Organizational Security Resource Estimation via Vulnerability Queueing

Abdullah Y. Etcibasi[1], Zachary Dobos[2], C. Emre Koksal[1]
[1]Electrical and Computer Engineering, [2]Integrated Systems Engineering
The Ohio State University, Columbus, OH, USA
{etcibasi.1, koksal.2}@osu.edu, dobos.20@osu.edu

***Abstract*—We provide an approach that closely estimates an organization's cyber resources directly from vulnerability timestamps, using a non-stationary queueing framework. Existing attack-surface metrics rely on static snapshots and fail to capture the underlying attack–defense dynamics, which are inherently bursty, heavy-tailed, and capacity-constrained. We model these dynamics through a queueing abstraction of the attack surface. Applied to both large-scale software supply-chain data and enterprise security workflows, the framework estimates key organizational resources, including active personnel and service capacity, directly from vulnerability report and patch timestamps. Our results provide 91-96% accuracy in estimation of resources, making the dynamic queueing framework an interesting option for understanding attack surface dynamics. Further, our framework exposes resource bottlenecks, establishing a new foundation for predictive workforce planning, patch-race modeling, and proactive cyber-risk management.**

***Index Terms*—vulnerability dynamics, queueing theory, resource management, dynamic security, cyber risk assessment**

I. Introduction

Information systems are continuously exposed to evolving vulnerabilities (e.g., software flaws, misconfigurations, zero-day exploits), and the set of unpatched vulnerabilities defines the time-varying *attack surface*, a key measure of cyber risk. As systems scale and attackers become more automated, the attack surface evolves dynamically and is inherently *non-stationary*, with fluctuating arrivals and resource-constrained patching leading to persistent backlogs. Traditional metrics, such as average open tickets or mean patch time, fail to capture these temporal dynamics and are therefore inadequate for risk assessment and workforce planning. Existing approaches largely rely on snapshot-based metrics [1]–[5], focusing on long-term averages. However, empirical evidence shows that vulnerability discovery and patching are bursty, heavy-tailed, and subject to regime shifts [6], [7], suggesting that the attack surface is better modeled as a stochastic, capacity-constrained dynamical system.

This paper addresses these gaps by providing a framework for reconstructing resource dynamics for predictive planning. In particular, we build a data-driven dynamic queueing framework to model the spatio-temporal evolution of an attack surface. In our approach, we treat vulnerabilities as jobs arriving, patches as services for these jobs, and the queue-length distribution (QLD) to understand the ensemble statistics of the attack surface. The model represents how vulnerabilities arrive, persist, and depart, while accounting for limited defense budgets, attacker–defender asymmetries, and AI-driven acceleration of attack and defense.

We demonstrate that our framework is highly effective in estimating latent organizational parameters, such as effective workforce size and throughput, from event logs. Towards that goal, we focus on two critical security environments: (1) software supply chains, where vulnerabilities propagate via third-party components and package dependencies (open-source data), and (2) large-scale IT service and ticketing systems (proprietary data). We map these traces as parameters within our model (e.g. arrival rates, exploit probabilities). Aligning model dynamics with observed security events, we were able to estimate the actual resources in personnel and their workload with an accuracy of 4-5%.

We used extensive amounts of data from an open-source software supply chain (publicly available) and the vulnerability ticketing system of a global supply chain enterprise (private), and use our novel projection techniques to fit the model parameters that are variable in time. We compare with actual values whenever available to demonstrate the accuracy of our resource estimates.

This paper makes four key contributions. First, we propose a dynamic queueing framework that captures the spatio-temporal evolution of the attack surface. Second, we develop a Gaussian mixture model (GMM)-based segmentation and simulation-driven estimation approach

This work was supported in part by The Ohio State University President's Research Excellence Catalyst Award, 139152.

ISBN 978-3-903176-82-9

to model non-stationary vulnerability dynamics and recover time-varying system parameters from empirical QLDs. Third, we demonstrate that latent organizational resources, including effective workforce size and service capacity, can be accurately inferred directly from event-level timestamp data. Finally, we validate the framework on both open-source and enterprise datasets, showing high fidelity to observed QLDs and strong agreement with independently observed capacity.

II. Related Work

Early approaches to cyber risk focused on probabilistic frameworks and attack graphs, modeling exploit reachability and attacker movement across systems [1], [3]–[5]. Extensions based on Bayesian and Markov models capture conditional dependencies and sequential exploit chains across complex environments such as cloud and industrial systems. More recent AI-assisted approaches further improve automation in attack modeling, but they still rely on static system snapshots and typically assume immediate patching, thereby overlooking backlog dynamics and resource constraints.

Empirical evidence shows that vulnerability discovery and patching processes are inherently bursty, heavy-tailed, and non-stationary, often leading to persistent backlogs [7], [8]. Real-world systems exhibit prolonged patch delays, multi-modal backlog distributions, and regime shifts over time, as highlighted by datasets such as ARVO [8]. These observations indicate that the attack surface evolves as a stochastic, capacity-constrained dynamical system, requiring models that account for both arrival variability and limited service capacity.

Among existing work, [6] is, to the best of our knowledge, the only study that adopts a dynamic modeling perspective for the attack surface. However, it uses epidemic dynamics and does not capture queueing behavior under finite service capacity or enable data-driven parameter estimation. Other stochastic and network-based models capture temporal dependencies and multi-step exploit patterns but do not model patching as a constrained service process or account for resource bottlenecks. In contrast, we model the attack surface as a non-stationary queueing system with finite capacity, capturing bursty arrivals, heavy-tailed service times, and time-varying dynamics, while enabling direct inference of organizational resources from event-level data.

III. System Model and Problem Formulation

A central contribution of this paper is a queueing-theoretic framework for modeling the attack surface of a system or organization, where vulnerabilities arrive stochastically and persist until removed by defensive actions. Let $N(t)$ denote the attack surface size (queue length), i.e., the number of active unmitigated vulnerabilities at time t. The state evolution of the attack surface is governed by the following discrete-time recursive equation:

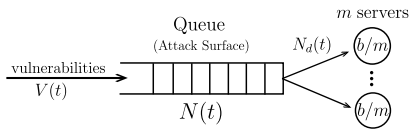


Fig. 1. Queueing representation of the attack surface.

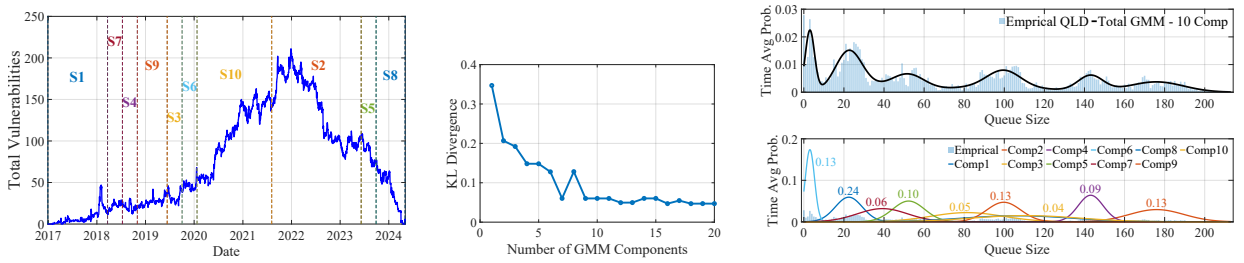
$$N(t{+}1) = [N(t) + V(t) - N_d(t)]^+ \tag{1}$$

where $V(t)$ is the number of new vulnerabilities discovered during the interval $[t, t+1)$, $N_d(t)$ represents the number of vulnerabilities mitigated by defensive actions like patching or system reconfigurations. The operator $[x]^+ = \max(x, 0)$ enforces non-negativity of the state. This abstraction, shown in Fig. 1 shows how vulnerabilities $V(t)$ arrive, join the queue of active vulnerabilities $N(t)$, while defenders remove a subset each time step via patching efforts. In the software supply chain setting (ARVO), $N(t)$ is the reported yet unpatched software bugs, while in the logistics enterprise dataset it represents open security tickets in the organizational workflow.

Stochastic processes for the arrival $V(t)$ and the defensive departure $N_d(t)$ are inherently non-stationary. Vulnerability arrivals are bursty and heavy-tailed, while patching times display substantial variability and long-tailed behavior. Since the departure is capacity-constrained and governed by these service dynamics, both arrival and departure induce temporal heterogeneity and dependence in the backlog $N(t)$.

To capture resource constraints, we adopt a $G/G/m$-b formulation, where arrivals and ST follow general distributions and the system has m parallel servers under a finite aggregate resource constraint b. Note that our notation differs from the standard Kendall's framework. While the $G/G/m/k$ model uses k to specify the maximum job capacity, the proposed $G/G/m$-b model uses m as the number of parallel service agents and b as a constraint on total service rate.

In the cyber-security setting considered, m represents the effective number of active patching agents. In the software supply chain case (ARVO), it corresponds to the number of developers or maintainers simultaneously resolving reported vulnerabilities, whereas in the logistics enterprise case it is the number of IT personnel assigned to vulnerability related tickets, with each individual handling at most one ticket at a time. The aggregate capacity parameter b captures the organiza-



(a) Queue-length evolution with cut points defining quasi-stationary segments.

(b) KL divergence between the empirical QLD and GMMs vs. number of components.

(c) Ten-component GMM fit to the empirical QLD with individual components and aggregate mixture.

Fig. 2. Segmentation and mixture-model fitting results for the ARVO dataset.

tion's total patching throughput, measured as completed vulnerabilities or tickets per unit time.

Each server has a fixed service capacity of b/m per unit time, given a total system capacity of b. Thus, m captures the scale of the defensive workforce, while b/m captures the average per-agent productivity, reflecting efficiency. This parameterization separates staffing levels from per-person effectiveness, offering a flexible and interpretable framework to analyze attack surface dynamics under resource limits.

Having characterized the attack surface, we next estimate the parameters of the queueing model using a simulation-based optimization framework. Given event-level vulnerability data comprising discovery and patch timestamps, we carefully segment the data into quasi-stationary periods. For each segment, our objective is to identify the parameter set θ^* that minimizes the Kullback-Leibler (KL) divergence between the empirical QLD, $\hat{P}(n)$, and the simulated QLD, $P(n, \theta)$, produced by the candidate model.

This measure quantifies the discrepancy between the empirical QLD and the distribution generated by a candidate model. The parameter space Θ is defined for the $G/G/m$-b queueing model such that for each segment, $\theta = \{m, b, F_{IA}, F_{ST}\}$, where m is the number of parallel servers, b is the total service capacity, and F_{IA} and F_{ST} are the parametric distributions governing the inter-arrival (IA) and service time (ST), respectively.

Our objective is not merely to match empirical and simulated QLDs via KL minimization, but to use this optimization to identify time-varying resource patterns in a non-stationary system. We model the effective workforce size and service capacity as segment-dependent quantities, $m(t)$ and $b(t)$, and estimate them over quasi-stationary intervals. This enables recovery of temporal variations in staffing and throughput that reproduce the observed backlog dynamics. While KL provides a measure of statistical fit, the primary goal is to infer time-varying organizational capacity from non-stationary data.

IV. A Segmented Framework for Modeling Non-Stationary Attack Surfaces

This section presents a data-driven algorithm for fitting a queueing model to non-stationary vulnerability dynamics. Due to strong temporal heterogeneity, a global stationarity assumption is inappropriate. We therefore proceed in four stages: (i) construct the empirical QLD from event-level data, (ii) identify quasi-stationary regimes via GMM, (iii) estimate segment-specific queue parameters through simulation-based matching, and (iv) evaluate the fit using parametric IA and ST models and comparisons with empirical and bootstrap QLDs.

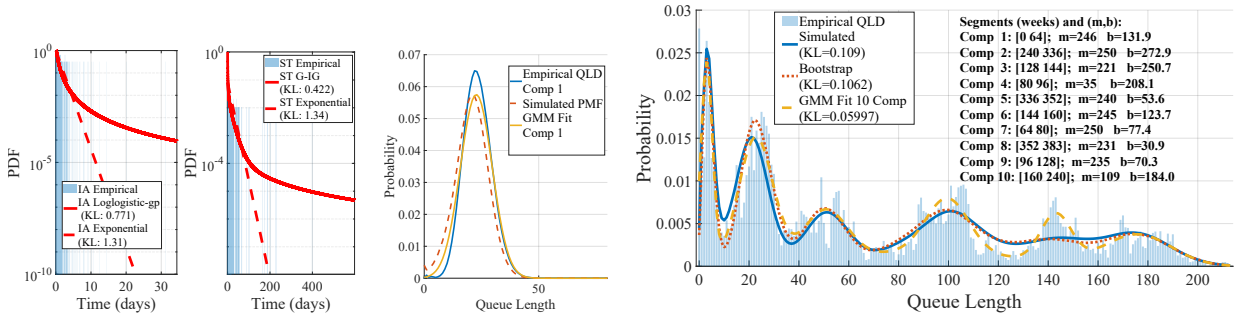
A. Empirical QLD Construction and Stationarity Assessment

The algorithm begins by aligning vulnerability discovery and patching timestamps to reconstruct the discrete-time backlog trajectory $N(t)$ (e.g., Fig. 2(a) and Fig. 4(a)). Rather than analyzing this highly irregular time series directly, we construct a time-averaged QLD $\hat{P}(n) = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}_{\{N(t)=n\}}$, where T is the observation horizon and $\mathbf{1}_{\{\cdot\}}$ is the indicator function. Thus, $\hat{P}(n)$ represents the fraction of time the system operates at backlog level n.

The time-averaged QLD captures the fraction of time spent at each backlog level, enabling detection of non-stationarity and multimodality due to regime shifts. Empirically, it often exhibits multiple modes (e.g., Fig. 2(c) and Fig. 4(c)), indicating that a single parametric model is insufficient. This motivates the use of a GMM to represent regime-dependent behavior via a finite mixture of quasi-stationary components.

B. GMM Fitting

To obtain the quasi-stationary segments, we first fit a univariate GMM with c components to the empirical QLD. Let the continuous mixture density be $p_c(x) = \sum_{k=1}^{c} \pi_k \mathcal{N}(x \mid \mu_k, \sigma_k^2)$, where $\pi_k \geq 0$, $\sum_{k=1}^{c} \pi_k = 1$,



(a) Best-fitting IA and ST models for Component 1 (Weeks 0–64). Heavy-tailed mixtures achieve the lowest KL.

(b) Simulated vs. empirical QLD for Component 1 under fitted parameters.

(c) Overall empirical QLD with segmented $G/G/m$-b simulation, bootstrap estimate, and 10-component GMM benchmark.

Fig. 3. Segment-level fitting and overall QLD validation for ARVO. The segmented $G/G/m$-b model closely matches empirical behavior (KL 0.109).

and $\{\mu_k, \sigma_k^2\}_{k=1}^{c}$ are the mixture parameters. Since the queue length n is discrete, we evaluate the density on integer values and normalize to obtain a discrete pmf, $P_c(n) = \frac{p_c(n)}{\sum_{\ell \in \mathcal{K}} p_c(\ell)}, \quad n \in \mathcal{K}$.

The parameters are estimated via the Expectation–Maximization (EM) algorithm using samples drawn from $\hat{P}$ in proportion to empirical time occupancy.

For each mixture order c, we evaluate the approximation error using the KL divergence $D_{KL}(\hat{P} \,\|\, P_c)$. This provides a quantitative measure of fit as a function of mixture complexity c (e.g., Fig. 2(b) and Fig. 4(b)).

To select the mixture order, we define the marginal improvement $\Delta_c = D_{KL}(\hat{P}\|P_{c-1}) - D_{KL}(\hat{P}\|P_c)$ and choose the smallest c such that $|\Delta_c| < \varepsilon$ for a predefined tolerance $\varepsilon > 0$. The selected order provides a stable representation of the empirical QLD and serves as a reference for subsequent identification of the $G/G/m$-b parameters $\theta = \{m, b, F_{IA}, F_{ST}\}$.

C. Identifying Quasi-Stationary Segments

Let $\{t_0 = 0 < t_1 < \cdots < t_S = T\}$ denote candidate cut points on a coarse weekly grid over the observation horizon $[0, T]$. Each pair (t_i, t_j) with $i < j$ defines a candidate segment $\mathcal{S}_{i,j} = [t_i, t_j)$.

For each candidate segment $\mathcal{S}_{i,j}$ and parameters (m, b), we generate a simulated QLD $P^{\text{sim}}_{m,b,i,j}(n)$ using segment-specific IA and ST samples, and evaluate its alignment with the GMM component P_c via $\mathcal{L}_{i,j}(m, b) = D_{KL}\left(P^{\text{sim}}_{m,b,i,j} \,\|\, P_c\right)$.

For each segment $\mathcal{S}_{i,j}$, we compute provisional parameters $(m', b') = \arg\min_{m,b} \mathcal{L}_{i,j}(m, b)$ and define the segment score $\mathcal{L}_{i,j} = \mathcal{L}_{i,j}(m', b')$. The goal of this step is not to precisely estimate (m, b), but to assess how well a candidate interval behaves as a quasi-stationary regime. Accordingly, the search over (m, b) is performed on a coarse grid for computational efficiency, with refined estimates (m^*, b^*) obtained after segmentation is fixed.

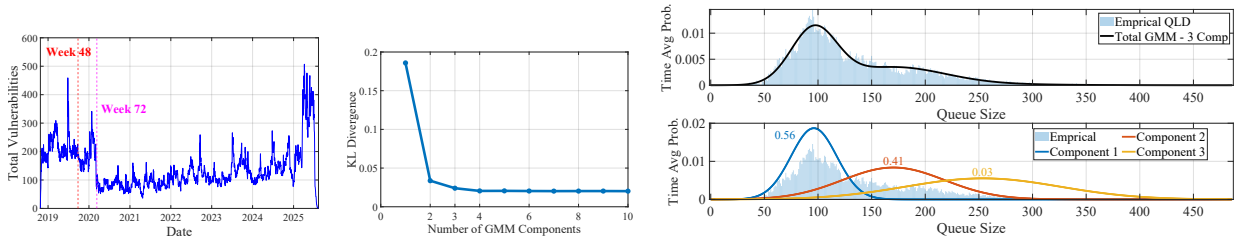
The simulated distribution $P^{\text{sim}}_{m,b,i,j}(n)$ is obtained via bootstrap simulation: IA and ST samples are resampled with replacement from empirical segment data. ST samples are scaled by m/b to satisfy the total capacity constraint, and the resulting queue trajectory is simulated to produce an empirical pmf. Repetitions are averaged until convergence of the running pmf estimate.

Among all feasible segments, we select the mutually exclusive partition identified via coarse grid search that achieves the lowest aggregate score $\sum \mathcal{L}_{i,j}$ over the observation horizon. These segments define quasi-stationary intervals as illustrated by the vertical dotted lines in Fig. 2(a) and Fig. 4(a).

D. Parametric Fitting and Validation

Conditioned on the segmentation in Section IV-C, we refine the queue parameters within each segment via a fine-grained search over (m, b). For each segment $\mathcal{S}_{i,j}$, we solve $(m^*, b^*) = \arg\min_{m,b} D_{KL}\left(P^{\text{sim}}_{m,b,i,j} \,\|\, P_c\right)$ using a higher-resolution search around (m', b'). The resulting (m^*, b^*) defines the refined service configuration.

Given fixed segmentation and (m^*, b^*), we model IA and ST distributions parametrically for each segment using samples extracted from event timestamps. We consider both light- and heavy-tailed families, including lognormal, log-logistic, generalized Pareto, generalized extreme-value distributions, as well as mixture models (e.g., lognormal GMMs and two-component EM fits), and select the best model based on empirical fit (Fig. 3(a), Fig. 5(a)–(b)). Each segment's model is then simulated under (m^*, b^*), and the resulting QLDs are aggregated as $P^{\text{agg}}(n) = \sum_s \pi_s P^{\text{sim}}_s(n)$. An aggregate bootstrap QLD is constructed in parallel, and both are compared with the empirical QLD and GMM mixture to assess how well the framework reproduces the observed dynamics.



(a) Queue-length evolution with cut points defining quasi-stationary segments.

(b) KL divergence between the empirical QLD and GMMs vs. number of components.

(c) Three-component GMM fit to the empirical QLD with individual components and aggregate mixture.

Fig. 4. Segmentation and mixture-model fitting results for the logistics enterprise dataset.

V. Results

We evaluate the proposed framework on two datasets: a large-scale public vulnerability dataset (ARVO) and a multi-year enterprise security dataset. Both provide event-level timestamps that enable reconstruction of arrival and service processes and the resulting attack-surface dynamics. Detailed dataset descriptions and preprocessing steps are deferred to [9].

This section focuses on empirical validation of the segmentation-based queueing framework. Beyond matching QLDs, our primary objective is to infer system parameters and assess whether the estimated (m, b) values align with true organizational capacity. We first evaluate the method on ARVO to demonstrate its ability to capture non-stationary, heavy-tailed dynamics. We then apply the framework to the enterprise dataset, where independently observed workforce and throughput data enable direct validation, showing close agreement with the inferred parameters.

A. Software Supply Chain

We begin by applying the segmentation-based queue modeling framework of Section IV to ARVO. The objective is to determine whether the segmented $G/G/m-b$ abstraction can faithfully reproduce the empirical attack surface dynamics observed in ARVO.

First we reconstruct the attack surface trajectory $N(t)$ from vulnerability report (arrival) and fix (service completion) timestamps. The resulting queue-length evolution over time is shown in Fig. 2(a). The system exhibits pronounced non-stationarity. Periods of backlog growth are followed by intervals of rapid reduction, indicating time-varying imbalance between vulnerability discovery and patching.

From this trajectory, we construct the empirical QLD (Fig. 2(c)). The distribution is clearly multi-modal, with multiple distinct peaks corresponding to different regimes. This observation confirms single stationary queue model is unlikely to capture the full dynamics of the attack surface.

Following Section IV-B, we approximate the empirical QLD using GMMs with varying numbers of components. The KL as a function of mixture complexity is shown in Fig. 2(b). Fit quality improves sharply up to 10-components and then exhibits diminishing returns. Based on this elbow behavior, we select a 10-component GMM to represent distinct quasi-stationary regimes.

Each mixture component is mapped onto the time axis by identifying contiguous intervals whose segment-level QLDs best match the corresponding GMM component in terms of KL. The resulting segmentation (Fig. 2(a)) partitions the queue-length trajectory into quasi-stationary regimes, each with distinct arrival and service dynamics. The clear shifts in $N(t)$ across segment boundaries support the validity of the GMM-based segmentation. Within each segment, we extract IA and ST samples and fit parametric mixtures as described in Section IV-D.

Fig. 2(a) shows representative IA and ST fits for an early segment (Weeks 0–64). The best-fitting models are a log logistic–generalized Pareto mixture for arrivals (KL = 0.771) and a gamma–inverse Gaussian (G-IG) for ST (KL = 0.422). The resulting simulated QLD, shown in Fig. 3(b), closely tracks the empirical distribution for that segment.

These results confirm that, once segmented, vulnerability arrivals and patching times for intervals admit stable **heavy-tailed** distributions that yield accurate queue-length predictions.

Finally, we aggregate the segment-wise simulations into a global model of the attack surface. Fig. 3(c) compares the empirical QLD with three baselines: a bootstrap simulation constructed from the empirical IA and ST, a 10-component GMM approximation, and the segmented $G/G/m-b$ model.

The segmented model achieves a divergence of 0.1074, closely matching the bootstrap estimate (0.1069) and approaching the GMM benchmark (0.05997). Across segments, the number of servers m remains stable (typically 221–250), while the effective capacity b varies widely (30.9 to 272.9), reflecting substantial

temporal fluctuations in patching throughput. These variations correspond to changes in patching intensity over time and shows how defensive resource allocation directly shapes the attack surface distribution.

The ARVO results reveal several important quantitative insights. First, the vulnerability arrival and patching exhibit pronounced non-stationarity and heavy-tailed behavior, as evidenced by the fitted IA and ST mixtures and the clear regime shifts visible in Fig. 2(a). Second, the identified quasi-stationary segments produce a global simulated QLD with KL 0.109, closely matching the empirical bootstrap benchmark (0.1062). This close quantitative agreement with the empirical QLD demonstrates that the segmented $G/G/m$-b model captures the essential structural dynamics of the backlog.

The recovered parameters reveal meaningful temporal variation in defensive capacity. While the effective number of active servers m remains relatively stable across regimes (typically in the range 221–250), the aggregate service capacity b varies significantly (30.9 to 272.9), reflecting substantial shifts in patching intensity over time. These variations are consistent with observable phases of backlog expansion and contraction in Fig. 2(a), demonstrating that the model does not merely reproduce distributional features but also identifies interpretable changes in throughput.

These results provide strong empirical validation that the proposed non-stationary queue reconstruction framework accurately captures the attack surface evolution observed in ARVO. In the next section, we apply the same methodology to the logistics enterprise dataset, where independent workforce and throughput records enable direct quantitative validation of the inferred (m, b) parameters.

B. Logistics Enterprise

As described in Section IV-A, we reconstruct the attack-surface queue from vulnerability discovery and patch timestamps. Fig. 4(a) shows the resulting queue-length trajectory, which is clearly non-stationary: the backlog is high in 2019–2020, then declines and stabilizes, with a brief surge near the end. To focus on representative regimes and avoid boundary effects, we restrict the analysis to 2019–2025. Using this interval, we construct the empirical QLD (Fig. 4(c)), which is distinctly multi-modal, with a dominant peak around 100 and significant mass near 150. This indicates multiple operating regimes, motivating the use of GMMs to identify quasi-stationary segments.

In Fig. 4(b), we report the KL divergence between the empirical QLD and GMMs with varying numbers of components. The marginal improvement diminishes sharply beyond three components, indicating limited benefit from additional complexity. Accordingly, we adopt a three-component GMM, corresponding to three quasi-stationary regimes. This contrasts with ARVO, where 10 components were required to capture stronger temporal variability, while the enterprise dataset exhibits more stable dynamics. The individual components and their weighted fit are shown in Fig. 4(c), where the three-component mixture closely matches the empirical QLD.

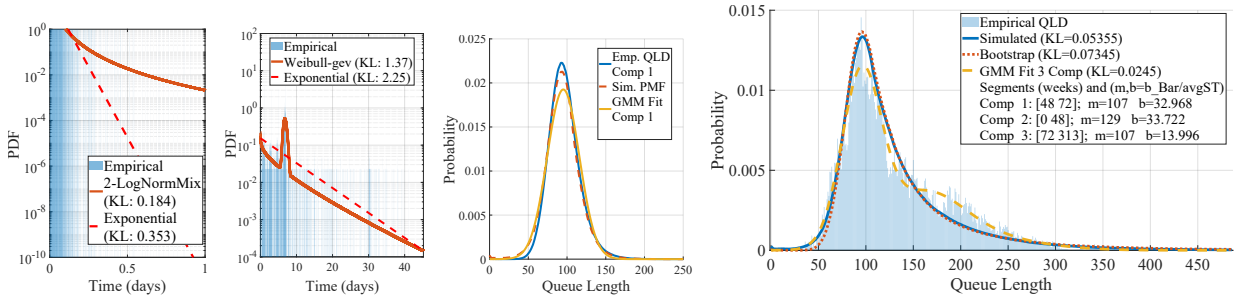
Using the segmentation procedure in Section IV-C, we identify three segments over the weekly time axis with cut points at $[0, 48]$, $[48, 72]$, and $[72, 350]$, corresponding to Jan. 2019–Oct. 2019, Oct. 2019–Apr. 2020, and Apr. 2020–Feb. 2025. As shown in Fig. 4(a), this segmentation isolates two quasi-stationary regimes separated by a short transition period. We then estimate the queue parameters (m, b) for each segment using the procedure in Section IV-D. Fig. 5(b) shows a representative example, where the simulated QLD closely matches the empirical distribution.

To obtain necessary capacity parameters, IA and ST should be identified for each segment. Although fine-tuning identifies an effective aggregate service parameter $\bar{b}_i$ for segment i, the average ST varies across segments, and thus $\bar{b}_i$ must be normalized. Specifically, we compute the effective service capacity as $b_i = \bar{b}_i \,/\, \mathbb{E}[\mathrm{ST}_i]$, where $\mathbb{E}[\mathrm{ST}_i]$ is the mean ST in segment i. This normalization ensures that the inferred service capacity reflects the actual throughput of the system rather than segment-specific temporal scaling effects, and is thus essential for consistent parameter interpretation (cf. Fig. 1).

Representative fitted distributions for IA and ST of the first segment are shown (Fig. 5(a)–(b)). The IA distribution exhibits a heavy-tail, indicating variability in vulnerability discovery rates, with extended periods of low activity interspersed with bursts of arrivals. In contrast, the ST distribution displays a distinct mode around seven days, reflecting patching and service-level targets favor of one week. The presence of this dominant time scale indicates structured patching workflows rather than memoryless service behavior.

Finally, by aggregating the segment-wise simulation results, we obtain the overall QLD shown in Fig. 5(d). The aggregated model accurately reproduces the empirical QLD, yielding a KL of only 0.05355 between the simulated and empirical QLDs.

Beyond distributional fidelity, the inferred model parameters admit a direct interpretation. In the proposed framework, m corresponds to the number of active personnel resolving tickets (with at most one ticket per worker at a time), while b represents the total capacity measured in jobs per day. The segment-wise analysis reveals that the effective workforce size remains relatively stable at $m \approx 110$, whereas the total capacity decreases over time, from roughly 33 jobs/day in the



(a) IA fits for Component 1 (Weeks 0–64).
(b) ST fits for Component 1 (Weeks 0–64).
(c) Simulated vs. empirical QLD for Component 1 under fitted parameters.
(d) Aggregated empirical QLD with segmented $G/G/m$-b simulation, bootstrap estimate, and GMM benchmark.

Fig. 5. Segment-level fitting and overall QLD validation for the logistics enterprise dataset. The aggregated segmented model closely matches the empirical QLD (KL = 0.05355).

TABLE I
SEGMENT-WISE ESTIMATES OF THE EFFECTIVE NUMBER OF ACTIVE DEFENDERS (m) AND AGGREGATE SERVICE RATE (b), WITH CORRESPONDING SIMULATION-DERIVED VALUES ($m_{\text{sim}}, b_{\text{sim}}$).

Segment	Weeks	m	b (jobs/day)	m_{sim}	b_{sim} (jobs/day)
1	$[48, 72]$	105.99	23.61	107	32.9
2	$[0, 48]$	119.72	33.10	129	33.7
3	$[72, 313]$	103.48	15.41	107	13.9

early segment to about 14 jobs/day in the final segment, reflecting measurable changes in throughput.

Independent organizational records enable direct quantitative validation of these estimates as reported in Table I. The inferred workforce size deviates by less than 1% in the first segment ($m = 107$ estimated versus 106 observed), and remains within 5% across all segments. Similarly, the inferred service capacity in the final segment is 14 jobs/day, compared to an empirical average of 15.41 jobs/day, corresponding to a relative error of 9%. Across segments, the recovered parameters consistently track observed staffing and throughput levels with low relative error.

These results demonstrate that the proposed non-stationary queue reconstruction framework does not merely reproduce QLDs, but accurately identifies the time-varying organizational resources driving backlog dynamics. Importantly, these resource estimates are obtained solely from event-level arrival and closure timestamps, without direct access to staffing data. This ability to recover m and b from traces highlights the practical value of the framework for workforce planning and capacity forecasting in large-scale cyber-security operations.

VI. CONCLUSION

In this work, we develop and validate a dynamic queueing-theoretic model to model cyber attack surfaces. The proposed model was evaluated across two distinct and operationally critical environments: software supply chain vulnerability data and large-scale IT service and ticketing systems. A key outcome of this study is the ability to recover latent organizational resources including the workforce size and load per personnel using only event level timestamps, with estimation errors typically within 4–9%. Our approach provides a quantitative foundation for operational security planning, enabling security leaders to quantify current staffing, predict resources for a given attack intensity, and evaluate the impacts of shifts in attack frequency or defensive throughput.

REFERENCES

[1] P. K. Manadhata and J. M. Wing, "An attack surface metric," *IEEE Transactions on Software Engineering*, vol. 37, no. 3, pp. 371–386, 2011.

[2] C. Harry, I. Sivan-Sevilla, and M. McDermott, "Measuring the size and severity of the integrated cyber attack surface across us county governments," *Journal of Cybersecurity*, vol. 11, no. 1, p. tyae032, 2025.

[3] J. A. Jones, *FAIR – Factor Analysis of Information Risk*. Risk Management Insight LLC, 2011.

[4] H. Wang, D. Zhang, and S. Jajodia, "An attack-graph based probabilistic security metric," in *IFIP Data and Applications Security '08*. Springer, 2008, pp. 109–124.

[5] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, 2012.

[6] K. Haldar and B. K. Mishra, "Mathematical model on vulnerability characterization and its impact on network epidemics," *International Journal of System Assurance Engineering and Management*, vol. 8, no. 2, pp. 378–392, 2017.

[7] A. Feutrill, M. Roughan, J. Ross, and Y. Yarom, "A queueing solution to reduce delay in processing of disclosed vulnerabilities," in *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 2020, pp. 1–11.

[8] X. Mei, P. S. Singaria, J. Del Castillo, H. Xi, T. Bao, R. Wang, Y. Shoshitaishvili, A. Doupé, H. Pearce, B. Dolan-Gavitt *et al.*, "Arvo: Atlas of reproducible vulnerabilities for open source software," *arXiv preprint arXiv:2408.02153*, 2024.

[9] A. Y. Etcibasi, Z. Dobos, and C. E. Koksal, "Organizational Security Resource Estimation via Vulnerability Queueing," *arXiv preprint arXiv:2604.10250*, 2026.