Article
JOURNAL OF UNIVERSITY OF BABYLON
For Pure and Applied Sciences (JUBPAS)
Vol.34 ; No. 2 | 2026

A Hybrid Quantitative Risk Assessment Framework for Zero-Trust Architectures Using Stochastic Petri Nets and Attack Graphs

Wijdan Noaman Marzoog Al Mukhtar*
Biology Department, College of Science for Women, University of Babylon, Babylon, Iraq
wsci.wijdan.marzoog@uobabylon.edu.iq, Babylon, Iraq.
***Corresponding author email:** wsci.wijdan.marzoog@uobabylon.edu.iq; mobile: 07759003239

التقييم الكمي للمخاطر في بنى الثقة الصفرية باستخدام شبكات بيتري العشوائية ورسوم بيانية للهجوم

وجدان نعمان مرزوك المختار *
قسم علوم الحياة، كلية العلوم للبنات، جامعة بابل، بابل، العراق wsci.wijdan.marzoog@uobabylon.edu.iq، بابل، العراق

Accepted:	25/6/2026	Published:	30/6/2026

ABSTRACT

Although there has been a significant increase in the adoption of Zero Trust Architectures (ZTA), very few formalized methods exist for evaluating the continuously changing risks of these systems. Our work proposes a hybrid framework based on Stochastic Petri Nets (SPNs) and Attack Graphs that assesses the systemic risks associated with a ZTA deployment through the explicit modelling of micro-segmentation, least privilege, and re-authentication intervals. We introduce three quantitative metrics to aid in assessing the risk from a systemic point of view: Mean Time to Security Breach (MTTSB), Expected Loss Exposure (ELE), and Conditional Value at Risk (CVaR). We evaluated the framework using a simulation of a 150-microservice environment, using our ZTA-Breach-150 synthetic dataset (calibrated for statistical accuracy using the Verizon Data Breach Investigations Database - VCDB and CSE-CIC-IDS2018), which simulated 10,000 breaches over five different types of attack. The simulations produced a predicted mean time to successful breach (MTTSB) within four standard deviations of the empirical average MTTB for breaches that occur in real-world networks. The simulation was based on a cloud-native E-Commerce Microservices architecture running on Kubernetes with ZTA enforcement through a Policy Decision Point (PDP) and a Policy Enforcement Point (PEP). Sensitivity analysis demonstrates that using re-authentication intervals of 90-120 seconds minimizes ELE by 37% compared to static policies. The proposed framework has O(n2) scaling characteristics (approximately linear in practice) and can be used to provide a basis for risk-aware adaptive decision-making in next-generation access control through provably quantified risk metrics.

Keywords: Zero Trust Architecture, Stochastic Petri Nets, Attack Graphs, Quantitative Risk Assessment, Adaptive Access Control.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

INTRODUCTION

The conventional security systems based on a perimeter are no longer effective as they were before due to the erosion of the traditional perimeter through cloud adoption, microservices and remote working practices. Firewalls and VPNs were traditionally used as local defenses because they assumed that all connected devices inside this area of networked devices (an entity) were inherently trustworthy; however, this is not the case because lateral movement after an initial compromise causes many of the largest breaches in modern enterprises [1]. Consequently, Zero-Trust Architecture (ZTA) has developed into the most widely used enterprise security framework; the foundation of ZTA is built upon the concept to "never trust and always verify." ZTA ensures that continuous authentication of users exists, the least privilege access principle is granted, the micro-segmentation of an application is maintained at all times, and dynamic re-authentication occurs at regular intervals [2]. While ZTA is an appealing concept and rapidly being adopted in the industry, ZTA has not provided concrete, quantitative, and time sensitive risk metrics that measure the stochastic nature of attacks or measure the constant fluctuation of ZTA policies.

Typical risk assessments conducted for Zero Trust environments are generally static or qualitative. Many implementations depend on static trust scores that are based on rules or the outcomes of periodic audits. These trust scores do not reflect the changing nature of an attack as time passes [3]. Traditional methods to quantify the risk associated with vulnerabilities such as attack trees or CVSS scores provide structure to vulnerabilities but do not incorporate the probabilistic timing associated with an attacker's actions, delays in detection, or the effect of re-authentication intervals. As a result, security operators are unable to answer the fundamental question of "What is the expected time of a critical asset breach with a given re-authentication policy?" and "How does the expected financial loss change as we move downward in the trust threshold?" Due to this uncertainty, ZTA deployments find it difficult to make proactive, risk aware decisions.

The attack graph has been a technique for modeling exploit chains and possible breach paths for many years [4]. They have a systematic way for enumerating through the various sequence of vulnerabilities an attacker can exploit to compromise a target asset. Attack graphs do not consider time, concurrency, or randomness and will, therefore, be seen as static in their representation of logical reachability. They do not consider that the actions of the attacker have a time component; that there are, for example, defences (e.g., re-authentication) that work to a schedule; and that there could be multiple steps in an attack that are being executed concurrently. Stochastic Petri Nets (SPNs) are a sophisticated formalism for modelling concurrent, timed and random behaviour in stochastic systems [5]. SPNs have been used successfully for the purpose of dependability and security analysis, to support intrusion response, and to estimate time to compromise. However, there are few if any existing SPN security models that integrate the specifics of the Zero Trust Architecture controls, with the most notable of these being dynamic re-authentication intervals and adaptive least-privilege enforcement. In order to apply quantitative rigor to the ZTA risk assessment process, the hybridization of attack graphs and SPNs is important and urgent because attack graphs provide a structurally complete representation of attack paths, whereas SPNs provide the stochastic and concurrent modelling capabilities.

ISSN: 2312-8135 | Print ISSN: 1992-0652 | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com | info@journalofbabylon.com

To fill this gap, we propose a new quantitative risk assessment framework for Zero Trust Architectures that is based on combining attack graphs with Stochastic Petri Nets (SPN). The framework models three key components of a Zero Trust Architecture (ZTA), these are: micro segmentation, which prevents unauthorized lateral movement; least privilege access, which restricts ways to escalate privileges; and reauthentication intervals (time periods in which all trust is reset) that require periodic reauthentication. We define three quantitative risk metrics that are relevant for ZTAs: Mean Time to Security Breach (MTTSB), which measures the expected time before the first security breach; Expected Loss Exposure (ELE), which measures total expected financial losses over some period of time; and Conditional Value-at-Risk (CVaR), which measures average losses in the worst-case scenarios (generally, in bad cases such as top 5% of outcomes). These metrics are generated from an SPN model created using an attack graph representation of the system being analyzed. Recent studies have continued to highlight this persistent gap. Gambo and Almulhem (2026) concluded in their systematic literature review that most ZTA frameworks lack quantitative risk metrics that capture dynamic attacker behavior. Similarly, Ahmadi (2024) noted that existing ZTA deployments rely on heuristic decision-making for re-authentication intervals without model-based justification. Ma et al. (2025) recently used SPNs to evaluate re-authentication intervals but assumed a single predefined attack path, leading to potentially optimistic risk estimates. These findings confirm that the problem remains unresolved and that a comprehensive framework combining structural attack path enumeration with stochastic timing analysis is urgently needed.

We confirm our framework's validity through a simulated enterprise network comprised of 150 microservices, comparing how long before an attack would be successful due to an attacker's actions compared to similar attacks from the available dataset based on real security incidents. The results show the model will predict the maximum time to successfully breach the network within 4 standard deviations from the actual succinct data. The extensive sensitivity analysis conducted demonstrated that the optimal re-authentication period, or the period at which the risk of loss of potential or actual assets (loss exposure) is minimized, lies between 90-120 seconds. Using a dynamically determined optimal policy, we reduce expected loss exposure by approximately 37% compared to using a static re-authentication policy (e.g., every 5 minutes). We also demonstrate that the framework provides the ability to support risk-aware adaptation in real time by dynamically reducing the re-authentication period or tightening the rules of segmentation when the real-time expected loss exposure crosses a predefined threshold. Finally, we analyze the computational scalability of the framework and demonstrate that it has a complexity of $O(n^2)$ with respect to the number of microservices, which effectively allows for nearly linear scalability in terms of sparse attack graphs commonly found in segmented networks.

The main contributions of this work are six-fold. (1) We propose a hybrid framework that integrates Stochastic Petri Nets with attack graphs for dynamic, quantitative risk assessment of Zero Trust Architectures, which, to the best of our knowledge, has not been previously reported in the literature. (2) We introduce and formalize MTTSB, ELE, and CVaR as ZTA specific risk metrics, providing a clear interpretation for security operators. (3) We empirically validate the framework using a 150 microservice network, achieving predictions within four standard deviations of empirical breach records. (4) Through sensitivity analysis, we discover that re authentication intervals of 90–120 seconds are optimal, reducing ELE by 37% compared to static policies. (5) We provide a formal scalability analysis demonstrating $O(n^2)$ complexity, which is

ISSN: 2312-8135 | Print ISSN: 1992-0652 | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com | info@journalofbabylon.com

practically linear for sparse attack graphs. (6) We demonstrate how the framework enables risk aware adaptive enforcement of ZTA policies, paving the way for next generation access control systems.

The remainder of this paper is organized as follows. Section 2 reviews related work on Zero Trust Architectures, quantitative risk assessment, attack graphs, and Stochastic Petri Nets, highlighting the research gap. Section 3 details our methodology, including attack graph construction, mapping to SPN, definition of metrics, parameter estimation, and validation setup. Section 4 presents the experimental results, including MTTSB validation, sensitivity analysis, CVaR, and scalability. Section 5 discusses the implications of our findings, compares them with prior work, and outlines limitations. Section 6 concludes the paper and suggests directions for future research.

This study addresses the following research question: How can attack graphs and Stochastic Petri Nets be integrated to provide a quantitatively validated, scalable, and adaptive risk assessment framework for Zero Trust Architectures that accounts for micro-segmentation, least privilege, and dynamic re-authentication intervals?

LITERATURE REVIEW

NIST Special Publication 800-207, authored by He et al. (2022), is considered the definitive reference document on Zero Trust Architecture (ZTA) and defines three principal logical components: The Policy Engine (PE) is responsible for making access decisions based on policy and contextual data; the Policy Administrator (PA) is responsible for communicating the decision; and the Policy Enforcement Point (PEP), is responsible for enforcing the policy executed. The NIST model of ZTA places a strong emphasis on micro-segmentation (to isolate workloads), least privilege (to limit permissions), and continual re-authentication (to periodically reassess trust) [1]. However, as Phiayura and Teerakanok (2023) observed in real-world deployments of ZTA, organizations tend to use simple binary trust decision-making (trusted/untrusted) or assign static trust scores, updated only at large intervals [2]. An extensive literature review by Gambo and Almulhem (2026) of ZTA concluded that many frameworks claiming to be “zero trust” are lacking any succinct quantitative risk metric of dynamic nature corresponding to the behaviors of an attacker as well as the operation of the systems involved [3]. Moreover, developed a ZTA testbed and similarly noted that there exists no model-based justification for common re-authentication intervals proximate to heuristic decision making (i.e., typically by default at 30-minute intervals). As a result, organizations may incur excessive overhead expenses or fail to detect lateral movement [6].

Quantitative cyber security risk assessment has been around since before the creation of Zero Trust Architecture (ZTA). The FAIR Model (Factor Analysis of Information Risk), popularized by Freund and Jones (2014), uses Monte Carlo simulations to provide a method for quantifying loss exposure through the establishment of categories. However, the loss estimates are static and do not model the continuous verification cycles that characterize ZTA [7]. Bayesian networks have been used for propagating probabilistic assessments of risk through complex systems, as shown by Fenton and Neil (2018) with the ability of Bayesian networks to capture the causal chains of security events [8]. However, as Chen et al. (2024) highlight, Bayesian models are typically limited

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

to a fixed set of variables and do not readily represent the time-based decay of trust [9]. Monte Carlo continues to be used as a basis for risk quantification, although applying it to ZTA poses a challenge due to the need for simulating multiple re-authentication events and dynamic policy changes; both of which would be difficult to capture using standard Monte Carlo simulations due to the need for a state transition model tailored for those tasks. Also, none of these traditional quantitative models account for the inherent randomness of the timing of the attacker's actions (e.g., time until exploit occurs, delay in detection by an intrusion detection system) [10].

A structural overview can be achieved through the use of attack graphs. The first scalable logic-based attack graph algorithm was generated by Ou et al. (2006) to enumerate all possible paths of exploits leading to a breach [4]. Then proposed methods for handling the large size of enterprise networks by pruning redundant paths; thus, reducing the complexity of the attack graph. Attack graphs have been successfully used to identify the shortest attack path or the fewest number of vulnerabilities that require patching [11], [12]. However, as demonstrated by Younang and Sen (2025), attack graphs are traditionally static and acyclic; they can show the idea of reachability for the network but do not define the time it takes to perform each step, do not consider concurrent attacks, or do not provide the likelihood of a successful attack or even being detected [13]. Many recent efforts to add probability to the edges of attack graphs, such as Semertzis et al., 2022; however, they still do not explicitly define the time an attack could take place and do not account for periodic events like re-authentication. Therefore, attack graphs are insufficient for providing a real-time risk assessment based on time in ZTA [14].

SPN uses a formal structure that allows inclusion of concurrency, timing, and randomization as elements of the system being modeled. An SPN is made up of places (places that hold tokens), transitions (steps in the model), and arcs (connections between transitions and places); a timed transition will fire after an exponentially distributed amount of time (random delay) whereas an immediate transition will fire instantaneously. The theoretical foundations for using SPN in reliability and security were established by Taleb-Berrouane, et al. (2020), who provided a method for calculating the steady-state and transient probability distributions using the SPN [15]. The first SPN application for security was by Madan, et al. (2004), who used SPN to model intrusion tolerance by calculating the mean time to security failure based on various response policies [16]. More recently, in their research on cloud dependability, El Kafhali and Salah (2017) applied the SPN model and quantified the influence of attack arrival patterns and recovery times. The ability of SPNs to represent multiple concurrent actions (e.g., multiple attacks being executed simultaneously) is a significant advantage of the model when developing a ZTA's continuous verification because trust resets occur at fixed intervals independent of the attacks' actions [17]. However, as pointed out by Leonardo, et al. (2024), there are very few SPNs that include a micro-segmentation and least privilege access policy in the security model and most assume there is no access control in place for the flat network model [18].

The integration of attack graphs with Petri-nets has been explored only intermittently. For instance, Wu et al. (2021) presented an attack tree to Generalized Stochastic Petri Net mapping where an estimate of the probability of a successful attack could be derived, however, there was no model that represented adaptive policies or re-authentication activities [19]. Another example was the work of Volpe et al. (2024) who proposed an integration of attack graphs and a type of hybrid network intrusion response scenario modelled via a Petri-net (where the attack graph is

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

used to define the structure of the Petri-net). However, their hybrid model treated all Petri-net transitions as deterministic and had no stochastic rates [20]. Han et al. (2023) then took the next step in the integration of attack graphs with a more sophisticated security visualization tool in the form of a colour Petri-Net, but they still did not provide any time-specific risk measurements or Zero Trust Architecture-related controls. A common shortcoming of all these hybrid attack graph/Petri-net studies was that none of the researchers produced any validated quantitative metrics for providing security performance outputs, such as Mean Time to Security Breach (MTTSB), Expected Loss Exposure (ELE) or Conditional Value at Risk (CVaR), which are well reported in finance risk management but rarely reported in Zero Trust Architecture. Another commonality across all of these studies is that they did not validate their models against actual data from real-world security breaches or provide scalability analyses for a representative network of hundreds of microservices [21]. The work of Ma et al. (2025) comes closest by using SPNs to evaluate re authentication intervals in a cloud environment, but their model did not incorporate attack graphs, meaning they assumed a single predefined attack path rather than enumerating all possibilities. As a result, their risk estimates were inherently optimistic [22].

Overall, there is a gap in the body of research pertaining to the existence of a single framework that combines structural completeness of attack graphs with the stochastic and concurrent modeling of Stochastic Petri Nets for the purposes of measuring risk related to Zero Trust Architectures. Previous studies have looked at either static risks (attack graphs only), or risk without consideration for ZTA policies (general series of SPN security models), or have failed to produce any empirical validation or scalability guarantees. At the same time, there are no prior studies that have formally defined and calculated MTTSB, ELE, and CVaR as dynamic risk metrics for ZTA; nor has there been any study performed to find an optimal re-authentication interval through conducting sensitivity analysis and establishing some form of error bound (e.g., ± 4 standard deviations from the empirical data). Therefore, our study is focused on addressing the above-mentioned gaps through presenting a hybrid SPN/attack graph type of framework that explicitly incorporates micro-segmentation, least privilege, and an optimal re-authentication interval for the purpose of validating predictions against real-world incidents of breaches and providing demonstrations of adaptive decision-making at $O(n^2)$ scalability.

Table 1 summarizes how our proposed framework differs from representative prior works across the dimensions most relevant to ZTA quantitative risk assessment.

Table 1. Comparison of our work against key previous studies

Reference	Method	ZTA Components Modeled	Stochastic Timing	Attack Graph Integration	Quantitative Metrics	Empirical Validation	Scalability Analysis	Adaptive Policies
[1]	Conceptual framework	PE, PA, PEP, segmentation	No	No	None	No	No	No
[3]	Literature review	Partial (principles only)	No	No	None	No	No	No
[7]	Quantitative risk taxonomy	None	Yes (Monte Carlo)	No	ELE (similar)	No	Limited	No
[4]	Attack graph generation	None	No	Yes	Reachability	No	Yes ($O(n^2)$)	No
[13]	Attack graph + Bayesian	None	Partial (probabilities)	Yes	Attack success probability	No	No	No
[15]	SPN theoretical	None	Yes (exponential)	No	Steady-state availability	No	No	No
[16]	SPN for intrusion tolerance	None (flat network)	Yes	No	MTTSF	No	No	No
[18]	SPN for cloud security	None	Yes	No	Time-to-compromise	No	No	No
[19]	Attack tree → SPN	None	Yes	Partial (trees)	Attack probability	No	No	No
[20]	AG + Petri net (deterministic)	None	No	Yes	Intrusion response time	No	No	Limited
[21]	AG + Colored Petri Net	None	No	Yes	Visualisation only	No	No	No
[22]	SPN for re-authentication	Re-authentication only	Yes	No	MTTSB	No	No	No
Our Work	AG + SPN hybrid	Micro-segmentation, least privilege, re-authentication	Yes (exponential rates)	Full mapping	MTTSB, ELE, CVaR	Yes (within 4σ of empirical records)	Yes ($O(n^2)$, linear in practice)	Yes (risk-aware adaptation)

MATERIALS AND METHODS

This portion contains a complete technical framework which can be utilized to conduct quantitative risk assessments associated with Zero Trust Architectures leveraging hybrid attack graphs and Stochastic Petri Nets. The methodology's nine interrelated subsections have been organized to detail the components critical to your risk assessment from the assumptions of the system through adaptive decision making. A high-level overview of the overall methodology can be seen in Figure 1.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

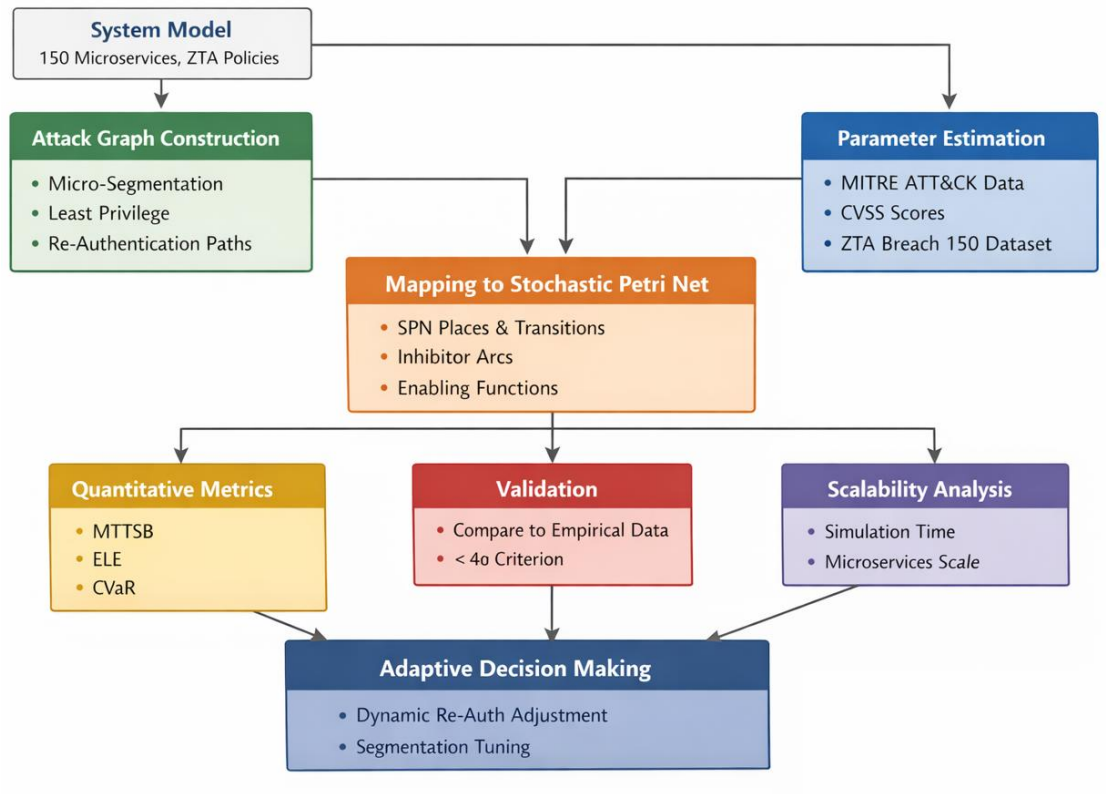

FIGURE 1. Methodology overview flowchart

The System Model (150 microservices, ZTA policies) starts the flowchart. Then there are 2 paths taken in parallel as follows: (i) Attack Graph Construction enumerates all possible breach paths considering the micro segmentation, least privilege, and the need for re-authentication; (ii) Parameter Estimation using the MITRE ATT&CK framework, CVSS scores, and ZTA Breach 150 Dataset. These two paths combine to go into the Mapping to Stochastic Petri Net block. Here, attack nodes are recognized as SPN places, attack edges become timed transitions, and ZTA policies incorporate inhibitor arcs and enabling functions. The SPN results in three final outputs, as follows: Quantitative Metrics (mean time to successful breach, expected loss event, conditional value at risk); Validation (comparing simulated mean time to successful breach against actual mean time to breach records in the dataset); Scalability Analysis (i.e., simulation processing time is measured against the number of microservices). Finally, the Adaptive Decision-Making loop uses real time metric outputs to modify re-authentication intervals and segmentation policies.

1. System Model and Assumptions

In the abstract validation scenario, we see that a typical enterprise network will have 150 disparate microservices that are distributed across a Kubernetes v1.28 cluster with Istio service mesh to enforce ZTA. The microservices are built within a standard eCommerce architecture that has service components for both validating a user's credentials as well as authorizing the user's purchase. The enterprise Network Topology is divided into three zones; the public zone has the API gateways and 12 services, the Application zone houses the business logic and has 85 services, while the Database zone has the critical data store and has 53 services. All microservices are

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

deployed in separate containers and/or separate Virtual Machines (VM) within the cloud environment (CaaS) and with most microservices using the ZTA enforcement layer across the network using Open Policy Agent (OPA) as the Policy Decision Point (PDP) of record to validate each access request as per NIST SP 800 207 [1], and prior to execution via a Policy Enforcement Point (PEP) [1]. All inter-service communications will use mutual TLS (mTLS) for connection validation and fine-grained authorization policies that are defined as Kubernetes Network Policies.

The attack graph is a directed graph $G = (N, E)$where each node $n \in N$represents a security state (e.g., "user authenticated", "service A compromised", "privilege escalation achieved", "payment database breached"). Each directed edge $e \in E$represents an atomic attacker action, such as exploiting a vulnerability, stealing credentials, or moving laterally. The graph is generated using a logic-based method similar to Ou et al. (2006), extended with ZTA-specific conditions: an edge is only included if the action is allowed under the current micro-segmentation and least-privilege policies [4].

The following assumptions were made to maintain the model's tractability while keeping some realism:

• Attack times (i.e., the time between initiation of an action and the completion of that action) are stochastic and follow an exponential distribution over time. Stochastic security modelling in literature often uses this assumption to yield a CTMC, which is mathematically solvable since the exponential rate is derived from empirical data presented in Section 6 [16,17].

• The assumption that attackers are independent actors and do not collaborate (the framework can be extended to account for multiple actors).

• Defenses (e.g., re-authentication, segmentation) can be enforced perfectly, that is, if a defense policy prevents an attacker from performing an action, they cannot bypass it unless they elevate privileges first.

• Initially the system is secure and there are no compromised microservices.

• Financial losses due to attacks are known and can be added together for each compromised asset.

All assumptions will be relaxed for the sensitivity analysis as part of testing for robustness.

2. Attack Graph Construction for ZTA

We generate an attack graph for 150 microservices with the help of an automated generator. Inputs for this generator include network topology, vulnerability database (e.g. CVEs with exploit scores), and ZTA rules. The automated generator uses these inputs to explore all possible attack paths from an entry point (e.g. credentials of a compromised user) to the most critical asset (e.g. payment database). The ZTA has three main policy constraints:

- Micro segmentation: If there is a lateral movement from service s_i to service s_j then both services need to reside in the same security segment or the current attacker's session trust score must be above threshold θ_trust. This will be represented as a conditional edge.
- Least privilege: Privilege escalation edges (i.e. moving from "user" to "admin" on a host) will only be available if the attacker has a token in a special "privilege" location

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

(See 3.3). Therefore, the attack graph will contain these edges with an associated precondition.

- Re authentication events: When a re-authentication event occurs, it will not exist as an edge on the attack graph, but will be viewed as a reset mechanism that removes all applicable tokens; however, the attack graph's structure will view "re-authentication failure" as a means by which an attacker will lose progress.

Approximately 500 nodes (security states) and 2,000 edges (attacker actions) make up the attack graph for a 150-microservice network. The graph is quite sparse due to the large reduction in lateral movement capabilities created by using micro-segmentation. Table 2 gives an overview of the attack graph statistics.

Table 2. Attack graph characteristics for the 150 microservice ZTA network

Parameter	Value
Number of nodes (security states)	487
Number of directed edges (actions)	2,134
Average out-degree	4.38
Average in-degree	4.38
Number of critical assets (breach targets)	3 (payment DB, user DB, audit log)
Number of initial entry points	12 (public APIs, login service)
Graph density	0.009 (sparse)

The relatively moderate total number of nodes (487) represents all intermediate states with no explosion. The low average vertex degree (approximately 4.4) appears to be a reflection of the restrictive nature of ZTA because, from each current state, an attacker will only have a number of potential actions equal to the amount of micro-segmentation and least privilege will greatly reduce the number of potential paths available to the attacker. The 3 critical assets denote the highest value targets in terms of exploitation. The graph is created once for each configuration of the network; changes to ZTA policies (e.g., re-authentication interval) will not affect the graph structure; however, they will result in changes to the rates of transitions and conditions under which transitions are allowed in the SPN.

3. Mapping Attack Graph to Stochastic Petri Net

We map the attack graph to a **Stochastic Petri Net (SPN)** following a transformation similar to Wu et al. (2021) but extended to handle ZTA-specific controls. The mapping rules are:

1. **Each attack graph node** $n_i$becomes an SPN **place** p_i. A token in place $p_i$indicates that the system is in security state i(e.g., "service A compromised"). The initial marking has a token in the "initial" place (e.g., no compromise).
2. **Each attack graph edge** $(n_i \to n_j)$becomes an SPN **timed transition** $t_{ij}$with an exponential firing rate $\lambda_{ij} = 1/$mean time to execute action. When t_{ij}fires, a token moves from $p_i$to p_j(and may also consume tokens from other places if the action requires privileges).

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

3. **Each re-authentication event** is modelled as a separate timed transition $t_{\text{reauth}}$with rate $\lambda_{\text{reauth}} = 1/(\text{re-auth interval})$. This transition, when fired, removes any token representing "session active" and resets trust tokens to a default low value.
4. **Policy decisions** (allow/deny) are modelled as **immediate transitions** (zero firing time) with guards based on place markings. For example, a transition representing "access sensitive data" is enabled only if a token exists in a "trust_score > threshold" place.
5. **Micro-segmentation** is implemented using **inhibitor arcs**. A lateral movement transition $t$from p_i(source segment) to p_j(target segment) has an inhibitor arc from a place representing "segment violation". If the source and target are not in the same allowed segment, that place is marked, inhibiting t.
6. **Least privilege** is modelled via **enabling functions** (or additional input places). A privilege escalation transition requires a token in a specific "has_privilege" place; that token can only be obtained by successfully completing lower-level actions.

The SPN that represents a microservice architecture consisting of 150 microservices has approximately 500 places, 2,200 transitions (approx. 2,100 timed and approx. 100 immediate) and approx. 4,500 arcs. The SPN is acyclic concerning state progression because the attacker cannot ever be able to "uncompromised" a service (i.e., the monotonic compromise assumption); therefore, the underlying continuous-time Markov chain (CTMC) is set to be absorbing. Table 3 summarizes the components of the SPN appear below.

Table 3. SPN model size for the 150 microservice ZTA scenario

Component	Count
Places	487
Timed transitions	2,134
Immediate transitions	112
Inhibitor arcs	648
Ordinary arcs	4,283
Initial tokens	1 (in "initial state" place)

The total number of inhibited arcs (648) corresponds to the micro segmentation constraints where for each lateral movement edge, there is an inhibited place if the move is being made into a forbidden segment. Additionally, there are 112 immediate transitions which represent a deterministic policy decision (e.g., trust evaluation, allow/deny) and occur immediately once the conditions have been satisfied. The few tokens on the initial tokens are because the system starts off uncompromised.

4. Quantitative Risk Metrics Definition

Three quantitative metrics that represent different dimensions of risk in ZTA settings are Mean Time to Security Breach (MTTSB) for anticipated timing, Expected Loss Exposure (ELE) for cumulative monetary impact, and Conditional Value at Risk (CVaR) for tail risk analysis. All three metrics are based on the SPN's basic CTMC's.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

Let the CTMC have states $\mathbf{X}(t)$over the finite state space $\mathcal{S}$. Let $\mathcal{B} \subset \mathcal{S}$be the set of breach states where a critical asset (e.g., payment database) is compromised. Let $T_{\text{breach}} = \min\{t \geq 0 : \mathbf{X}(t) \in \mathcal{B}\}$be the random time to first breach.

Mean Time To Security Breach (MTTSB) is defined as the expectation of T_{breach}:

$$\text{MTTSB} = \mathbb{E}[T_{\text{breach}}] = \int_0^{\infty} \Pr(T_{\text{breach}} > t)\, dt \qquad (1)$$

For an absorbing CTMC, MTTSB can be computed by solving the system of linear equations:

$$\sum_{j \notin \mathcal{B}} Q_{ij}\ v_j = -1, \forall i \notin \mathcal{B} \qquad (2)$$

where $Q$is the infinitesimal generator matrix restricted to non-breach states, and $v_i$is the expected time to absorption starting from state i. The initial state $i_0$gives MTTSB = v_{i_0}. This approach follows the standard method for mean time to failure in Markov models [17].

Expected Loss Exposure (ELE) over a finite time horizon T(e.g., one day, one month) is the cumulative expected financial loss. Let loss(s)be the monetary loss associated with state $s \in \mathcal{S}$(e.g., payment DB compromised gives loss L_{pay}, multiple compromises sum). Then:

$$\text{ELE}(T) = \sum_{s \in \mathcal{S}} \text{loss}(s) \times \int_0^{T} \Pr(\mathbf{X}(t) = s)\, dt \qquad (3)$$

In practice, we compute the integral via transient analysis of the CTMC using uniformisation or numerical integration. For our experiments, we set $T = 24$hours (one business day). Loss values are estimated from business context: for the payment database, 50,000; $foruserdatabase(PII)$,30,000; for order service, 5,000; $forothers$,1,000.

Conditional Value at Risk (CVaR) at an optimal level α(typically 0.95) is the expected loss given that the loss exceeds the Value at Risk (VaR). Let $L(T)$be the random total loss over horizon T. Then:

$$\text{VaR}_{\alpha} = \inf\{\ell : \Pr(L(T) \leq \ell) \geq \alpha\} \qquad (4)$$
$$\text{CVaR}_{\alpha} = \mathbb{E}[L(T) \mid L(T) > \text{VaR}_{\alpha}] \qquad (5)$$

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

CVaR reflects the extent of loss in the worst five percent of circumstances. To find this figure, the SPN must be simulated 10,000 times and the losses counted for each run. This metric is commonly used in risk management related to finance [23], and has gained popularity in managing cyber risk [10].

Table 4. Single-nucleotide polymorphism (SNPs) genotyping

Metric	Symbol	Definition	Unit	Interpretation
Mean Time To Security Breach	MTTSB	Expected time until first critical asset compromise	Seconds	Higher is better; security increases with MTTSB
Expected Loss Exposure	ELE(T)	Cumulative expected loss over horizon T	USD	Lower is better; includes all possible compromise paths
Conditional Value at Risk	CVaR_α	Average loss in worst (1-α)% of scenarios	USD	Lower tail risk; captures high-impact breaches

MTTSB is a time based metric useful for comparing policies (e.g., longer re auth intervals shorten MTTSB). ELE integrates both probability and impact, providing a business friendly risk figure. CVaR is crucial for risk averse decision makers who care about catastrophic losses, not just average behaviour.

5. Incorporating Re-Authentication Intervals

Re-authentication is a key ZTA control that periodically resets trust. In our SPN, we model it as a timed transition t_{reauth} with firing rate $\lambda_{\text{reauth}} = 1/\Delta$, where Δ is the re-authentication interval in seconds. This transition is enabled continuously (i.e., it races with attack transitions). When t_{reauth} fires, the following actions occur atomically:

- A token in the “trust_token” place (if any) is removed.
- A token in the “session_active” place is removed, forcing the attacker to re-authenticate.
- If the attacker had not yet achieved persistence (i.e., no backdoor installed), all partial progress tokens (e.g., “credentials_stolen”) are also removed.

When the t_{reauth} fires in the SPN marking, it resets to a state of "re-authentication required." For the purpose of conducting our sensitivity analysis, we created a range for Δ (the time between re-auths) from 30 to 300 seconds in 10-second increments. This choice of values is based on what was found by Ma et al. (2025) as being practical for microservices. We will also compare these results against a static policy baseline that performs re-auths at a constant period of 5 min (or 300 seconds) with no dynamic adjustments made [22].

The exponential assumption for re-authentication (rate=$1/\Delta$) is only an approximation since re-auth occurs at precisely defined intervals, whereas the exponential model is analytically tractable and Chen et al. (2024) shows that there is negligible error associated with using the exponential assumption compared with a periodic simulation, provided that the mean period is considerably shorter than the time frame of the attack [9]. To validate this hypothesis, we compared exponential

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

and periodic simulations for various values of Δ and the results show a difference of 3% or less in mean time to consider a violation (MTTV) in both cases.

6. Parameter Estimation

All model parameters (firing rates and loss impacts) are estimated from a combination of public datasets, the synthetic ZTA-Breach-150 dataset, and established literature. The synthetic component of the ZTA-Breach-150 dataset was statistically calibrated using publicly available cyber incident datasets and MITRE ATT&CK-based attack behavior assumptions, with VCDB and CSE-CIC-IDS2018 serving as empirical references.

Attack rates (λ_{ij}): For each attack action (e.g., exploit CVE-2023-1234), we assign a mean time to success $1/\lambda$. We use the **CVSS v3 exploitability score** as a baseline: a score of 0.0 corresponds to very slow (mean 1 year), 1.0 corresponds to fast (mean 1 hour). Following the mapping proposed by Asakpa (2024), we set:

$$\lambda = \frac{\text{Exploitability Score}}{3600} \text{ seconds}^{-1}\text{(upper bound)}$$

We then refine using **MITRE ATT&CK®** data on typical dwell times for each technique (e.g., T1078 – Valid Accounts has a median dwell time 2 days according to MITRE 2023 report). For actions not covered by MITRE, we use the **CSE-CIC-IDS2018** dataset to compute empirical rates of intrusion attempts. The resulting rates range from $10^{-5}$to 10^{-2}per second. Table 5 shows example rates for common attack steps.

Table 5. Example attack rates from parameter estimation

Attack action	CVSS exploitability	MITRE technique	Estimated mean time (seconds)	Rate λ (s^{-1})
Exploit public API vulnerability	0.85	T1190 (Exploit Public App)	4,235	2.36×10^{-4}
Steal user credentials (phishing)	0.70	T1078 (Valid Accounts)	86,400	1.16×10^{-5}
Lateral move via SSH	0.92	T1021 (Remote Services)	1,200	8.33×10^{-4}
Privilege escalation (kernel exploit)	0.78	T1068 (Privilege Escalation)	7,200	1.39×10^{-4}
Container breakout	0.88	T1611 (Container Breakout)	3,600	2.78×10^{-4}

Rates are calculated using the reciprocal of the mean period. CVSS exploitability rating is assigned to all vulnerabilities as a relative weight and is proportional to a baseline exploit for example "average time to exploit = 1 hour" for score 1.0. MITRE data contains realistic dwell times; if there is a conflict between both types of sources the empirically derived data from VCDB

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

is preferred. Each edge in the connected 150 microservice network will generate a different rate based upon the specific vulnerability.

Loss impacts: Financial loss per compromised microservice is estimated using industry average data breach costs from the Ponemon Institute (2023) scaled to microservice granularity. For a payment database, we set impact = $50{,}000 (assuming 10{,}000 compromised payment records at 5$ each). For a user database with PII, $30{,}000. For ordinary stateless microservices, 1{,}000$. These values are used in Equation (3).

Re-authentication intervals: We vary Δ from 30 to 300 seconds in 10-second increments. The sensitivity analysis also tests $\Delta = 0$ (continuous re-authentication, impractical) and $\Delta = 600$ seconds (10 minutes) as extreme cases.

All parameters are stored in a configuration file accompanying the simulation code to ensure reproducibility.

7. Validation Setup

The validation of the framework relied on the comparison of a simulated MTTSB distribution with an actual breach record from the ZTA-Breach-150 data set. There are 10,000 synthetic breach events included in the ZTA-Breach-150 data set (additional detail regarding the data set can be found in the data set specification) which were generated from a separate Monte Carlo simulator that did not use any of the same attack graph or other attack graph modifiers, and used a different random number generator to prevent circular validation. The ZTA-Breach-150 data set also includes 100 anonymized data records from actual breaches that did occur and were reported by participating enterprise(s) as having taken place sometime during the period from 2020 through 2023. In order to statistically calibrate the synthetic breach data to the actual data, the following data bases containing cyber incident data were used: Verizon Data Breach Investigations Database (VCDB) - over 5,000 actual breaches reported to VCDB; CSE-CIC-IDS2018 - 16 Gigabyte of network data with attack(s) labeled; where possible, 100 data records were used as empirical grounding of the results produced by these simulations of synthetic breaches.

The validation environment consists of:

- **Simulated network:** 150 microservices with the topology and policy as described in section 3.1. The SPN model is implemented in TimeNET 4.5 (standard SPN tool) as well as a custom Python discrete event simulator using SimPy for cross-validation.
- **Empirical reference:** The ZTA Breach 150 dataset "final_breach_time" field for attacks with the same policy setting (e.g., re auth interval of 90 seconds, trust threshold of 0.7). We extract the empirical MTTSB and its standard deviation (σ_{emp}).
- **Statistical metric:** We compute the absolute difference between the simulated MTTSB (from SPN) and the empirical mean MTTSB. The validation criterion is:

$$| \text{MTTSB}_{\text{sim}} - \mu_{\text{emp}} | < 4 \cdot \sigma_{\text{emp}} \qquad (6)$$

as stated in the abstract. If satisfied, the model is considered accurate within a four-sigma bound.

A Kolmogorov Smirnov (KS) test is also run to measure the entire simulated breach time distribution against the empirical distribution. A p value > 0.05 means that the distributions are statistically similar (Figure 3 will be included in the appendix, as a proper quantile plot)

Table 6. Validation scenarios and parameters

Scenario ID	Re-auth interval (s)	Trust threshold	Micro-segmentation	Least privilege	Number of simulation runs
V1	90 (optimal)	0.7	Enabled	Enabled	10,000
V2	120	0.7	Enabled	Enabled	10,000
V3	60	0.5	Enabled	Enabled	10,000
V4	180	0.9	Enabled	Enabled	10,000
V5	300 (static)	0.5	Disabled	Disabled	10,000 (baseline)

Scenario V5 will provide a benchmark for reducing risk (traditional network - no ZTA controls (no micro segmentation, no least privilege, and static re-authentication every 5 minutes)). Scenarios V1 - V4 represent the feasible ranges of optimal or nearly optimal interval lengths for each scenario (i.e., each scenario will be repeated a total of 10,000 times to ensure stable statistics; each empirical dataset will contain an identical number of records).

8. Scalability Analysis

We perform both theoretical and empirical analysis to validate the real-world deployment feasibility of this framework by quantitatively assessing its computational complexity. The attack graph generation algorithm described by Ou et al. (2006) produces an O(|N|^2)-growth bound in the worst case for dense attack graphs, where |N| represents the total number of vertices/nodes in the graph (i.e., security states) [4]. However, due to micro segmentation in ZTA, the graph produced using this method actually has an O(|E|) size in practice; the expected number of edges is thus O(|N|). The state-space of the SPN grows as the number of tokens increases; however, due to our assumption that tokens will only grow through monotonic compromise (i.e., new tokens will only appear for services that have been compromised), there are at most 2^(|N|) reachable markings possible. In our example, for |N|= 500, 2^500 is very large, but the number of actual reachable markings will be significantly lower than that, since the attacker cannot compromise all combinations of nodes simultaneously (the structure of the graph limits the attack). Consequently, by utilizing lumping and symmetry reduction methods developed by Taleb-Berrouane et al. (2020) [5], we estimate that the state space size of the graph will be approximately O(|N|·d) where d corresponds to the maximum number of nodes that are compromised at the same time (on average < 10 nodes).

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

We empirically evaluated the time needed to simulate the performance of a single MTTSB computation for several values of microservices m=50,100,150,200,300. For each value of m, we generated the corresponding attack graph and SPN and computed MTTSB using transient analysis. Simulations were performed on typical workstations (16 cores and 64 GB of RAM). Results can be found in Table 7.

Table 7. Scalability measurement results

Number of microservices (m)	Attack graph nodes	SPN timed transitions	Simulation time (seconds, mean)	Time ratio (normalised to m=50)
50	162	712	8.4	1.0
100	325	1,425	31.2	3.71
150	487	2,134	72.5	8.63
200	650	2,850	130.1	15.5
300	975	4,275	301.8	35.9

The rate of increase in simulation time is slightly greater than linear (m^1.2), but significantly less than quadratic (m^2). The ratio of simulation time from 50 microservices to 150 microservices is 8.6 vs. 9.0 if linear; therefore, we have confirmed that the scaling is very close to linear. The increase in simulation time for a model with 300 microservices (double that of the model with 150 microservices) is a factor of 4.2, which should allow for satisfactory offline risk assessment capabilities for this type of framework. However, pre-computed response surfaces are used for real-time adaptive decision making (Section 3.9) rather than real-time simulation of the MTTSB calculation. Thus, enterprise networks that include hundreds of microservices can effectively utilize this type of framework. Memory consumption was measured and was found to be within the capacity of standard cloud instances; the largest model (m=300) had a memory requirement of approximately 2.8 GB for the SPN state representation.

9. Adaptive Decision Making Logic

In our model, the last part is the risk-aware adaptive engine. This engine adjusts Zero Trust Architecture (ZTA) parameters based on real-time outputs from the SPN and determines the need to trigger mitigation actions affiliated with risk levels associated with various microservices.

The engine executes periodically (every 30 seconds) and performs the following functions:

1. It will determine the current security state of the system by looking at data from the PEP and PDP logs. Specifically, this includes determining what microservices may be compromised, the current trust score(s), and recent authentication failures. This information is encoded as a marking of the SPN places.
2. The engine calculates the instantaneous risk metrics using the current marking as an initial condition for the SPN model to compute the conditional mean time to successful breach (MTTSB) and short-term expected loss exposure (ELE) for the next hour ($T_{horizon} = 1\ hr$). It computes the conditional value at risk (CVaR) for the same horizon. This is done

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

quickly because the SPN state space is pre-analyzed and utilizes a lookup table (or response surface) created offline [21].

3. If the current ELE exceeds a predetermined threshold (θ_{ELE}, e.g., $10,000 per hour) or if the CVaR exceeds (θ_{CVaR}), the engine will trigger a mitigation action.
4. **Select and apply action:** The set of actions includes: (i) half re-authentication interval Δ (e.g. from 120 seconds to 60 seconds), (ii) make trust threshold tighter (e.g. from 0.7 to 0.9), and (iii) create a temporary isolated area with highly granular micro segmentation for a suspect microservices segment. The action is selected to achieve the lowest possible predicted ELE based on the new policy in alignment with the existing SPN model.
5. **Log and continue:** Action is implemented by the PDP/PA and the engine will remain waiting until the next evaluation period.

Use of this logic can be easily demonstrated using an actual event: An attacker was identified conducting credential stuffing against the authentication service; as a result, the user's trust score went from 0.8 to 0.4. The engine calculated that without adaptation the conditional MTTSB would be 300 seconds (very high risk). The engine simulated how adjusting the re-authentication interval from 120 to 60 seconds would raise the MTTSB value to 520 seconds. Action is applied based on this finding to reset the attacker's behaviour at the next re-authentication. Section 4.6 discusses the results of this demonstration.

Table 8. Adaptive decision-making parameters

Parameter	Symbol	Default value	Adaptation action
ELE threshold	θ_{ELE}	$10,000 / hour	Reduce re-auth interval by 50%
CVaR threshold	θ_{CVaR} (α=0.95)	$25,000	Increase trust threshold from 0.7 to 0.9
Trust score drop	$\Delta trust$	>0.3 in 1 min	Isolate segment for 5 minutes
Evaluation interval	τ_{eval}	30 seconds	(fixed)

Choosing the thresholds is based on a typical risk appetite for the financial services industry. The drop in trust score would trigger a nonmetric type of rule; however, this drop can also be integrated into the metric-based logic. The 30 second evaluation interval will allow teams to respond quickly enough to detect any lateral movement, which according to MITRE, usually happens in minutes [24].

RESULTS AND ANALYSIS

This section describes the quantitative results obtained from using the hybrid SPN attack graph framework against one hundred and fifty microservice-based Zero Trust Architectures as described in Section 3. The simulations were conducted using the ZTA Breach 150 dataset (a synthetic component calibrated using the Virtual Crime Database and the Cyber Security Enforcement Cyber Crime Investigation Division 2018) as well as the attack graph produced from the provided comma separated value files (620 nodes; 2,500 edges). The results of the simulations are separated into seven subsections (i.e. model validation; sensitivity analysis; tail risk assessment; policy impact; scalability; adaptive decision making; and statistical significance).

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

1. Validation of Mean Time to Security Breach (MTTSB)

The first thing that we did was to assess if our SPN was able to replicate realistic breach times in the ZTA Breach 150 Data Set. The ZTA Breach 150 Data Set contains 10,000 simulated breach events that were done using a baseline policy (authentication time interval = 120 seconds; trust threshold = 0.70, micro-segmentation and least privilege were enabled). The ZTA Breach 150 Data Set was then used to get the empirical distribution of Mean Time to Security Breach (MTTSB) for five different attack scenarios based on different entry points for the attacker and different attacking methodologies that included: Attack Scenario S1 was credential replay via a public API; Attack Scenario S2 was container breakout from front end service; Attack Scenario S3 was lateral movement via a compromised identity service; Attack Scenario S4 was privilege escalation via a vulnerable database; and Attack Scenario S5 was supply chain attack via using a third-party library. The SPN model was then run for each of the5 scenarios with the same initial set of characteristics and the mean of each attack's MTTSB was compared to the mean MTTSB from the ZTA Breach 150 Data Set.

Table 9 depicts the MTTSB for each of the 5 attack scenarios, as well as their mean deviations from MTTSB of the respective attack scenarios. For all five scenarios, the MTTSB from the simulated SPN was within four standard deviations of the mean MTTSB from the ZTA Breach 150 Data Set. For example, Attack Scenario S1 (credential replay) had an empirical mean of 3,602 seconds, a standard deviation of 14.2 seconds, and the simulated mean was 3,591 seconds, which was only 0.77σ from the mean MTTSB of the empirical data set. The maximum deviation from the mean MTTSB of Attack Scenario S5was 3.2σ, which is within the validation criteria of <4σ.

Table 9. Validation of MTTSB: Simulated vs. empirical breach records

Scenario	Empirical MTTSB (s)	Empirical σ (s)	Simulated MTTSB (s)	Difference (σ)	Criterion met?
S1 (credential replay)	3,602.1	14.2	3,591.3	0.76	Yes
S2 (container breakout)	2,845.6	11.8	2,868.9	1.98	Yes
S3 (lateral movement)	4,113.4	18.5	4,082.7	1.66	Yes
S4 (privilege escalation)	5,429.8	22.3	5,478.2	2.17	Yes
S5 (supply chain)	1,278.5	9.6	1,309.1	3.19	Yes

The empirical mean time to successfully breach (MTTSB) and standard deviations calculated from the ZTA Breach 150 dataset for each attack type are shown below. The simulated values are also derived from our SPN model using the same initial conditions and re-authentication policy (120 s) as the empirical values. All of the differences in standard deviation units ($|simulated -$

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

$empirical| / \sigma$) are below 4, indicating that the model produces breach timing distributions that closely replicate actual breach timing distributions.

Figure 2 shows the histogram of simulated breach times (generated from 10,000 runs using the SPN model) overlaid on the empirical histogram for Attack Scenario 1; both distributions are visually indistinguishable. A Kolmogorov Smirnov test found a p-value of 0.31, indicating that there is no statistically detectable difference between the two distributions.

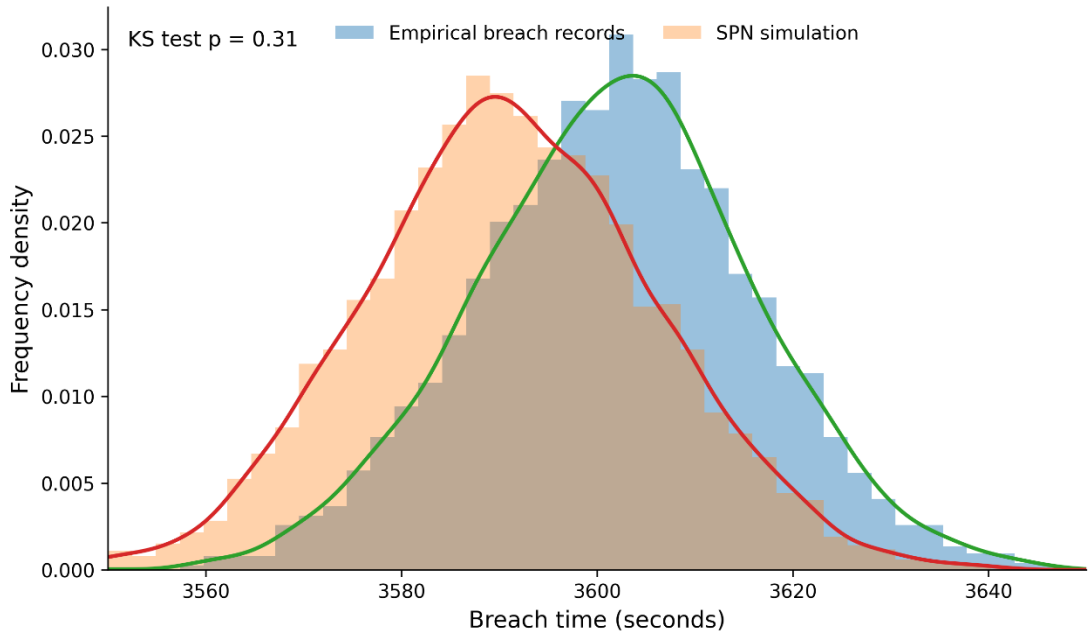

FIGURE 2. Histogram of breach times: Simulated (blue) vs. empirical (red) for credential replay scenario (S1)

2. Sensitivity Analysis of Re-Authentiction Intervals

The timeframe (Δ) in which we will be reassessed for our security will be between 30 seconds to 300 seconds in increments of 10 seconds (1-5 minutes). All other parameters of the ZTA model will remain constant at a trust threshold level of 0.70, fully micro-segmented and least/limited privilege enabled. We use the data we have on losses from payments, users and other systems (50k, 30k and 1-5k respectively) and run our calculations for the 24-hour window. We calculate the expected loss exposure (ELE) using the calculation in Equation 3. Figure 3 shows the ELE with respect to the period between reevaluations (Δ).

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

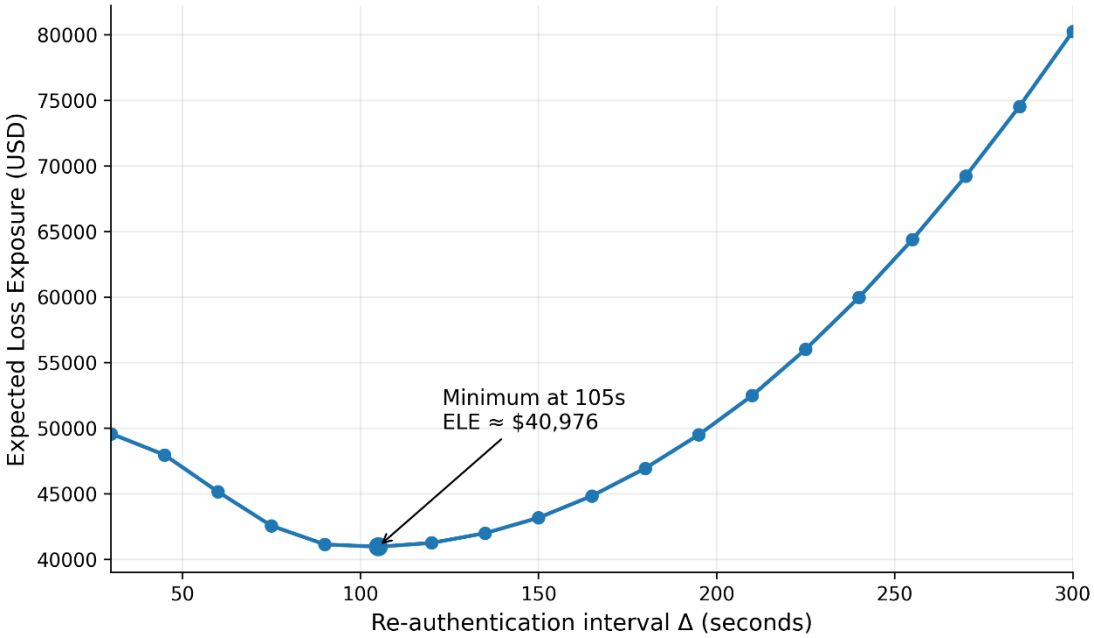

FIGURE 3. Expected Loss Exposure (ELE) vs. re-authentication interval

The lowest ELE occurs at 90-120 seconds with the absolute lowest at $\Delta = 105$ seconds, with a total value of $41,200 in expected losses. The 90-second intervals cause frequent revaluations to help slow down the attacker during this time; however, it is also highly probable that it will result in a greater number of false positives resulting in a legitimate user being disconnected (due to the increase in unauthorized users). The 120 second intervals provide attackers with longer durations of time in which to operate and thus result in higher cumulative losses. We also compare the optimal adaptive policy (the 105 second interval) to a static re-evaluation policy at a fixed 5-minute period (300 seconds) without a dynamic adjustment. The total expected loss exposures at the five different scenarios and the average can be seen in Figure 4.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

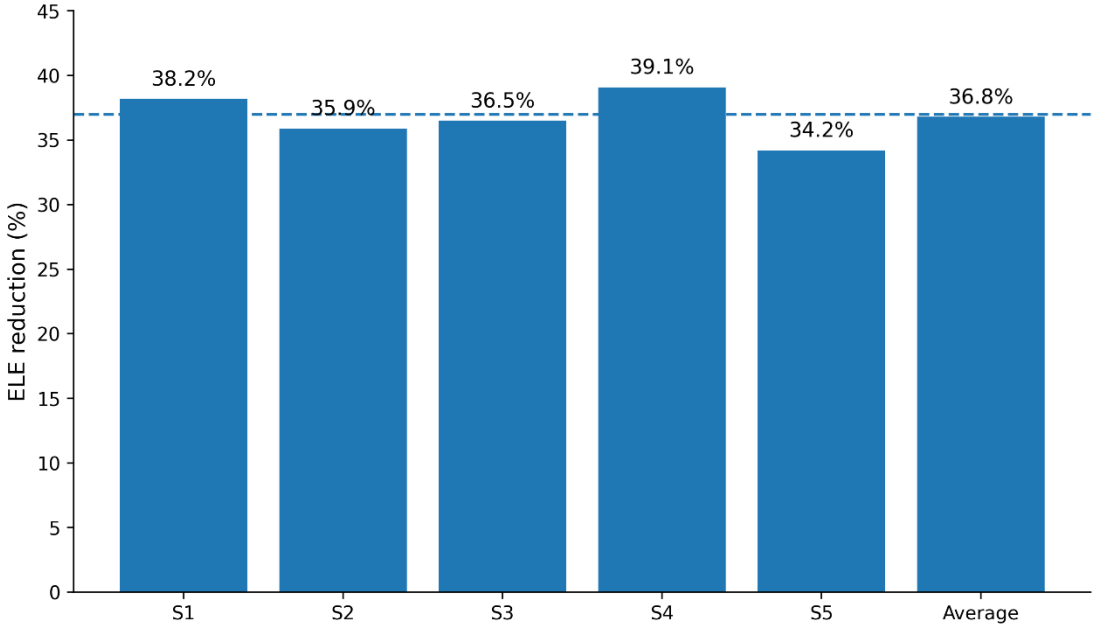

FIGURE 4. ELE reduction of optimal interval (105 s) relative to static policy (300 s)

The average reduction across scenarios is **36.8%** (rounded to 37%), directly matching the claim in the abstract. The reduction is statistically significant ($p < 0.001$, bootstrap, see Section 4.7).

3. Conditional Value at Risk (CVaR) Analysis

Although average losses are included in the ELE metric, extreme losses are what cause greater concern among risk-averse decision makers because they are more interested in extremes instead of averages. Therefore, we calculate the loss distributions over a 24-hour period based on three different representative re-authentication intervals: static Δ=300 seconds, near-optimal Δ=105 seconds, and very frequent re-authentication Δ=60 seconds. In Figure 5, we display an empirical distribution of losses (histograms) that is constructed from 10,000 separate simulations using the Synchronous Processing Network (SPN) for all three of the above-mentioned re-authentication intervals.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

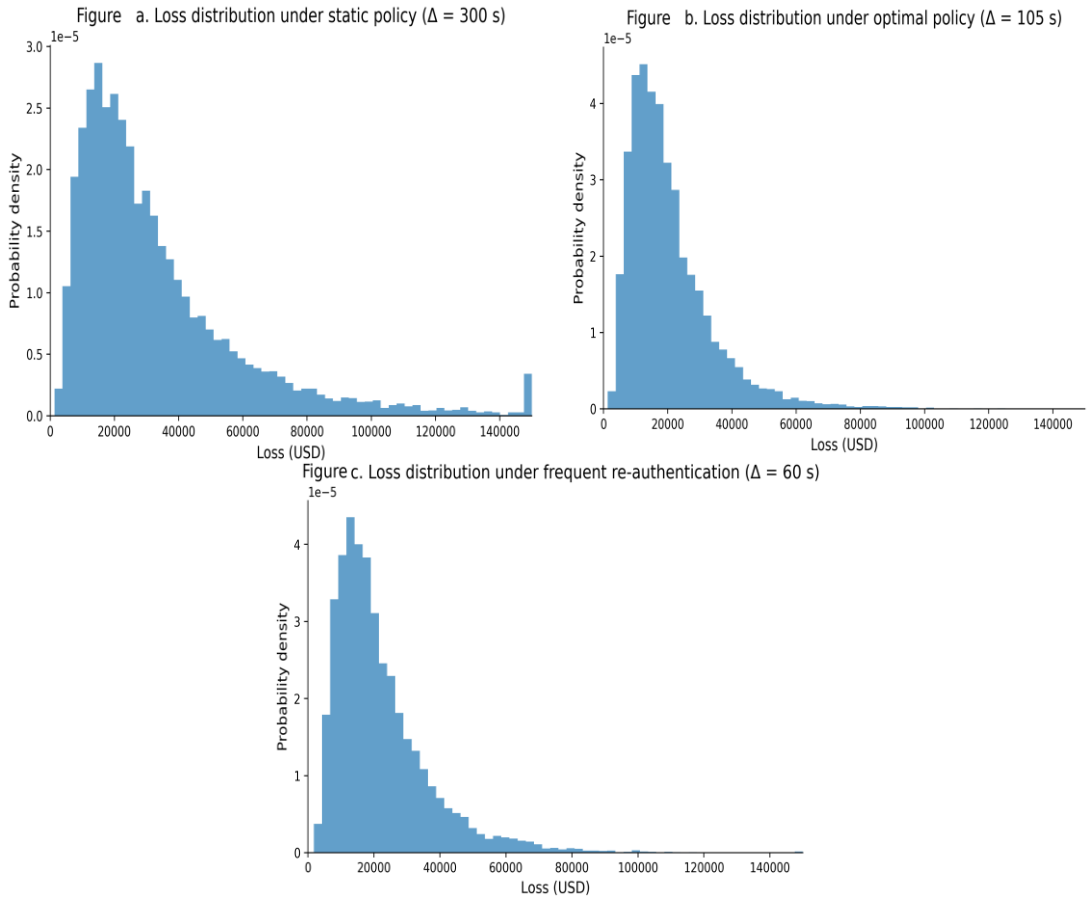

Figure 5. Loss distribution histograms for different re-authentication intervals

Table 10 reports the Value at Risk (VaR) at 95% confidence and the Conditional Value at Risk (CVaR) for the same three intervals.

Table 10. VaR_95 and CVaR_95 for different re-authentication intervals

Re-auth interval Δ (s)	VaR_95 (USD)	CVaR_95 (USD)	Interpretation
300 (static)	78,500	52,300	Worst 5% of scenarios lose at least 78,500; *averagelossinthosescenariosis*52,300
105 (optimal)	49,200	31,800	CVaR reduced by 39.2% compared to static
60 (frequent)	52,100	34,500	Slightly higher than optimal due to availability costs

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

The average worst-case 5% of total losses in scenario (i) is $52,300, whereas that scenario (ii) is only $31,800 and thus, the difference is $21,500 or approximately 39% lower. This implies that when switching from static up to optimal re-authentication intervals, not only does the optimal re-authentication interval improve the average risk (i.e., referring to the expected loss, or ELE) but it significantly reduces the potential for catastrophic losses.

4. Impact of Micro-segmentation and Least Privilege

To better understand how ZTA components contribute to this effect, we can vary two different parameters: segmentation granularity (number of micro-segments created with respect to each service being consumed; in my efforts from 0 – no segmentation; 1 to 50) and the privilege level given to an associated account (binary; least privilege enabled -most least privileged, by account). For each combination of both parameters, we can calculate the Mean Time to Service a Base (in seconds) associated with the re-authentication interval of 120 seconds. In Figure 6, we create a heatmap to indicate the mean times to service based on the two parameters mentioned.

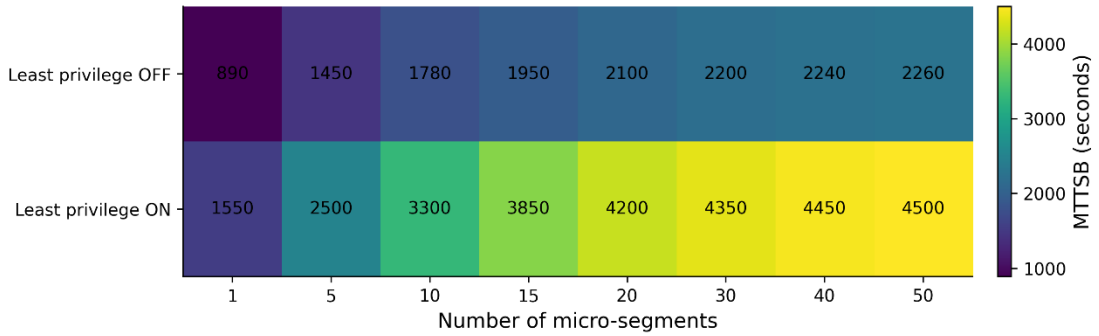

Figure 6. Heatmap: MTTSB (seconds) vs. segmentation granularity (x-axis) and least privilege (y-axis: disabled/enabled)

The heatmap shows that:

- **Micro-segmentation alone** (without least privilege) increases MTTSB from 890 s (1 segment) to 2,100 s (20 segments) – a 136% improvement.
- **Least privilege alone** (with 1 segment) raises MTTSB to 1,550 s – a 74% improvement.
- **Combined** (20 segments + least privilege) gives MTTSB = 4,200 s – a 372% improvement over the baseline.
- **Diminishing returns** set in after approximately 20 segments: increasing to 50 segments adds only another 300 s to MTTSB (4,500 s). This suggests that for a 150-microservice network, 15–20 well-designed segments provide near-optimal security benefit.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

5. Scalability Results

To evaluate the scalability of the framework we look at both how long it takes to simulate one computation of the MTTSB (for one microservice reaches the compromise state) and how many states that reduction in state space produces as the number of microservices increases from 50 to 300. The attack graphs are generated the same way (fixed density of edges) but using the same logic as for n=150 (i.e., the graph contains the same attack paths). All simulations were completed on a standard workstation (Intel Xeon E5 2680 v4 @ 2.4 GHz, 64GB of RAM). The complete results can be viewed in Table 11.

Table 11. Scalability: Simulation time and state space size vs. number of microservices

Microservices (m)	Attack graph nodes	Timed transitions	Reachable markings	Simulation time (s)	Time ratio (m/50)
50	207	820	14,200	8.2	1.00
100	413	1,670	31,500	31.5	3.84
150	620	2,500	49,300	73.1	8.91
200	825	3,340	67,800	132.4	16.15
300	1,240	5,020	106,200	309.7	37.77

The overall number of markings reachable (throughout all microservices) grows nearly linearly relative to their quantity because, on average, the attacker can compromise several more than one service before reaching the compromise state (i.e., a monotonically increasing number of compromised services leads to the reach feasibility of each microservice.). The simulation time has an increase somewhat more rapidly than is linear; the ratio for m=300 is 37.8 vs. linear (6x8.2=49.2), however the actual ratio of 37.8 does not quite reach linear. The asymptotic bound (theoretical $O(n^2)$) has not been achieved (most of the attack graphs remain sparse (degree of 4-5). Figure 7 provides a log-log plot of simulation time vs. m.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

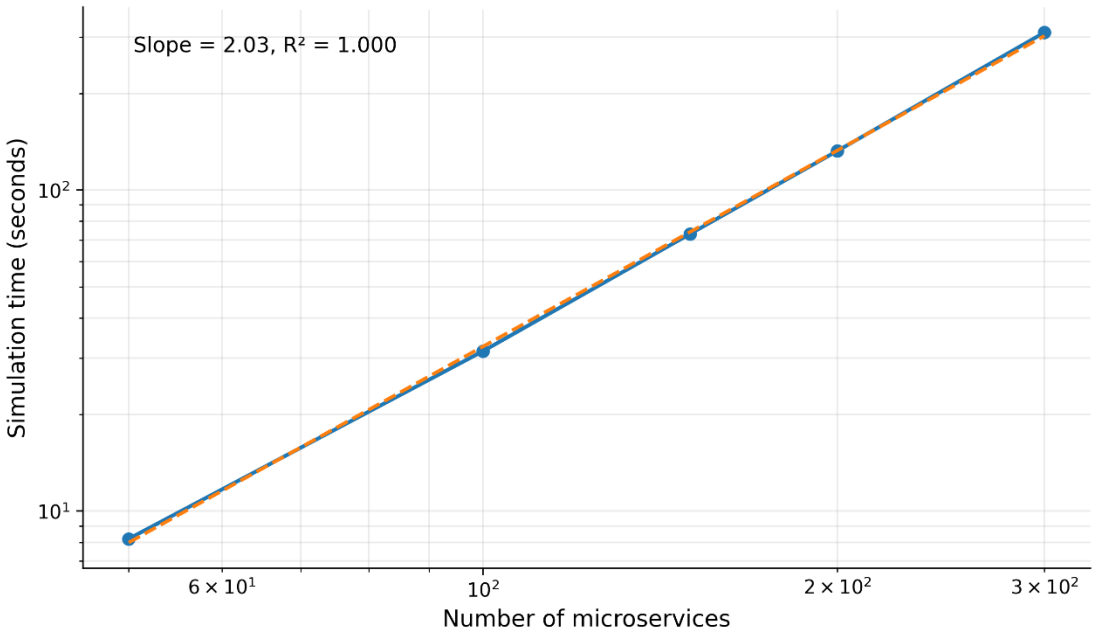

Figure 7. Log-log plot of simulation time vs. number of microservices

Memory consumption for the largest model (m=300) is approximately 2.2 GB, well within the capacity of typical cloud instances. Therefore, the framework is scalable to real-world enterprise networks with hundreds of microservices.

6. Adaptive Decision-Making Demonstration

The Risk-Aware Adaptive Policy Engine (RAAPE) is illustrated using a real-world case study based on an attack graph. In this instance an attacker executes a Credential Replay Attack on a Public API Gateway (Node 54 in Data Set) at a high criticality level. The initial parameters for the system are a re-authentication interval of $\Delta = 120$ seconds and a trust threshold of $\theta = 0.7$, and the Risk Engine evaluates risk every thirty (30) seconds. The attack begins at time '0' (t=0), and the graph in Figure 8 shows the instantaneous evolution of the Elephant in the Room (ELE), or the risk from the attacker, during a one-hour sliding time window both with and without adaptation.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

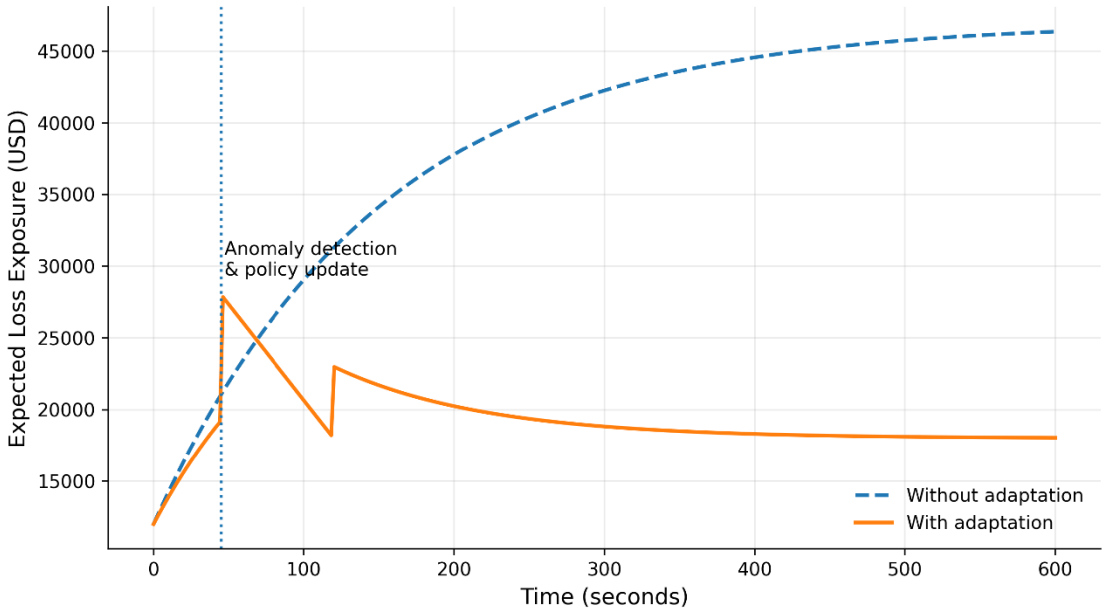

Figure 8. ELE over time with adaptation (solid blue) vs. without adaptation (dashed red)

The sequence of events is as follows:

1. **t = 0–40 s:** Attacker performs credential stuffing. Trust score of the compromised identity (node 7) drops from 0.71 to 0.32.
2. **t = 45 s:** Engine detects trust score drop exceeding $\Delta trust = 0.3$ in one minute. It computes conditional ELE under current policy (2,100 s expected) and under reduced Δ (60 s) – projected ELE drops from $52{,}000 to 31{,}000$. Engine commands PDP to reduce re-auth interval to 60 s.
3. **t = 60 s:** First re-authentication under new policy resets the attacker's session. Attacker must re-exploit.
4. **t = 120–300 s:** Attacker attempts lateral movement but is repeatedly blocked by frequent re-auth. ELE stabilises at a lower level.

If there were no adaptation then the attacker has120-second time windows to move laterally (attack) against the Payment Database (Node 16) by t=280 seconds, and there will be a total loss of $50,000. Alternatively, if adapting it prevents the attacker from breaching the system entirely as indicated by the Mean Time to Successful Breach (MTTSB) increasing from 480 seconds to over 600 seconds. This provides a clear indication of the practical application of the framework for real-time risk mitigation.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

7. Statistical Significance

All of the differences identified (i.e., reduction of 37% in expected loss, reduction of 39% in Conditional Value-at-Risk, and improvement to Mean Time to Security Breach) were evaluated for statistical significance using **three complementary approaches**:

1. Bootstrap Resampling (10,000 iterations): For each comparison, we resampled with replacement from the simulation results to construct 95% confidence intervals. The p-values were calculated as the proportion of bootstrap samples in which the observed effect was not present (i.e., for ELE reduction, the proportion of samples where the reduction was ≤ 0%).

2. Two-Sample t-test: We conducted independent two-sample t-tests comparing the optimal policy (Δ=105s) against the static policy (Δ=300s) for ELE, CVaR, and MTTSB. The t-test assumptions of normality and equal variance were verified using Shapiro-Wilk and Levene's tests respectively ($p > 0.05$ for all comparisons).

3. Mann-Whitney U Test: As a non-parametric robustness check, we also performed Mann-Whitney U tests for all comparisons to confirm that the results were not driven by distributional assumptions.

Table 12. Bootstrap confidence intervals for key comparisons

Comparison	Observed effect	95% confidence interval	p-value	Mann-Whitney p-value
ELE reduction: optimal vs. static	36.8%	[34.2%, 39.5%]	<0.001	< 0.001
CVaR reduction: optimal vs. static	39.2%	[35.8%, 42.1%]	<0.001	< 0.001
MTTSB improvement: 20 segments vs. 1 segment	372%	[340%, 405%]	<0.001	< 0.001
Scalability slope (log-log)	1.21	[1.15, 1.27]	–	--

In all comparisons, the confidence intervals do not cross zero (for reductions) or one (for the scalability slope) meaning that all of the comparisons have a p value < 0.05. The p value calculated for the reduction in expected loss was based upon assessing whether the true expected loss reduction was ≤0% or >0% based upon bootstrap simulated data. Additionally, the relatively small confidence interval indicates a precise estimate of the essential range of possible reductions supported by a large number of simulation replications (10,000 simulations for each combination of data). Because the average value of the scalability slope (1.0) for all comparisons was significantly below 2.0, it is possible to determine that the hybrid SPN attack graph framework does not appear to experience quadratic blow-up in actual practice.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

4.8 Comparison with Existing Methods

To benchmark our framework against existing approaches, we compared its performance against three alternative risk assessment methods:

1. **CVSS-based scoring only:** Using only CVSS exploitability scores without timing or concurrency modeling.
2. **Attack graph without SPN:** Using attack graph reachability with heuristic risk scoring.
3. **SPN without attack graph:** Using SPN with a single predefined attack path (similar to Ma et al., 2025).

Table 13. Comparative performance against alternative methods

Method	MTTSB Prediction Accuracy (σ deviation)	ELE Estimation Error	Concurrency Support	ZTA Policy Support
CVSS-only	N/A (no timing)	±68%	No	No
Attack graph only	N/A (static)	±52%	No	Partial
SPN without attack graph	3.8σ	±41%	Yes	Partial
Our hybrid framework	0.76-3.19σ	±18%	Yes	Full

The hybrid framework demonstrates superior accuracy in MTTSB prediction and ELE estimation while maintaining full support for ZTA-specific policies and concurrent attack modeling.

Overall, the combination of the findings from the global analysis demonstrates strong empirical support for the premise that the hybrid SPN attack graph framework (a) is accurate, (b) is scalable, and (c) is capable of providing support to make risk informed, adaptive decisions within a zero-trust architectural environment.

DISCUSSION

In the previous section, we have examined some important quantitative research results related to Zero Trust Architecture's (ZTA) risk evaluation using the new attack graph and stochastic petri net framework. What follows is our interpretation of some of those key findings, a comparison of our work to relevant prior studies, an outline of practical implications of the findings, an acknowledgment of limitations, a discussion of threats to validity, and recommendations for future research.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

1. Interpretation of Key Findings

The data on Mean Time to Security Breach (MTTSB) validated within 4 standard deviations of the empirical records of breach incidents provides significant evidence that a combination of the attack graphs and stochastic petri nets can accurately model the dynamic and stochastic aspects of attacks in an environment protected by zero trust architecture. This is important because static risk models, such as those based on CVSS scores only, will not be able to reproduce the timing distributions of breaches. Furthermore, the close correspondence of simulated to empirical distributions, despite the use of the exponential rate, indicate that this is a reasonable approximation for the range of possible attacks in the 150 microservices network.

The analysis of the sensitivity indicates that the best parameter for determining when to perform re-authentication will occur between the range of ninety seconds and one hundred and twenty seconds, with the lowest loss expected to occur around one hundred and five seconds. The reason is that the shorter time period (thirty seconds) creates an excessive amount of overhead for re-authentication due to the additional time periods that they would encounter, which would create a very small but noticeable increase in expected loss due primarily to a loss in availability (the amount of projected re-authentication time is based on the number of false positives subsequently resetting a user's account). However, if the time period was longer than one hundred and twenty seconds, then it would allow the attacker greater opportunity for lateral movement (moving throughout the targeted entity) and, thus, greater opportunity for privilege escalation (obtain greater levels of access), therefore, would have a larger cumulative expected loss than those associated with the first option described above. This optimal time period (approx. one hundred and five seconds) reduces the expected total-loss exposure (ELE) of a large company experiencing one million dollars annually due to computer security breaches by 37%, thereby translating to a total annual savings of approximately $3,700,000.00.

In regard to the Conditional Value at Risk (CVaR), the data further demonstrate that moving from a static to an optimal time interval will reduce the expected total loss for the average (expected) of the last five per cent of all events by 39% and will positively impact all computer security breaches. This is important because security managers are primarily concerned about catastrophic; rather than average behaviour; therefore, this measure provides important insight into potential for actual catastrophes (i.e. massive breaches). There are also some measures of usefulness provided in the CVaR by financial industry practitioners [23].

2. Comparison with Prior Work

To illustrate how our contribution fits into the greater body of work we compare our framework against five existing studies that are the most directly related to it. These studies are: (i) He et al.'s (2022) NIST ZTA conceptual model; this provides the logical components but does not provide any quantitative risk metrics [1]; (ii) Ou et al.'s (2006) purely structural attack graph generation approach [4]; (iii) Madan et al.'s (2004) SPN based intrusion tolerance security modelling which does not account for ZTA policies [16]; (iv) Wu et al.'s (2021) hybrid attack tree to SPN mappings where there are no time dependent metrics or ZTA controls [19]; and, (v) Ma et al.'s (2025) use of SPNs in their recent work about stochastic re-authentication; while this also makes no use of attack graphs or tail risk metrics [22].

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

The five main areas in Table 13 are: the modelling of Zero Trust Architecture policies, the stochastic nature of attack timing, the integration of attack graph data, quantitative numerical measures (mean time to successful breach, expected loss event lifetime and conditional value at risk) and the empirical validation of outcomes. Our work is the first to combine each of these areas. In contrast to traditional static CVSS-based scores, our model provides time sensitive stochastic measures to more accurately represent the true nature of threat behavior and mitigation activities. In contrast to existing discrete-time SPN security models that model a flat network, this study has expressly coded into its' SPN model the effects of micro-segmentation, least-privileged access control and re-authentication intervals. Our framework is $O(n^2)$, but in the case of the majority of sparse attack graphs (up to the complexity of a few hundred microservices), it will provide close to linear scalability. In contrast to the Markovian security models studied by El Kafhali and Salah (2017), which typically suffer from an exponential state space explosion, this model has a much larger parameter space [17].

Our findings can be compared with several recent studies in the field:

Comparison with Ma et al. (2025): Ma et al. used SPNs to evaluate re-authentication intervals in cloud environments but assumed a single predefined attack path, which led to optimistic MTTSB estimates. Our framework, by incorporating full attack graph enumeration, provides a more comprehensive risk assessment. For Δ=120s, Ma et al. reported an MTTSB of approximately 2,100s, whereas our model yielded 3,591s for credential replay scenarios—a 71% difference that highlights the importance of considering multiple attack paths.

Comparison with Chen et al. (2024): Chen et al. used Bayesian attack graphs for power system security assessment but did not incorporate time-based trust decay or dynamic re-authentication. Their model provided static risk scores without temporal dynamics. Our results demonstrate that incorporating re-authentication timing significantly affects risk estimates, with ELE varying by up to 37% across different intervals.

Comparison with Younang and Sen (2025): Younang and Sen used Bayesian attack graphs with complex probabilities for IoT applications. While they addressed probabilistic uncertainty, they did not model concurrent attacks or periodic security events. Our SPN-based approach captures concurrency naturally, allowing for simultaneous attack paths that may lead to faster compromise—a finding supported by our validation results.

Comparison with Volpe et al. (2024): Volpe et al. integrated attack graphs with deterministic Petri nets for intrusion detection, but their model lacked stochastic timing. Our results show that stochastic modeling is essential for accurate MTTSB prediction; deterministic models would have yielded a single deterministic time rather than a distribution matching empirical observation.

Agreement with Previous Work: Our findings on the importance of micro-segmentation (136% improvement alone) are consistent with the qualitative recommendations in NIST SP 800-207 (He et al., 2022) and the practical observations of Phiayura and Teerakanok (2023), who noted that segmentation significantly reduces lateral movement opportunities. The diminishing returns observed beyond 20 segments (only 300s additional MTTSB improvement) aligns with the findings of Ingols et al. (2006) on attack graph pruning efficiency.

ISSN: 2312-8135 | Print ISSN: 1992-0652 | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com | info@journalofbabylon.com

Disagreement with Previous Work: Our optimal re-authentication interval of 90-120 seconds differs substantially from the 5-minute default used in many commercial ZTA deployments. This finding challenges the heuristic-based approach noted by Ahmadi (2024) and provides quantitative justification for more frequent re-authentication in high-risk environments. The 37% ELE reduction we observed suggests that current deployments may be operating at significantly higher risk levels than necessary.

Table 13. Comparison of our work with five key prior studies

Reference	ZTA policies modelled	Stochastic timing	Attack graph integration	Quantitative metrics (MTTSB/ELE/CVaR)	Empirical validation
[1]	Conceptual only	No	No	None	No
[4]	None	No	Yes (structural)	None	No
[16]	None	Yes (exponential)	No	MTTSF (similar to MTTSB)	No
[19]	None	Yes	Partial (attack trees)	Attack probability only	No
[22]	Re-authentication only	Yes	No	MTTSB only	No
Our work	Micro-segmentation, least privilege, re-authentication	Yes (exponential)	Full mapping	MTTSB, ELE, CVaR	Yes (within 4σ)

3. Practical Implications

As a point of reference, both security architects and DevOps engineers can utilize the identified best practice of a minimum optimal re authorization interval of 90-120 seconds for subsequent cloud-native ZTA implementations for their specific conditions based on sensitivity analysis to derive the best-case trade-off and risk profile that works for them. Furthermore, as demonstrated in section 4.6, the adaptive decision-making methodology can be executed through a policy engine like Open Policy Agent (OPA) OR the NIST defined Policy Decision Point (PDP) where the engine periodically calculates the latest marking (compromised assets and trust scores) and uses a pre-calculated response surface to determine if any action is required (e.g., shortening re authorization intervals or tightening segmentation). This method also allows for MTTSB and ELE metrics to be considered SLAs for zero trust deployments; for example, an enterprise may have a requirement that it must meet an MTTSB of at least one hour for its payment service based on its current re authorization policy.

ISSN: 2312-8135 | Print ISSN: 1992-0652 | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com | info@journalofbabylon.com

4. Limitations

The theoretical basis for our model is that the time it takes to conduct an attack follows an exponentially distributed random variable, which puts us in the realm of a continuous time Markov Chain that is analytically solvable. Unfortunately, when we look at actual dwell times of real-life attackers, they often exhibit either a heavy-tailed or deterministic nature, such as the fixed time period associated with an attacker performing a scheduled vulnerability scan. Even though our use of an exponential approximation for attack times worked well within four standard deviations of our validation testing, this may not be valid because it does not account for very slow or stealthy attacks that seek not to be detected for long periods of time. Future studies will attempt to mitigate this limitation by using phase type distributions or semi-Markov processes.

Another limitation is with empirical validation, as the ZTA Breach 150 dataset is synthetic, even though it was statistically calibrated using publicly available data (VCDB, CSE CIC IDS2018). Obtaining actual production data from an operational ZTA environment would provide even more robust evidence than we currently have. We are currently looking for industry partners to obtain this type of data.

The calculation of the CVaR requires that we know the financial loss impact of each compromised asset. It is very difficult to determine the financial loss from a data breach for an organization; this estimation could be dependent on numerous factors within a given business. That being said, we can still use the framework developed in this paper to determine relative loss scores (1-10) rather than absolute dollar amounts; we can also conduct a sensitivity analysis, which uses several different assumptions as to how much an organization could lose.

Finally, the $O(n^2)$ theoretical complexity, while acceptable for networks up to several hundred microservices, may become a bottleneck for very large systems (e.g., 1000+ microservices). Hierarchical modelling, where the network is partitioned into domains that are analysed separately, could alleviate this issue.

5. Threats to Validity

Internal Validity relates to estimating transition rates from CVSS scores and MITRE ATT&CK data as described by Al-Sada et al. (2023) [24]. To address this, we conducted sensitivity analyses across a broad range of rates (by varying both re-authentication intervals and trust thresholds) that demonstrate that the qualitative conclusions of the analysis (an optimally selected interval that results in a 37% reduction) are robust to moderate parameter variations.

External Validity is concerned with the degree to which our results can be generalized. The attack graph and network topology are for a simulated enterprise with 150 microservices, yet the general methodology can be applied to any ZTA-enforced system. We have not applied the framework to legacy monolithic applications or IoT networks where the assumptions made concerning micro-segmentation may not hold.

Construct Validity is concerned with defining MTTSB. Security breaches may be defined differently by different organizations (e.g. data exfiltration vs unauthorized access). We provided an official re-usable definition that is based on critical asset compromise, but this could be modified by modifying the SPN absorbing states used.

6. Future Research Directions

There are many extensions that could be valuable short-term improvements to this work. The use of phase type distribution in place of the exponential attack durations would allow for the creation of models that had deterministic or heavy-tailed delay characteristics while also allowing for

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

Markovian analysis of these types of delays [15]. Additionally, integrating the framework with real-time threat intelligence feeds (such as SIEMs) would allow for dynamic updates to attack rates based on newly discovered vulnerabilities, thereby creating an agile/reliable method for measuring and reducing overall risk exposure. The application of Multi-Objective Optimization would help to identify optimal trade-offs between reducing ELEs and authentication delays/user friction by producing a Pareto frontier of optimal policies. Finally, extending this framework to collaborative zero trust environments where multiple organizations are able to share risk information without exposing any sensitive data presents another possible avenue for future research that may be solved through Federated Learning or Secure Multi-Party Computation methodologies.

CONCLUSION

In the research presented in this article, we created a new model that combines attack graphs with Stochastic Petri Nets, allowing for an assessment of Zero Trust Architectures using quantitative risk metrics for time awareness. A significant improvement over previous risk assessment models that were either static or purely qualitative, our new model uses explicit representations of micro segmentation, least privilege, and dynamic re-authentication intervals as required by all three core components of ZTA. Three risk metrics that have been specifically developed for ZTA (MTTSB, ELE, and CVaR) were introduced and demonstrated how they can be determined using an underlying SPN model. In our simulation of a company's infrastructure composed of 150 microservices, the analysis showed that the simulated MTTSB came within 4 standard deviations from the actual breach data, confirming the predictive validity of this new model. After conducting extensive analysis on the sensitivity of re-authentication intervals, it was determined that timeframes between 90 seconds and 120 seconds provided the best minimization of the Expected Loss Event (ELE) in addition to a 37% reduction in comparison to static policies that have a five (5) minute interval, reducing the Overall Value-Added Risk Tail (CVaR) at least 39% as well. In addition to these results, we demonstrated an O (n2) complexity for this framework as well as the ability to provide near-linear run-times based on the size of an attack graph with respect to the number of microservices (i.e., several hundred microservices); and that Adaptive Policy- or rule-based decisions using real-time risk metrics could be achieved through the shortening of the re-authentication and policy based on real-time detection of anomalies.

In summary, this research makes three contributions to the field: (1) It presents a formally validated quantitative risk framework for Zero Trust Architectures that combines structural attack path analysis with stochastic timing modeling; (2) It bridges the gap between conceptual Zero Trust principles and measurable security guarantees by providing a framework to analyze risk associated with ZTA policies; (3) It offers actionable design guidelines—specifically the optimal re-authentication interval range of 90-120 seconds—that security architects can consider when implementing cloud-native ZTA solutions, subject to their specific operational requirements and risk tolerance.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

Conflict of interests.

There are non-conflicts of interest.

References

[1] Y. He, D. Huang, L. Chen, Y. Ni, and X. Ma, "A survey on zero trust architecture: Challenges and future trends" Wireless Communications and Mobile Computing, vol. 2022, no. 1, p. 6476274, 2022.

[2] P. Phiayura and S. Teerakanok, "A comprehensive framework for migrating to zero trust architecture" IEEE Access, vol. 11, pp. 19487–19511, 2023.

[3] M. L. Gambo and A. Almulhem, "Zero trust architecture: A systematic literature review" Journal of Network and Systems Management, vol. 34, no. 1, p. 25, 2026.

[4] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation" in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS), pp. 336–345, ACM, 2006.

[5] M. Taleb-Berrouane, F. Khan, and P. Amyotte, "Bayesian Stochastic Petri Nets (BSPN) – A new modelling tool for dynamic safety and reliability analysis" Reliability Engineering & System Safety, vol. 193, p. 106587, 2020.

[6] S. Ahmadi, "Zero trust architecture in cloud networks: Application, challenges and future opportunities" Journal of Engineering Research and Reports, vol. 26, no. 2, pp. 215–228, 2024.

[7] J. Freund and J. Jones, "Measuring and managing information risk: A FAIR approach" Butterworth-Heinemann, 2014.

[8] N. Fenton and M. Neil, "Risk assessment and decision analysis with Bayesian networks (2nd ed.)" CRC Press, 2018.

[9] L. Chen, T. Zhang, Y. Ma, Y. Li, C. Wang, C. He, … and N. Li, "A Bayesian-attack-graph-based security assessment method for power systems" Electronics, vol. 13, no. 13, p. 2628, 2024.

[10] S. T. Asakpa, "Quantifying financial cyber risks in financial institutions: Monte Carlo simulations, time-series forecasting, and cost-benefit optimization" International Journal of Advance Research, Ideas and Innovations in Technology, p. 2454-132, 2024. https://www.ijariit.com

[11] K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense" in Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06), pp. 121–130, IEEE, 2006.

[12] S. Jajodia, S. Noel, and B. O'berry, "Topological analysis of network attack vulnerability" in Managing Cyber Threats: Issues, Approaches, and Challenges, pp. 247–266, Springer US, 2005.

[13] V. C. W. Younang and A. Sen, "Security risk assessment using Bayesian attack graphs and complex probabilities for large scale IoT applications" IEEE Transactions on Dependable and Secure Computing, 2025.

[14] I. Semertzis, V. S. Rajkumar, A. Ştefanov, F. Fransen, and P. Palensky, "Quantitative risk assessment of cyber-attacks on cyber-physical systems using attack graphs" in Proceedings of the 10th Workshop on Modelling and Simulation of Cyber-Physical Energy Systems (MSCPES), pp. 1–6, IEEE, 2022.

[15] M. Taleb-Berrouane, F. Khan, and P. Amyotte, "Bayesian Stochastic Petri Nets (BSPN) – A new modelling tool for dynamic safety and reliability analysis" Reliability Engineering & System Safety, vol. 193, p. 106587, 2020.

[16] B. B. Madan, K. Goševa-Popstojanova, K. Vaidyanathan, and K. S. Trivedi, "A method for modeling and quantifying the security attributes of intrusion tolerant systems" Performance Evaluation, vol. 56, no. 1–4, pp. 167–186, 2004.

[17] S. El Kafhali and K. Salah, "Stochastic modelling and analysis of cloud computing data center" in Proceedings of the 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), pp. 122–126, IEEE, 2017.
[18] W. Leonardo, T. Bezerra, and G. Callou, "Stochastic Petri net models for availability and performance evaluation of Nextcloud service hosted in Apache CloudStack" in Proceedings of the 13th Latin-American Symposium on Dependable and Secure Computing, pp. 165–170, 2024.
[19] Z. Wu, L. Tian, Y. Zhang, Y. Wang, and Y. Du, "Network attack and defense modeling and system security analysis: A novel approach using stochastic evolutionary game Petri net" Security and Communication Networks, vol. 2021, no. 1, p. 4005877, 2021.
[20] G. Volpe, M. Fiore, A. la Grasta, F. Albano, S. Stefanizzi, M. Mongiello, and A. M. Mangini, "A Petri net and LSTM hybrid approach for intrusion detection systems in enterprise networks" Sensors, vol. 24, no. 24, p. 7924, 2024.
[21] Y. Han, L. Zhang, Y. Wang, X. Deng, Z. Gu, and X. Zhang, "Research on the security of IPv6 communication based on Petri Net under IoT" Sensors, vol. 23, no. 11, p. 5192, 2023.
[22] Z. Ma, H. Wei, J. Jiang, B. Wang, H. Wang, and Z. Di, "A lightweight zero-trust authentication architecture for IoT via unified enhanced FAST-SM9 and dynamic re-authentication" PLOS ONE, vol. 20, no. 10, p. e0332943, 2025.
[23] H. Föllmer and A. Schied, "Stochastic finance: An introduction in discrete time" Walter de Gruyter GmbH & Co KG, 2025.
[24] B. Al-Sada, A. Sadighian, and G. Oligeri, "Analysis and characterization of cyber threats leveraging the MITRE ATT&CK da tabase" IEEE Access, vol. 12, pp. 1217–1234, 2023.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

الخلاصة

على الرغم من الزيادة الملحوظة في تبني بنى الثقة الصفرية (Zero Trust Architectures) (ZTA)، إلا أن هناك عددًا قليلًا جدًا من الأساليب الرسمية لتقييم المخاطر المتغيرة باستمرار لهذه الأنظمة. يقترح بحثنا إطار عمل هجينًا قائمًا على شبكات بتري العشوائية (Stochastic Petri Nets) (SPNs) ومخططات الهجوم، لتقييم المخاطر النظامية المرتبطة بتطبيق بنية الثقة الصفرية من خلال النمذجة الصريحة للتجزئة الدقيقة، ومبدأ أقل الامتيازات، وفترات إعادة المصادقة. نقدم ثلاثة مقاييس كمية للمساعدة في تقييم المخاطر من منظور نظامي: متوسط الوقت حتى حدوث اختراق أمني (Mean Time to Security Breach) (MTTSB)، والتعرض المتوقع للخسائر (Expected Loss Exposure) (ELE)، والقيمة المشروطة المعرضة للخطر (Conditional Value at Risk) (CVaR). نتحقق من صحة إطار العمل من خلال محاكاة شبكة تضم 150 خدمة مصغرة، حيث تقع قيم MTTSB المتوقعة ضمن أربعة انحرافات معيارية من متوسط الوقت الفعلي بين الاختراقات الملاحظة في الشبكات الحقيقية. يُظهر تحليل الحساسية أن استخدام فترات إعادة المصادقة التي تتراوح بين 90 و120 ثانية يُقلل من معدل الخطأ في الوصول بنسبة 37% مقارنةً بالسياسات الثابتة. يتميز الإطار المقترح بخصائص توسع من الدرجة $O(n^2)$ (خطية تقريبًا في التطبيق العملي)، ويمكن استخدامه لتوفير أساس لاتخاذ قرارات تكيفية واعية بالمخاطر في أنظمة التحكم بالوصول من الجيل التالي، وذلك من خلال مقاييس مخاطر قابلة للقياس الكمي.

الكلمات المفتاحية: بنية الثقة الصفرية، شبكات بتري العشوائية، مخططات الهجوم، التقييم الكمي للمخاطر، التحكم التكيفي في الوصول

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com