
(d) (e) (f)

Fig. 2. Demonstration of the spectrums of IID legitimate events.

To demonstrate our findings, we first use Matlab function *normrnd* to generate a Gaussian-based time series by seting $\mu = 5$, $\sigma = 1$, and $L = 100$ and round these random variables to the nearest integers. A Gaussian-based time series represents the connection statistics following Gaussian distribution at each time unit. Fig. 2(a) depicts the spectrum of such a time series. We can see that there are no clear peaks in this spectrum which tells that the traffic pattern of this kind does not concentrate on specific frequencies. We then repeat this experiment by the same setting for 1000 times and compute the mean and standard deviation of the magnitude at each frequency from these spectrums, and show the average results in Fig. 2(b). We can see that the magnitude at each frequency has zero mean, except at frequency 0, and their standard deviations fall in a very narrow range. Based on the standard deviation, we can infer that the variance at each frequency is about 0.01. Both figures match the mathematical model mentioned above. Further, we increase the length of the time series to 1000 and repeat the experiment of Fig. 2(b). In Fig. 2(c), we can see that the mean magnitude at each frequency in the spectrum is still zero and the standard deviation unitedly becomes smaller. This observation is also consistent with our mathematical model.

In Fig. 2(d), we illustrate a spectrum converted from a Poisson-based time series generated by using Matlab function *poissrnd* and taking $\lambda = 1$ and $L = 100$. Poisson-based time series satisfies the traditional traffic model that user behavior is controlled by an arrival rate with memoryless property. We can see that there are also no clear peaks in this spectrum, similar to Fig. 2(a). We again compute the average spectrums for Poisson-based time series in the same way as Fig. 2(b) and 2(c) and depict the average spectrums in Fig. 2(e) and 2(f). We find that the results both satisfy our mathematical model which indicates a flat spectrum of an IID time series.

2) Non-IID Time Series: As mentioned, the real-world network traffic may not be IID. We now adopt the two-level model [20] to model real-world traffic and study their spectral properties. We configure the parameters as those defined in [20] and generate a 25-day long time series with mean connection statistics about 6. We also measure the Hurst parameter [23] of the derived time series and the value is about 0.63. For self-similar processes, the values of the Hurst parameter are larger than 0.5 and the degree of self-similarity increases as the value increases. This indicates that



(a) (b)

Fig. 3. The average spectrums of time series generated by two-level model.

the synthesized legitimate time series holds the self-similar nature.

We randomly select a short period time series with length 100 and 1000 seconds from this 25-day long time series and perform spectrum analysis. We run the experiment 1000 times, compute the mean magnitude and the standard deviation at each frequency, and plot these values in Fig. 3(a) and 3(b) separately. We can see that since the generated connections may have certain correlation, both the mean magnitudes and the standard deviations slightly increase at the lower frequencies. However, the rest part of the spectrum still has zero mean with nearly consistent standard deviations in magnitude, which corresponds to the observations of an IID time series. Moreover, for a longer time series, the spectrum has lower magnitudes in average.

C. Summary

In comparing the attack and legitimate time series and their spectrums, we summarize our findings as follows:

- The spectrum of a legitimate time series tends to be a flat spectrum that might have high peaks in low frequency.
- For a persistent attack time series, the spectral properties, such as number of peaks and the peak intervals, are mainly affected by its recurring period and are independent to the length of the time series.
- While comparing an attack time series to a legitimate time series, they have opposite effects on their spectral patterns when their time series are increased. For an attack time series, increasing length can mitigate the leakage problem in the spectrum, and hence improve the capability of recognizing the pattern of its impulse train. In contrast, lengthening a legitimate time series decreases the average magnitude in its spectrum.
- In general, the attack spectrum has certain properties which are more stable and suitable for pattern recognition, while the legitimate spectrum is more random and may affect the recognition of the attack.

Based on these findings, we are motivated to design an adaptive detection method by adjusting the length of the inputted time series and inspecting impulse train-like spectral patterns in order to detect persistent activities.

V. Attack Detection

A. Problem Description

Considering a monitored network with M active hosts, N of these M hosts are infected by an attack. All hosts generate legitimate traffic independently, while infected hosts further generate attack traffic. We assume that the attack behavior of infected hosts will not disturb their normal operation; this assumption is tenable especially for slow-paced attacks. Binned connection statistics of the monitored network are recorded to form a time series. A time series x with length L is denoted by $x = \{x_t | t = 0, 1, \ldots, L-1\}$. For a bin t, x_t records connection statistics within time interval $[bt, b(t+1))$ with bin size b seconds.

For a monitored network, x can be further formulated as the aggregation of a legitimate time series y and an attack time series z. That is

$$x = y + z = \sum_{i=1}^{M} y^i + \sum_{j=0}^{N} z^j,$$

where y^i represents the time series of legitimate connections generated by host i and z^j represents the time series of attack traffic generated by infected host j.

We denote X as the spectrum of x by using DFT and our goal is to identify the patterns in X that can be used as the evidence to detect the existence of persistent activities in x. According to the linearity property of DFT, we know that

$$X = Y + Z = \sum_{i=1}^{M} Y^i + \sum_{j=0}^{N} Z^j,$$

where Y and Z are the spectrum of y and z, respectively, and Y^i and Z^j are the spectrum of y^i and z^j, respectively.

There are significantly different spectral features between attack time series and legitimate time series. Therefore, it is practicable to determine the existence of a persistent activity by identifying its spectral pattern from an aggregate spectrum. In the following, we describe the concept and the algorithm of the proposed frequency-based detection method for detecting slow-paced persistent activity embedded in a time series.

B. Detection Concept and Conditions

We treat the detection as a pattern recognition task for finding peculiar spectral pattern of an impulse train in the frequency domain. According to the observations in the previous section, our strategy is to adaptively increase the length of the inputted time series to discover attack events. Increasing time series length decreases the degree of interference caused by legitimate traffic, while the spectral pattern of attack traffic is still preserved. For each iteration, we compute the spectrum of the time series and select candidate peaks using a magnitude threshold (denoted by θ_m). This magnitude threshold is calculated from the time series and will be updated for each iteration. At last, we examine the frequency intervals of these candidate peaks and decide whether there has any persistent activity embedded in the time series. Based on this concept, we design a detection algorithm in the next subsection.

We now discuss how we select θ_m and infer the required length which is defined as the low bound of the length of the time series for a successful detection. According to the mathematical models mentioned in Section IV-B, we know that the magnitudes at each frequency of the spectrum of an IID time series are determined by its spectral variance. These spectral variances are proportional to the variance of the time series. Further, based on central limit theorem, the magnitudes in a legitimate spectrum would be normally distributed. We argue that the magnitudes at the frequencies generated by persistent activity would be higher than the magnitudes at other frequencies. Therefore, we define $\theta_m = \rho\sqrt{v/L}$ and use standard deviation as the basic value for inferring the magnitude threshold. In the experiments, we take $\rho = 3$ because in statistics three standard deviations can account for 99.7% of the magnitudes in the spectrum [24]. In order to have a successful detection, the length of the time series has to be sufficiently long to reduce the spectral variance such that the peaks from attack time series can stand out. To detect a T-periodic persistent activity, the condition that $1/T > \theta_m$ has to be met, as from (3), $1/T$ is the magnitude of the peaks of the attack time series. By the definition of θ_m, we can know that the required length satisfies

$$L > 9vT^2. \tag{11}$$

Practically, we can use this low bound as the initial length for our time series spectrum analysis. However, there are two problems remained. The first one is we cannot know the value of T and number of attackers in advance. Second, the real world traffic may not be an ideal IID time series. Therefore, in our detection algorithm, we use an initial length (denoted by L_{init}) and for each iteration we increase L for discovering patterns with a finer frequency resolution. In our experiments, the maximum length is configured as the length of the traffic trace we had collected from the monitored network. The effect of different time bin sizes will be explored and discussed in the experiments.

We now use Fig. 1(f) as an example to illustrate our detection method. First, we compute θ_m from the analyzed time series and select the candidate peaks from the spectrum. In this case, θ_m is close to 0.11 and the peaks higher than θ_m locate at the frequency indices $\{0, 17, 25, 42\}$. (Note that not all of the attack peaks are above θ_m when we only consider a 100-second long time series in this case). Then, we derive three peak intervals as $\{17, 8, 17\}$. We find that the peak intervals are not the same but the smallest interval is close to the greatest common divisor (GCD) of all intervals. To tolerate this obscuration which may be caused by picket fence effect, leakage problem, or multiple attackers, we define a is a weak GCD (denoted as WGCD) of b if $b = s \times a + r$ where $r \leq s$ or $r \geq a - s - 1$, $s, r \in \mathbb{N}$. We further design a function called $\mathrm{wgcdCount}(a, B)$ in the detection algorithm to count the number that a is a WGCD of other peak intervals in B. Our goal is to find a peak interval who has the highest number of WGCD count, and based on this peak interval we can compute the recurring period of the potential persistent activity. In this example, the peak interval 8 has the largest WGCD count. As the frequency unit is 0.01 Hz, we can infer the target frequency as $1/0.08$, which is 12.5 (close to 12). The result can be more accurate by using a longer time series.

C. Detection Algorithm

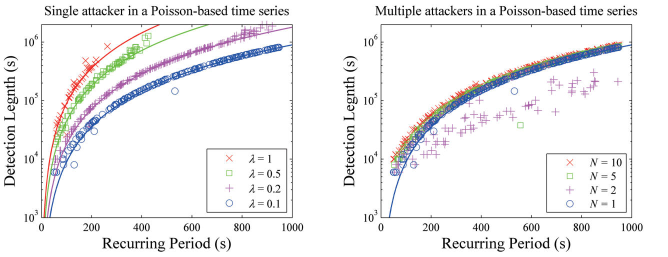
The pseudo code of our detection algorithm is shown in Fig. 4. The inputs of this algorithm are a time series, the bin size of this time series, and a predefined initial length for the detection. The outputs are a flag which indicates whether an impulse train-like pattern is found or not and an identified recurring period and time series length if the flag is set to 1. As mentioned, the algorithm will adaptively adjusts L until it reaches the maximum data length or a successful detection.

```
Algorithm. Detect the Impulse Train-like Pattern in a Time Series
Input: time series $x$, bin size $b$, and initial length $L_{init}$
Output: detection result $r$, recurring period $T_{out}$, and length $L_{out}$
1.  $L_{max} = \mathrm{length}(x)$;
2.  $L = L_{init}$ and $L_{out} = 0$;
3.  while($L < L_{max}$)
4.      $X \leftarrow$ get non-negative frequency spectrum of $\mathrm{abs}(\mathrm{fft}(x(1:L))/L)$;
5.      $\theta_m \leftarrow \rho\sqrt{\mathrm{var}(x(1:L))/L}$;
6.      Select Freq. sections $S_i, i = 1, 2, ...,$ such that $\forall f \in S_i : X_f > \theta_m$;
7.      Select Freq. indices $p_i, i = 1, 2, \ldots$ of the highest impulse for every $S_i$;
8.      Compute $\Delta p = \Delta p_i, i = 1, 2, \ldots$ where $\Delta p_i = p_i - p_{i-1}, p_0 = 0$;
9.      Compute $q_i = \mathrm{wgcdCount}(\Delta p_i, \Delta p)$ for every $\Delta p_i$;
10.     $q_{max} \leftarrow \max(q_i)$; $\Delta p_{out} \leftarrow \Delta p_i$ that generates $q_{max}$;
11.     if $q_{max} > \beta \times \|\Delta p\|$
12.         $f_{out} = \Delta p_{out}/(bL)$;
13.         $r = 1$, $T_{out} = 1/f_{out}$, and $L_{out} = L$;
14.         return;
15.     else
16.         $r = 0$ and $T_{out} = 0$;
17.         $L = \mathrm{increment}(L)$;
18.     end
19. end
```

Fig. 4. Pseudo code of persistent activity identification from a time series.

In each iteration, we first compute the spectrum of the time series by using FFT. Only the spectrum belonging to non-negative frequencies are used for the following investigation (line 4). As mentioned, we compute θ_m by three ($\rho = 3$) standard deviation of the magnitude of the spectrum which can be calculated from the inputted time series (line 5). We ignore the impact of the potential persistent activities in the time series on computing θ_m. We then use θ_m to distinguish outstanding magnitudes from a generally flat spectrum. Due to the leakage problem, the magnitudes of neighboring frequencies around the target frequency may exceed θ_m; they together form a so called frequency section (line 6). We select the highest impulse of each frequency section as the representative impulse and record the frequency (line 7). Then, we compute frequency intervals of these representative impulses (line 8) and count the WGCD of these intervals using $\mathrm{wgcdCount}(a, B)$ (line 9). If the interval with the largest WGCD count (line 10) is a factor of more than β of all peak intervals (where $0 < \beta < 1$), we claim to identify a persistent activity (line 11). The reason of line 11 is to remove insignificant peak intervals incurred from noise. We then compute the output parameters (line 12 to 13) and return (line 14). Otherwise, we further increase the length of the time series and do the inspection again (line 15 to 19). The increment function in line 17 can be arithmetic increment (i.e., $L = L + \alpha$) or geometric increment (i.e., $L = (1 + \alpha)L$), up to the decision of users. The time complexity of the algorithm is $O(L \log^2 L)$.

Note that in some cases, our algorithm may falsely report a rate which is the multiple of the target recurring rate. That means T_{out} is a factor of true recurring period T; this is because the inputted attack time series is not long enough to obtain necessary peaks exceeding θ_m. In such case, the algorithm usually stops earlier (than it supposes to be) and we argue that we still have a successful detection with a shorter period, as this information is still useful in detection the attack. We call this phenomenon as early detection effect.



(a)

(b)

Fig. 5. Detecting persistent activities in a Poisson-based time series. (a) The detection lengths of a single attacker. λ indicates the arrival rate of the legitimate traffic. (b) The detection lengths of multiple attackers. λ is fixed as 0.1. The solid lines in (a) are derived from Eq. (11).

D. *Power Aggregation at Low Frequencies*

In the preliminary experiments, we have shown that legitimate time series tend to generate a flat spectrum except slightly higher magnitudes at the lower frequencies. The truth is we also observe this kind of phenomenon in the spectrum of our collected network traffic time series. For a long-term investigation, the daily effect of the network traffic, which has different traffic at day and night time, may result in this long-term regularity as well. Therefore, we also consider to use a frequency threshold (denoted by θ_f) to only keep the spectrum at high frequencies and ignore the spectrum at frequencies lower than θ_f. In our experiments, we will show that by using θ_f the proposed detection algorithm will have higher accuracy.

VI. Performance Evaluation

In this section, we first describe the dataset and the metrics for the experiments. We then demonstrate the evaluation results of our proposed detection method for both synthesized time series and the time series acquired from real-world trace.

A. *Dataset*

For an attack time series, we not only configure the recurring period but also consider other parameters to make the simulated time series model more realistic. Specifically, we extend (2) to model a T-periodic attack time series of length L for individual host as follows:

$$z_t^j = \sum_{k=0}^{R_j-1} \delta(t - kT - \varepsilon_j - d), \qquad (12)$$

where ε_j (where $0 \le \varepsilon_j < T$) represents an initial delay of infected host j, d simulates a random network delay in the monitored environment, and R_j is number of impulses of each host during the observation (i.e., $R_j = \lfloor (L - \varepsilon_j)/T \rfloor$ when ignoring the effect of d).

For a monitored network, we generate an attack time series by configuring the number of attack sources (N) and the recurring period (T) based on (12). We also configure a value for the possible maximum network delay (denoted by d_{max}) and model d as a random variable uniformly distributed within 0 and d_{max} for each event. The initial delay of individual host (ε_j) is automatically randomly chosen from 0 to $T - 1$.

We inject the attack time series into three types of legitimate time series to validate the correctness of proposed model and the capability of algorithm. For the first synthesized time series, the connection statistics is a Poisson distributed random variable. It is anticipated that this type of time series tends to generate a flat spectrum. Secondly, we generate legitimate time series based on a more realistic two-level model as mentioned in Section IV-B. At last, we study the characteristics of real-world traffic time series with injected synthesized attack time series. We extract this real-world time series from 164.8 GB NetFlow traces collected from 24 subnets of a campus network for 25 days (i.e., around $2,160,000$ seconds).

TABLE I
Number of Successful Detection Under Various Experiment Settings in Different Figures

5(a)		5(b)		6(a)		6(b)		8(a)		8(b)	
λ	D	N	D	θ_f	D	N	D	d_{max}	D	b	D
1	32	10	180	0	22	10	180	8	95	1	95
0.5	75	5	180	0.05	180	5	180	4	132	2	139
0.2	174	2	180	0.1	180	2	180	2	175	4	179
0.1	180	1	180			1	180	0	180	8	179

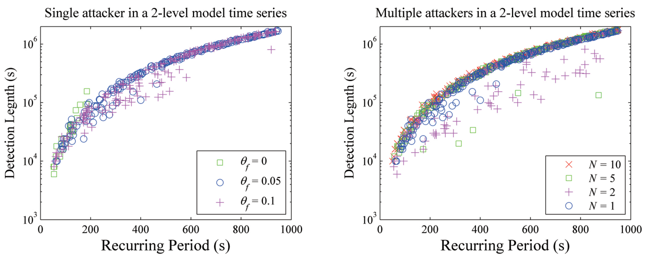
Note: the first row shows the figure number. The number of successful detection is denoted by D*.*

B. *Synthesized Traffic Simulation*

In this section, we focus on the accuracy of the proposed detection algorithm by examining the required length of the time series for successful detection under various experiment settings. We now use Fig. 5(a) as an example to describe the methodology of our experiments. We use detection length in the y-axis to represent the required length of the time series for successful detection. In Fig. 5(a), we run a series of experiments for a single attacker in a Poisson distributed legitimate time series. We use different arrival rate λ to compare the impacts of different legitimate traffic volume to the detection algorithm. We randomly select T from 50 to 950 for an experiment and repeat the experiment for 180 times. We plot each outputted detection length (L_{out}) in the figure if the detection is successful (i.e., the algorithm returns a detection result.) In the algorithm, we configure $L_{init} = 2000$ and use geometric increment (by setting $\alpha = 0.2$) to update time series length. We set $\beta = 0.5$ for selecting peak intervals with sufficiently large WGCD count.

In Fig. 5(a), we can see that the detection length increases rapidly against the recurring period which coincides with (11). The solid lines in Fig. 5(a) represent the low bound of the required length derived from (11). If $\lambda = 0.1$, the detection algorithm can detect all the persistent activities within a three-week long time series, and the numbers of successful detections decrease with the increase of λ, as found in Table I. The reason of failed detection is mainly because the required time series length is longer than $2,000,000$ seconds. Note that, in Fig. 5(a), some experiments have shorter successful detection length due to the early detection effect described in Section V-C. In Fig. 5(a), there are only $(6, 2, 1, 2)$ early detections for $\lambda = (1, 0.5, 0.2, 0.1)$ respectively.

In Fig. 5(b), we study the accuracy of our detection algorithm when there are multiple attackers in a Poisson-based legitimate time series. We fix λ as 0.1 for each attacker, vary N from 2 to 10, and also run the experiments for 180 different recurring periods for each N. We found that the proposed detection algorithm can detect all the persistent activities in



(a) (b)

Fig. 6. Detecting persistent activities in a two-level model-based time series. (a) The detection lengths of a single attacker by using θ_f. (b) The detection lengths of multiple attackers.

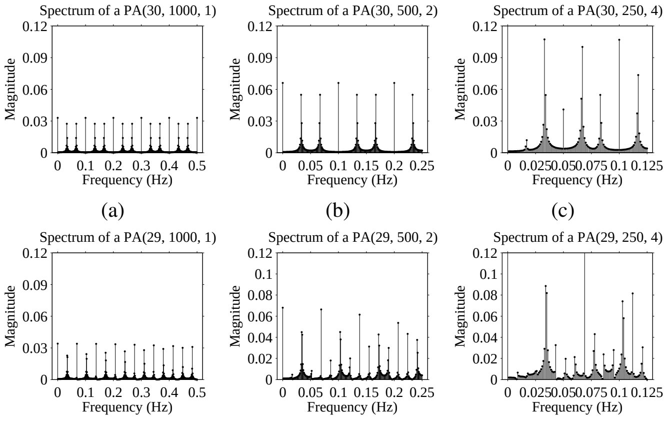
these experiments even if the number of the attack source is 10. Moreover, the detection length for detecting multiple attackers is similar to the detection length for detecting a single attacker due to more attack evidence in the time series. Besides, we notice that only for the case of $N = 2$, we have many early detections (62 times in this case). This is because two overlapped persistent activities have a higher chance to form a regular series than a complex case with more persistent activities.

We then study the cases that the legitimate traffic is modeled by a more realistic two-level model. As mentioned, the user session is modeled by Poisson arrival, while the connections within individual user session are controlled by biPareto and Weibull distributions. By tuning these distribution parameters, we keep the average connection arrival rate of the generated time series as 0.1 connection per second for the experiments in order to compare to the results in Fig. 5(a). In Section IV-B, we have shown that this kind of non-IID time series tend to have higher magnitudes at the lower frequencies of the spectrum. Therefore, we use the frequency threshold (θ_f) to filter out the spectrum at lower frequencies and only focus on the rest of the spectrum for the detection.

In Fig. 6(a), we demonstrate that when $\theta_f = 0$, which means we look at the whole spectrum for the detection, we can only detect persistent activities with small recurring periods. The θ_m is not able to prevent the interferences caused by some random peaks at lower frequencies. However, if we increase θ_f to 0.05 or 0.1 Hz, the numbers of successful detections raise to 180 for both cases (see Table I). In Fig. 6(b), we set $\theta_f = 0.05$ and again measure the performance of our detection algorithm when there are multiple attackers in a time series. We apply each attacker the same configuration as that in Fig. 6(a) and we can see that the results are close to the results in Fig. 5(b). Although we may have some early detection, the proposed algorithm can successfully detect persistent activities even if they are generated from multiple attack sources. By using a suitable frequency threshold, we can detect all the persistent activities even if there is more than one source of persistent activities.

C. The Effects of Using Time Bin

As mentioned in Section III-C, using different bin sizes in forming a time series will result in the frequency resolution change of the converted spectrum due to the change of measurement unit (i.e., sampling unit) of the original time series. For a time series with bin size b and length L, the Nyquist frequency of the converted spectrum is $1/2b$ and the frequency resolution becomes $1/Lb$. In this subsection, we discuss the effects of using a time bin for persistent activity detection.



(a) (b) (c)

(d) (e) (f)

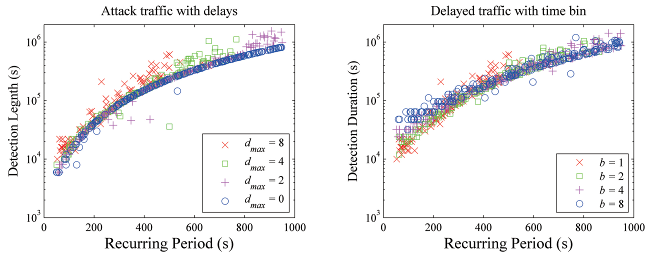
Fig. 7. Demonstration of the spectral effects of a binned time series.

Legitimate time series. To form a time series from IID distributions, using different bin sizes only changes the mean and variance of the binned time series, which still maintains IID and the converted spectrum tends to be flat as well. The main factor to shape the spectrum is the variance of the binned time series. For a more complex case, such as the time series generated by the two-level model, using time bin may alter the pattern because the connections are not IID. However, we argue that the aggregate time series of network traffic are from multiple sources. The overall traffic pattern may become random, and hence the binned time series preserve the similar spectral properties as the original time series.

Attack time series. A binned time series of a persistent activity still has temporal regularity. If the recurring period T can be divided by the bin size b (i.e., $\mathrm{mod}(T, b) = 0$), the time series contains a persistent activity with period T/b. In this case, it is clear that impulse train-like pattern can be observed in the frequency domain. For the cases when mod $(T, b) \neq 0$, the situation would be more complex. However, we claim that in this case the binned time series still maintains its temporal regularity according to Lemma 1 (Its proof is in Appendix A).

Lemma 1: *If a binary time series of a T-periodic event is binned at a bin-size b where $b < T$ and b and T are integers, then the binned time series will be equivalent to the aggregate of b/φ interleaved (T/φ)-periodic events where $\varphi = gcd(T, b)$.*

Note that the case that $\mathrm{mod}(T, b) = 0$ is covered by Lemma 1 as well. According to Lemma 1 and (4), we conclude the spectral properties for a binned persistent activity as follows. The spectrum of a binned persistent activity will have rounded (T/φ) peaks and the frequency intervals of these peaks are closing to φ/bT. At frequency 0, we can find the maximum peak with magnitude as b/T and the magnitudes of other peaks are equal to or lower than b/T.



(a) (b)

Fig. 8. The impacts of network delay and the use of a time bin for the detection method. (a) The detection lengths under different conditions of network delay. (b) The detection lengths by using a time bin to deal with network delay.

In Fig. 7, we demonstrate the changes of the spectral pattern of a binned attack time series, for both the cases of $\mathrm{mod}(T, b) = 0$ and $\mathrm{mod}(T, b) \neq 0$. Fig. 7(a) depicts the spectrum of a $\mathrm{PA}(30, 1000, 1)$ which represents a 30-periodic time series with $L = 1000$ and $b = 1$. If we form a time series by setting $b = 2$, the binned time series, $\mathrm{PA}(30, 500, 2)$, will only be 500 long. According to Lemma 1, we know that this binned time series will contain only one 15-periodic time series. In this case the Nyquist frequency becomes 0.25 Hz and the spectrum will contain 8 peaks at the non-negative frequencies (see Fig. 7(b)). If the original time series is summarized by using $b = 4$, from Lemma 1, it is equivalent to 2 15-periodic time series interleaved in this 250-second-long binned time series. Therefore, in Fig. 7(c), the Nyquist frequency becomes 0.125 and the number of peaks at non-negative frequencies is also 8.

Fig. 7(d) shows the spectrum when the period of a time series is 29 seconds. According to Lemma 1, we know that using a time bin only increases the interleaved periodic events co-existing in the binned time series. Therefore, in Fig. 7(e) and 7(f), the number of peaks is the same as that in Fig. 7(d), while the magnitude of these peaks is affected by the modulation effect caused by using time bin.

In a quick summary, we understand that the spectrum of a binned time series inherits certain spectral properties from the spectrum of the original time series. For analyzing a time series with the same duration, using time bin can improve the efficiency on spectrum analysis.

Another advantage of using time bin is that it can defeat frequency fluctuation caused by network delay to certain level. We now demonstrate this advantage on our detection algorithm by using simulation. We add some uniformly distributed random delays into the attack time series to create the fluctuation of the persistent activity. We assume that there is a single attacker in the time series and we use $\lambda = 0.1$ of Poisson arrival to build the legitimate time series. We found that if the problem of delay is getting worse, we require a longer length of the time series for a successful detection, as found in Fig. 8(a) and Table I. Network delays may result in a reduction of the number of peaks passing the magnitude threshold test of our algorithm at higher frequencies.

To overcome the impact of network delay, we reconstruct a time series for a large time scale by using a time bin. We fix d_{max} as 8 and rerun the experiments by configuring the bin size b from 1 to 8. In Fig. 8(b), we found that the numbers of successful detections are increased as shown in Table I. Note that, in Fig. 8(b), the y-axis represents the duration, defined as Lb, of successful detection. Therefore, for a larger time bin, the actual length of the time series used for spectrum analysis is shorter, which also means the spectrum analysis is more efficient.

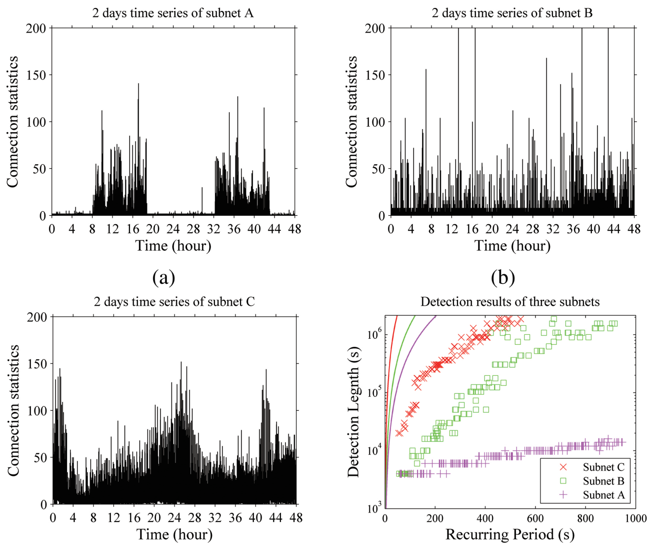
D. Real-World Trace

We also evaluate our detection algorithm against the real-world traffic collected for 25 days in a campus network. We monitor all the outgoing connections at an egress interface of a dorm network with 24 class-C subnets that each subnet has different characteristics in their network traffic. For example, the average number of connections per second ranges from 0.45 to 92.73 and the standard deviations ranges from 2.39 to 85.06. The Hurst parameters for the daily time series and 25-day time series of each subnet range from 0.67 to 0.96 and from 0.69 to 0.96, respectively, confirm that the self-similar nature do exist in the collected real-world traffic traces.

In this experiment, we first extract legitimate time series from these real-world traffic traces. Then, we manually inject synthesized attack time series into the time series of different subnets to check the capability of our detection algorithm. Specifically, for each subnet we inject ten persistent attackers ($N = 10$), each of which generates T-periodic time series based on (12). For each round of the experiment, we only change the value of T, while ignore the network delay (i.e., keep $d_{max} = 0$). Since the real-world traffic volume is huge, we found that it is extremely hard to detect a slow-paced persistent activity in such a short time series.

In Fig. 9, we show the time series of three subnets that have very different connection characteristics and the detection lengths of multiple attackers. The Hurst parameters of the daily time series of these three subnets are 0.78, 0.64, and 0.92 respectively. We can see that subnet A has relatively light traffic and distinct traffic patterns for daytime and nighttime (Fig. 9(a)). Our detection algorithm can detect almost all the attacks with detection length less than $20,000$ seconds (see Fig. 9(d)) because subnet A has a small amount of traffic at night. Subnet B generates a medium traffic rate through the whole observation period (Fig. 9(b)) and subnet C generates a heavy traffic (Fig. 9(c)). Their average numbers of connections are 0.7 and 11.0 per second and the standard deviations are 4.1 and 10.1 respectively. In Fig. 9(d), we can see that our detection algorithm can detect most of the attacks in subnet B with some false negatives. For subnet C, only the attacks whose recurring periods are smaller than 600 seconds can be detected.

The solid lines in Fig. 9(d) are derived from (11) using v of the whole time series. The results show that our detection algorithm can detect persistent activities whose recurring periods are longer than our expectation. For example, according to (11), a 25-day time series of subnet C are expected to detect persistent activities when their recurring periods are smaller than about 50 seconds, while in the real world trace, the recurring period larger than 400 seconds can be detected. The reason is that the real-world traffic is fluctuating, such as



(a) (b)

(c) (d)

Fig. 9. The time series and the detection results of three different subnets. (a), (b), and (c) are the time series of subnet A, B, and C. (d) depicts the detection lengths of multiple attackers ($N = 10$) in the subnets ($\rho = 3$). The solid lines from left to right are for Subnet C, B, and A.

daytime and nighttime patterns and even unexpected bursts, which presents a large variance in the time series. Consequently, the prediction is under-estimated. These experiments show that the proposed detection algorithm can deal with the uncertainty of the real-world traffic, as expected from the proposed formulation in Section IV.

In Table II, we show the impacts of different ρ (i.e., different θ_m) to the detection results in three different subnets. In subnet A, the proposed method can detect persistent activities for all ρ and all recurring periods, because of the small volumes of legitimate traffic at night that helps discover the persistent activities. Most of the false positive cases of subnet A are actually distorted persistent activities with wrong recurring periods. The reason is that early detection loses the frequency resolution to analyze the persistent activities. In subnets B and C, the false positives are mainly caused by the interferences of real-world traffic, and not all the persistent activities are detected due to insufficient length of real-world trace. In subnets A, B, and C, the number of false positives drops rapidly with larger ρ, which indicates a longer detection length. Note that the number of detection in subnet C decreases when $\rho = 4$ because the required length for successful detection exceed the length limitation of the experiment.

Based on our analysis, we also find that in some subnets the spectrums usually have peaks at lower frequencies and the intervals of these impulses are close to 0.03 Hz. This implies that some events with period lengths of about 32 $\sim$ 34 seconds are embedded in the time series, and the occurrence of these events seems more dynamic in the time domain. By checking the original traffic trace, we find that the main cause is the popularity of P2P applications in the dorm network. Many hosts in the traffic trace has similar behaviors, in that every 32 $\sim$ 34 seconds one host will make connections to a number of different peers at different destination ports. Practically, these kinds of legitimate traffic incur false positives to our detection method. An approach to eliminate FPs is to filter out traffic generated by this kind of applications in advance by using a whitelist.

TABLE II
NUMBERS OF DETECTION AND FALSE POSITIVE IN DIFFERENT SUBNETS

	Detection/False positive		
Subnet	$\rho = 2$	$\rho = 3$	$\rho = 4$
A	180/104	180/78	180/26
B	105/92	130/42	176/0
C	59/46	92/1	60/0

VII. Discussion and Conclusion

Malware is getting sophisticated and the means of hackers have evolved. In order to enforce network security, we need to defeat unknown attacks. The proposed detection method intends to detect the stealthy persistent activities in the frequency domain and the experiments show to overcome the interference caused by network noise. However, if the malware is crafted to obfuscate the temporal regularity of the planned attacks, e.g., by introducing a random delay between operations, our approach may fail. But such obfuscation may create detectable signature that can be detected by other detection methods, such as signature-based system. We believe the proposed approach can cooperate with other detection methods to build a robust network security mechanism.

We select to use DFT as our spectrum analysis tool due to its efficiency and simplicity. Other spectrum analysis tools, such as short-term Fourier transform (STFT) or wavelet, can also be used to investigate spectral characteristics of a time series. These tools may have advantages in investigating spectral characteristics in different time scales.

In this paper, we propose using spectrum analysis to detect slow-paced persistent activity, especially for network activities. We investigate the spectral properties for both persistent attack activities and legitimate traffic by using mathematical models, simulations, and real-world traces. We prove that the proposed method can adaptively identify persistent activities in a time series amid legitimate traffic based on spectral analysis. Moreover, we believe this method is applicable for analyzing other periodic phenomena. This method, however, still has some limitations. For example it cannot deal with the situation that there are multiple types of attacks exist in the same time. This limitation will be a significant focus of our future work.

Appendix A

Proof of Lemma 1: We first define that a binary time series $x(t)$ ($t = 0, 1, 2, \ldots$) is said to be T-periodic iff $x(t) = 1$ when $t = \alpha T + \delta$, where $\alpha \in \mathbb{N}$ (including 0) and δ is a constant and $\delta < T$, and $x(t) = 0$, otherwise. Then, we denote a binned time series by $x_{\overline{b}}(k)$ ($k = 0, 1, 2, \ldots$) and $x_{\overline{b}}(k) = \sum_{i=0}^{b-1} x(bk + i)$. Since the value of δ can be regarded as an initial delay and does not affect the periodicity of the binned time series, we choose $\delta = 0$ to make the following description more concise.

Let's call the list of t such that $x(t) = 1$ list A and the list of k such that $x_{\overline{b}}(k) = 1$ list B. We have $A = \{nT | n = 0, 1, 2, \ldots\}$ and $B = \{\lfloor nT/b \rfloor | n = 0, 1, 2, \ldots\}$. Let's define

$T = \beta b + \gamma$. We can see that if $\gamma = 0$, $x_{\overline{b}}(k)$ is simply β-periodic. However, if $\gamma \neq 0$, the periodicity of list B will be disturbed due to the value of γ. We further assume that b and T are coprime (i.e., $\varphi = 1$) and separate list B into b sub-list B_m where $B_m = \{\lfloor (nb+m)T/b \rfloor | n = 0, 1, 2, \ldots\}$ and $m = 0, 1, \ldots, b-1$. Replacing T with $\beta b + \gamma$, we have

$$\begin{aligned} B_m &= \{\lfloor ((n\beta b + m\beta + n\gamma)b + m\gamma)/b \rfloor | n = 0, 1, 2, \ldots\} \\ &= \{n\beta b + m\beta + n\gamma + \lfloor m\gamma/b \rfloor | n = 0, 1, 2, \ldots\} \\ &= \{nT + C | n = 0, 1, 2, \ldots\}, \end{aligned}$$

where $C = m\beta + \lfloor m\gamma/b \rfloor$ which is a constant for each sub-list B_m. Although C may be larger than T, we can easily add this large part from the second term to the first term and said that B_m is also T-periodic. At last, since list B can be reconstructed by sub-lists B_m based on a regular pattern, we can infer that the binned time series will contain b interleaved T-periodic events.

In the cases of b and T are not coprime, we can first modify each sub-list by reducing the fraction by φ. By doing the same procedure, we can derive that in these cases B_m are all (T/φ)-periodic, and hence, the binned time series will contain b/φ interleaved (T/φ)-periodic events.

References

[1] A. Lakhina, M. Crovella, and C. Diot, "Characterization of Network-Wide Anomalies in Traffic Flows," in *Proc. of Internet Measurement Conference*, Oct 2004.

[2] C. C. Zou, W. Gong, D. Toesley, and L. Goa, "The Monitoring and Early Detection of Internet Worms," *IEEE/ACM Transaction on Networking*, Oct 2005.

[3] S. Staniford, J. A. Hoagland, and J. M. McAlerney, "Practical Automated Detection of Stealthy Portscans," *Journal of Computer Security*, 10, 2002.

[4] V. Sekar, Y. Xie, M. K. Reiter, and H. Zhang, A Multi-Resolution Approach for Worm Detection and Containment, in Porc. IEEE/IFIP International Conference on Dependable Systems and Networks, Jun 2006.

[5] F. Giroire, J. Chandrashekar, N. Taft, E. Schooler, and D. Papagiannaki, "Exploiting Temporal Persistence to Detect Covert Botnet Channels," in *Proc. International Symp. on Research in Attacks, Intrusions and Defenses*, Sep 2009.

[6] F. Akujobi, I. Lambadaris, and E. Kranakis, "An Integrated Approach to Detection of Fast and Slow Scanning Worms," in *Proc. ACM Symp. on Information, Computer and Communications Security*, Nov 2009.

[7] Y. Gao, Y. Zhao, R. Sch Schweller, S. Venkataraman, Y. Chen, D. Song, and M.-Y. Kao, "Detecting Stealthy Spreaders Using Online Outdegree Histograms," in *Proc. International Workshop on Quality of Service*, Jun 2007.

[8] W. Yu, X. Wang, A. Champion, D. Xuan, and D. Lee, "On Detecting Active Worms with Varying Scan Rate," *Computer Communications*, Jul 2011.

[9] L. M. Chen, M. C. Chen, W. Liao, and Y. S. Sun, "A Scalable Network Forensics Mechanism for Stealthy Self-Propagating Attacks," *Computer Communications*, May 2013.

[10] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies," in *Proc. ACM Internet Measurement Workshop*, Nov 2002.

[11] M. Zhou and S. D. Lang, "Mining Frequency Content of Network Traffic for Intrusion Detection," in *Proc. IASTED International Conference on Communication, Network, and Information Security*, Dec 2003.

[12] B. Kim, H. Kim, and S. Bahk, "FDF: Frequency Detection-based Filtering of Scanning Worms," *Computer Communications*, Mar 2009.

[13] S. Shin, G. Gu, N. Reddy, and C.P. Lee, "A Large-Scale Empirical Study of Conficker," *IEEE Transactions on Information Forensics and Security*, Apr 2012.

[14] P. Porras, H. Saidi, and V. Yegneswaran, "An Analysis Of Conficker's Logic and Rendezvous Points," *SRI International, Tech. Report*, Mar 2009.

[15] R. G. Lyons, "Understanding Digital Signal Processing," *Prentice Hall*, 2nd ed., 2004.

[16] R. Walter, "Real and Complex Analysis," *New York: McGraw-Hill*, 3rd ed., 1987.

[17] V. Paxson and S. Floyd, "Wide-area Traffic: the Failure of Poisson Modeling," *IEEE/ACM Trans. Networking*, Jun 1995.

[18] M. E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Trans. Networking*, Dec 1997.

[19] A. Feldmann, "Characteristics of TCP Connection Arrivals," *Tech Report*, Jan 1998.

[20] C. Nuzman, I. Saniee, W. Sweldens, and A. Weiss, "A Compound Model for TCP Connection Arrivals for LAN and WAN Applications," *Computer Networks*, Oct 2002.

[21] J. Kannan, J. Jung, V. Paxson, and C. E. Koksal, "Semi-automated Discovery of Application Session Structure," in *Proc. of Internet Measurement Conference*, Oct 2006.

[22] I. W. C. Lee and A. O. Fapojuwo, "Analysis and Modeling of a Campus Wireless Network TCP/IP Traffic," *Computer Networks*, Oct 2009.

[23] J. Beran, "Statistics for Long-Memory Processes," *Chapman and Hall*, Oct 1994.

[24] T. T. Soong, "Fundamentals of Probability and Statistics for Engineers," *Wiley* 1st ed., Mar 2004.

Li Ming Chen received the MS degree in Computer Science, and the PhD degree in Electrical Engineering from National Taiwan University, Taiwan, in 2004 and 2013, respectively. He holds a postdoctoral research fellowship at the Institute of Information Science, Academia Sinica, Taiwan. His research interests are in the areas of network security and computer networks.

Shun-Wen Hsiao received his BS and PhD degree from the Department of Information Management from National Taiwan University in 2004 and 2012, respectively. Currently, he holds a postdoctoral research fellowship at Institute of Information Science, Academia Sinica, Taiwan. His research interests are in the area of computer networks, network security, and the secure and performance issues in virtualized environment.

Meng Chang Chen received the BS and MS degrees in Computer Science from National Chiao Tung University, Taiwan and the PhD degree in Computer Science from the University of California, Los Angeles, in 1989. Then, he joined AT&T Bell Labs as Member of Technical Staff. Currently, he is a Research Fellow with Institute of Information Science, Academia Sinica, Taiwan. His current research interests include wireless network, network security, information retrieval, and auto-quiz for English learning.

Wanjiun Liao received her BS and MS degrees in Computer Science from National Chiao Tung University, Taiwan, in 1990 and 1992, respectively, and her PhD degree in Electrical Engineering from the University of Southern California in 1997. She is a Distinguished Professor and Chair of Department of Electrical Engineering, National Taiwan University. Her research focuses on the design and analysis of wireless multimedia networking, cloud-data center networking and green communications. Prof. Liao was on the editorial boards of IEEE Transactions on Wireless Communications and IEEE Transactions on Multimedia. She was a recipient of the Republic of China Distinguished Women Medal in 2000 and was elected as a Distinguished Lecturer of IEEE Communications Society for 2011-2012. She is a Fellow of IEEE.