UDC: 004.75+004.8:519.23
DOI: https://doi.org/10.17721/ISTS.2024.8.26-33

Serhii TOLIUPA, DSc (Engin.), Prof.
ORCID ID: 0000-0002-1919-9174
e-mail: tolupa@i.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Maksym KOTOV, PhD Student
ORCID ID: 0000-0003-1153-3198
e-mail: maksym_kotov@ukr.net
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

PROTECTION MODEL AGAINST DISTRIBUTED GRADUAL DEGRADATION ATTACKS BASED ON STATISTICAL AND SEMANTIC APPROACHES

Background. *Nowadays, every critical sector of social institutions conducts its operations on top of distributed processing systems. Contemporary digital infrastructure heavily relies on user-provided data in its operation. As a result, distributed attacks based on botnets are in a continuous state of arms race with the protection methods that filtrate malicious data influx. A common method to do so often relies on heuristics and human-oriented verifications. As the new advancements in the field of artificial intelligence emerge, such attacks adopt new oblique paths towards achieving their goals. The successful execution of the said plan could lead to a gradual resource depletion on the target system. The purpose of this research is to address such threats with a combination of statistical and semantic approaches.*

Methods. *The following research conducts a theoretical analysis and systematization of the distributed gradual degradation attack in distributed systems and its implication in the context of the evolving technologies of artificial intelligence. Mathematical modeling is leveraged to define the proposed model's properties and execution process. The proposed model heavily relies on statistical methods for analyzing time series data and its deviations, as well as classification neural networks for semantic detection of suspicious behavior.*

Results. *As a result of the following research, a new model is developed that leverages statistical and semantical verification for anomaly detection. The continuous monitoring and detection process is optimized towards highly loaded systems with a constant flurry of data streams.*

Conclusions. *Since the distributed attacks could be potentially equipped with intelligent means to bypass existing security measures, the development of a protection model against potential resource leaks is gaining relevance. The recent success in the development of artificial generative intelligence leads to raising concerns about the safety and adequacy of the current security measures against automation-based distributed attack vectors. It is often a case that the protection models are inclined towards prevention of the attack rather than recovery. This approach, while targeting the source of risks, often leads to complacent design decisions without considering the potential outcomes of a successful breach. The proposed model provides a theoretical foundation for building systems that both react to the active execution of threats and perform recovery mechanisms, assuming that the attack may potentially bypass initial security measures.*

Keywords: *distributed systems, gradual degradation attacks, resource exhaustion, statistical analysis, semantic approaches, resilience, LightGBM, Distilbert, EWMA.*

Background

The majority of contemporary cloud and distributed systems work on top of a client-server model. Within the context of such an approach, the server expects requests from a client system to initiate an interaction process. Once the message is received, the service nodes perform desired computation and resource allocation as defined per operational logic. It is often a case for such RPC or HTTP endpoints to have an authentication system in place that relies on multi-factor human interaction verification by involving multiple external services or identification credentials, such as phone number. Nonetheless, it is also a common approach to have a dedicated public subset of services since it could be useful to provide a succinct demonstration of the system's capabilities or simply used to request initial access rights.

In that context, it is important to emphasize the difference between transient and mutation requests. Transient requests are executed on a stateless basis. Each subsequent request does not influence the system's state nor the results of any subsequent execution. Such requests often rely on data reads or real-time computation rather than deferred, asynchronous execution or static storage services. Mutating requests involve the allocation or creation of additional computational or storage assets as per the request's parameters. These requests could be generalized further by including state changes of non-persistent logical resources. An example of such could be rate limits for external services.

Additionally, mutating requests could be further split into categories of idempotent and cumulative state-changing interactions. Where the first guarantees that any subsequent operation of the same nature will result in the same state of the system. As a matter of fact, the idempotent category spans between a subset of mutating and the whole set of transient requests. Each subsequent transient request does not change the state at all, hence leaving it the same after an arbitrary amount of requests of the same nature. Whereas cumulative state operations gradually modify the shared state and always impact the allocation of computational assets in one way or another.

Service providers could be categorized into two categories: stateful and stateless systems. Stateless systems typically rely on providing transient services. Such systems are extremely scalable and easy to coordinate since no consensus is required. In addition to that, they do not rely on persistent resource allocation and utilize exclusively dynamic operational hardware. Stateful systems are characterized by their operational uniqueness. Interaction with such a system could involve the allocation or deallocation of persistent resources, mainly storage. It is usually the case that within a single distributed system that provides a complex multi-step service, there are sets of both categories.

Having established the context of interaction-driven services, we can now discuss the security implications and attack vectors for each described processing model. In the context of this article, automation-based distributed attacks

ISSN 2707-1758

are in the limelight of the research. The most common and known attack of such type is Deny of Service (DOS) and its invariant Distributed Deny of Service (DDOS). The origins of this attack took root in the times of developing global interconnection networks. Its implications, strategies, and modern protection mechanisms are well studied and defined in numerous scientific works (Mirkovic, & Reiher, 2004; Douligeris, & Mitrokotsa, 2004; Srivastava et al., 2011; Zargar, Joshi, & Tipper, 2013; Zhang, Wang, & Chen, 2017). DOS is characterized by its aggressive nature and is applicable to every type of request described previously.

Though the impact could differ greatly between mutating and transient requests, the primary goals are generally the same: overwhelm the target system with a barrage of nonsensical computational tasks to stifle or preclude execution of concurrent legitimate processing threads. The congestion often leads to temporal downtimes, loss of reputation, trust, and, as a result, of fungible assets. The protection methods often involve massive cloud networks with innate capabilities of service masking and traffic distribution. Such networks often rely on pattern-recognition models to differentiate between authentic and malicious packet streams.

In contrast, this article aims to define a potential new attack vector that is more subtle and clandestine. The Distributed Gradual Degradation attack does not rely on aggressive bombardment of the target systems. It relies on the gradual creation and execution of mutative cumulative requests that incrementally reduce the system's capacity to provide a service. Since it's not based on congestion surge, it's less conspicuous and is more likely to be executed successfully. The said attack is inefficient for transient and has limited efficiency on idempotent requests since they by definition have a limited impact on the system's state. The common approach to fending off automated requests is based on human authenticity verification based on interaction heuristics, such as mouse moves or choices made.

With the recent development of generative artificial intelligence and its ever-improving qualities, a new arms race between such challenge-based approaches and their automated solution is ongoing. AI models are potentially capable of mimicking human behavior and decision-making processes to a sufficient degree to bypass current identification methods. This led to the development of yet more confounding challenges with multiple logical steps. Though with a rapid evolution of the generative AIs, it remains unclear whether those measures are sufficient (Hernández-Castro et al., 2017; Kovács, & Tajti, 2023; Sukhani et al., 2021).

The purpose of the article. The intention and goal of this research is to develop a theoretical protection model against the distributed gradual degradation attack vector. This article aims to provide a solid set of active and passive measures towards ensuring adequate usage of the system's resources by external requests. Within the scope of this research, we consider and outline the integration of the aforementioned protection model within the context of streaming asynchronous communication services and static storage engines.

The key principles that form the foundation of this model are efficiency and a knowledge-based approach. The first principle is straightforward: as the model aims to protect resources, it should rationally utilize them itself. The latter means that ability should be tapered towards capabilities of classification neural networks in juxtaposition to generative AIs. The reasoning behind this approach stems from the significant computational complexity involved in developing and training these models. The training and execution of generative models require exponentially more time than that of classification models. Hence, the attack becomes ineffectual since it would require more sources than it would seemingly degrade on a target machine.

Analysis of literary sources. The distributed gradual degradation attack vector enhanced by recent developments and improvements in artificial intelligence is not extensively studied as of now but is a looming topic of scientific research. Nevertheless, its foundational components, such as bypassing contemporary bot detection systems, and its implications are growing rapidly in relevance.

Active research on DDOS has been ongoing since the early 2000s. Significant contributions towards attack definition, classification, and potential prevention methods are provided by the works of (Mirkovic, & Reiher, 2004; Douligeris, & Mitrokotsa, 2004; Srivastava et al., 2011; Zhang, Wang, & Chen, 2017).

With the development of deep learning models, computer vision, and artificial intelligence in general, common protection methods against automation-based attacks become increasingly susceptible. In that direction, impactful research results were published by (Na et al., 2020; Sukhani, et al., 2021; Kovács, & Tajti, 2023; Hernández-Castro et al., 2017).

Improving detection model efficiency involves statistical estimation and evaluation of time series data. Exponentially Weighted Moving Average (EWMA) is described within the works of (Hunter, 1986; Lucas & Saccucci, 1990; Cox, 1961). The Integrated Moving Average (ARIMA) method and its implications are outlined by (Box, & Pierce, 1970; Nelson, 1998). Semantic detection in the context of phishing attacks is assessed by the following studies: (Buchyk et al., 2024; Buchyk, Shutenko, & Toliupa, 2022; Toliupa et al., 2023). The authors provide and describe models of detecting suspicious contents of emails with a set of semantic methods such as cosine distance between data-driven vectors.

Classification neural networks serve as the backbone of the proposed model. Their key feature is a simplified and efficient training process that is exponentially faster than that of the generative AIs. The significance and operational basis of such technology are described within the works of (Zhang, Zhang, & Yu, 2017). Decision trees LightGBM and XGBoost are described within works of (Leevy et al., 2020; Zhao, Wang, & Wang, 2023). Last, but not least, the language processing model Distilbert is assessed and studied by (Adoma, Henry, & Chen, 2020; Büyüköz, Hürriyetoğlu, & Özgür, 2020).

Methods

The following research conducts a theoretical analysis and systematization of the distributed gradual degradation attack in distributed systems and its implication in the context of the evolving technologies of artificial intelligence. Mathematical and graphic modeling are leveraged to define the proposed model's properties and execution process. The proposed model heavily relies on statistical methods for analyzing time series data and its deviations, as well as classification neural networks for semantic detection of suspicious behavior.

This work additionally describes an exemplary architecture of a target distributed system that utilizes static-storage and stream-oriented services to outline an

integration of the proposed model in the workloads based on different processing paradigms. Since most contemporary distributed systems utilize queuing and asynchronous communication models, the implications and middleware-oriented integration method of the developed protection model are described within the context of the streaming architectures. Additionally, this paper outlines the external monitoring-oriented integration of the said protection model for the static data warehouses analysis and resource deallocation.

Results

The following section presents a novel theoretical model of protection against distributed gradual degradation attacks. We will initially outline the statistical approach towards detecting anomalies. After that, we will delve into the application of a neural network for deep semantical scanning of suspicious activity spikes. This theoretical model will later be applied to an exemplary distributed system's architecture. First and foremost, we will describe the architecture itself, its services, and intercommunication methods. After that, an integration plan for the static data storage engines and real-time streaming services will be provided to facilitate integration of the proposed method into modern distributed systems.

Protection model against distributed gradual degradation attacks. The automated activity detection model is comprised of multiple parts. Firstly, it defines the global and internal timeframe segregation and boundaries to achieve better performance results. Based on those intervals, an Exponentially Weighted Moving Average (EWMA) method is used to control the degree of suspiciousness. That degree influences the aggressiveness of semantic scanning.

We will first start with the management aspect of the proposed model. Let us define time parameters:

- T: Total timeline over which data (incoming requests) are observed.
- t: Specific time point within T.
- $R(t)$: Set of records (incoming requests) at time t.
- W: Size of the sliding window (in time units).
- $W(t)$: Sliding window at time t, containing records from $t - W$ to t.

The interval parameters are expressed as:

- G_s: Size of each global interval (Global Interval Size).
- I_s: Size of each internal interval within a global interval (Internal Interval Size).
- G_i: The i-th global interval, $G_i = [(i-1)G_s, iG_s)$.
- $I_{i,j}$: The j-th internal interval within G_i, $I_{i,j} = [(i-1)G_s + (j-1)I_s, (i-1)G_s + jI_s)$.
- n_I: Number of internal intervals per global interval, $n_I = G_s/I_s$.

Semantic sampling parameters are defined as follows:

- n_s: Initial number of samples per global interval ($Num\_Samples$) ($0 < n_s \leq n_I$).
- T_s: Sampling threshold ($Sampling\_Threshold$), expressed as a percentage.
- α: Smoothing factor for EWMA ($0 < \alpha \leq 1$).
- β: Sensitivity factor for adjusting the number of samples based on statistical anomalies.
- δ: Sensitivity factor for adjusting the number of samples based on semantic anomalies.
- $N_{i,j}$: Number of records in internal interval $I_{i,j}$.
- $N_{\text{total},i}$: Total number of records in global interval G_i.

At any time t:

$$\mathrm{W(t)} = \{\mathrm{R(s)} \mid \mathrm{t - W \leq s \leq t}\}. \quad (1)$$

The sliding function defines a set of records that are being buffered and evaluated. It practically limits the resources allocated to the detection model and provides a granular control for environments with different memory capacities. The sliding function continuously moves in time and contains a set of the most recent records. This function could also be used to establish a retrospective analysis by propelling the window backwards in time rather than forward.

The entire timeline T is divided into global intervals:

$$\mathrm{G_i} = [(\mathrm{i}-1)\mathrm{G_s}, \mathrm{iG_s}), \quad \mathrm{i} \in \mathrm{N}. \quad (2)$$

Global intervals allow set boundaries for semantic sampling strategy. Since semantic sampling is computationally heavy, as we will see later on, it is important to use it as a last resort rather than brute force. Records are assigned to these intervals based on timestamps.

Each global interval G_i is subdivided into n_I internal intervals:

$$I_{i,j} = [(i-1)G_s + (j-1)I_s, (i-1)G_s + jI_s), \quad j = 1,2,\dots,n_I. \quad (3)$$

Internal discretization is another aspect of performance improvement. It allows limiting sampling size to a controlled set of records. The combinations of these parameters allow to manage the risks and resource utilization, where the latter is of high importance because the entire purpose of the model is to save resources.

Moving on the semantic sampling, the dynamic number of samples for time t is defined as follows:

$$n_s(t) = n_s + \lceil \beta \times D(t) \rceil. \quad (4)$$

Where $D(t)$ is the degree of anomalies detected at time t; $\lceil \cdot \rceil$ is a ceiling function to ensure an integer number of samples. This approach allows to continuously react in stochastic environments.

Statistical approach based on EWMA is used to manage $n_s(t)$ and react efficiently to ongoing security events at each time t (Cox 1961, p. 414; Hunter, 1986, p. 203; Lucas, & Saccucci, 1990, p. 1):

- Compute the average number of records in the sliding window:

$$\mathrm{x_t} = \frac{\mathrm{N_W(t)}}{\mathrm{W}}, \quad (5)$$

- where $N_W(t) = |W(t)|$.
- Update EWMA:

$$\mathrm{EWMA(t)} = \alpha \mathrm{x_t} + (1-\alpha)\mathrm{EWMA(t-1)}, \quad (6)$$

- Compute the deviation:

$$D(t) = |x_t - \mathrm{EWMA}(t)|. \quad (7)$$

An anomaly is detected if $D(t)$ exceeds a predefined anomaly threshold A_T. The number of samples $n_s(t)$ is adjusted dynamically in real time as shown in equation 4.

The sampling is done at random for n_s internal timeframes. Let us first define utility functions:

- $P_{\text{bot}}(r)$: $R \to [0,1]$ a function that maps each record $r \in R$ to a probability $P_{\text{bot}}(r)$, where $P_{\text{bot}}(r)$ represents the likelihood that record r is bot-origin. We will discuss its definition later.
- $\mathrm{A(r)}$: activation function outputs 1 if $P_{\text{bot}}(r) \geq T_{\text{bot}}$, and 0 otherwise.
- T_{bot}: is the threshold for determining bot-origin.

The process itself is defined as follows:

For $k = 1$ **to** $n_s(t)$**:**

- Randomly select an internal interval I_{i,j_k} within G_i.
- Collect records R_{i,j_k} in I_{i,j_k}.
- Compute $N_{i,j_k} = |R_{i,j_k}|$.

ISSN 2707-1758

- **For each** $r \in R_{i,j_k}$, **do:**
  - Apply the semantic recognition function $P_{\text{bot}}(r)$, which returns a probability from 0 to 1 that the record is bot-origin.
  - Compute the output of the activation function:

$$\mathrm{A(r)}=\begin{cases}1 & \text{if } P_{\text{bot}}(\mathrm{r}) \geq T_{\text{bot}} \\ 0 & \text{if } P_{\text{bot}}(r) < T_{\text{bot}}\end{cases}, \quad (8)$$

- Compute the total number of likely bot-origin entries in I_{i,j_k} as:

$$\mathrm{B_{i,j_k}} = \sum_{\mathrm{r}\in \mathrm{R_{i,j_k}}} \mathrm{A(r)}\,, \quad (9)$$

- Evaluate the sampling condition:

$$\frac{\mathrm{B_{i,j_k}}}{\mathrm{N_{i,j_k}}} > T_s\,, \quad (10)$$

- **If** the condition is met increment $n_s(t)$ by δ (increase sampling rate).
- **Else** continue to the next sample without adjusting $n_s(t)$.

This process allows the algorithm to react and adjust itself depending on events inside stochastic environments. This model is also highly customizable, allowing for different threshold and resource management parameters.

Semantic sampling model. Now let's move on the discussion of $P_{\text{bot}}(r)$ and its definition. Firstly, the model processes input data consisting of structured features x_s and textual content x_t. The structured data $x_s \in R^m$, where m is the number of structured features, is transformed into a feature vector $h_s \in R^{d_s}$ through a function f_{LGBM} (Ke et al., 2017, p. 3149; Leevy et al., 2020, p. 190; Zhao, Wang, Y., & Wang, J., 2023, p. 622):

$$\mathrm{h_s} = \mathrm{f_{LGBM}(x_s)}. \quad (11)$$

Function f_{LGBM} represents the processing performed by a LightGBM model. LightGBM is a gradient boosting decision-tree model that maps the structured input x_s to a learned feature representation h_s of dimension d_s. The output of that function is a high-dimensional vector that captures features of the provided data.

Simultaneously, the textual data x_t is mapped to an embedding vector $h_t \in R^{d_t}$ using a function $f_{\text{DistilBERT}}$ (Adoma, Henry, & Chen, 2020, p. 117; Büyüköz, Hürriyetoğlu, & Özgür, 2020, p. 9; Dogra et al., 2021, vol. 248):

$$\mathrm{h_t} = \mathrm{f_{DistilBERT}(x_t)}. \quad (12)$$

In this case, $f_{\text{DistilBERT}}$ represents the DistilBERT model, which processes raw text and converts it into a contextual embedding of dimension d_t. DistilBERT is a transformer-based language model that captures the semantic meaning and contextual nuances of the textual data and provides a dense vector representation h_t.

The key factors while choosing the models were performance, accuracy, and their ratio. Since the main goal is to preserve resources and reduce costs, the decision was made towards most efficient available models.

The feature vectors $\boldsymbol{h_s}$ and $\boldsymbol{h_t}$ are then concatenated to form a combined feature vector $h_c \in R^d$:

$$\mathrm{h_c} = \begin{bmatrix}\mathbf{h_s} \\ \mathbf{h_t}\end{bmatrix}. \quad (13)$$

Where $d = d_s + d_t$ is the total dimensionality after concatenation.

The combined feature vector $\boldsymbol{h_c}$ serves as the input to a sequence of L fully connected layers. Each layer l in this sequence performs a linear transformation followed by a non-linear activation function ϕ, the ReLU (Rectified Linear Unit) (Arora et al., 2018).

For $l = 1$ **to** L**:**

- Linear transformation:

$$\mathrm{z^{(l)}} = \mathrm{W^{(l)}a^{(l-1)}} + \mathrm{b^{(l)}}, \quad (14)$$

- Activation function:

$$\mathrm{a^{(l)}} = \phi\left(\mathrm{z^{(l)}}\right) = \max\left(0, \mathrm{z^{(l)}}\right). \quad (15)$$

Where:

- $W^{(l)} \in R^{n_l \times n_{l-1}}$ is the weight matrix for layer l.
- $b^{(l)} \in R^{n_l}$ is the bias vector for layer l.
- $a^{(l-1)}$ is the activated output from the previous layer (with $a^{(0)} = h_c$).
- n_l is the number of neurons in layer l.
- $n_0 = d$ is the size of the input layer.

After processing through the L fully connected layers, the model applies a final linear transformation followed by a sigmoid activation to produce the output probability $\hat{y}$ (Arora et al., 2018; Pratiwi et al., 2020, vol. 1471):

- Linear transformation:

$$\mathrm{z^{(L+1)}} = \mathrm{w^{(L+1)^T}a^{(L)}} + \mathrm{b^{(L+1)}}\,, \quad (16)$$

- Sigmoid activation:

$$\hat{\mathrm{y}} = \sigma\left(\mathrm{z^{(L+1)}}\right) = \frac{1}{1+\mathrm{e^{-z^{(L+1)}}}}. \quad (17)$$

Where:

- $w^{(L+1)} \in R^{n_L}$ is the weight vector for the output layer.
- T signifies that the matrix is transposed.
- $b^{(L+1)} \in R$ is the bias scalar for the output layer.
- The sigmoid function σ maps the input to a probability between 0 and 1.

The overall function of the model can be summarized as:

$$\hat{\mathrm{y}} = \mathrm{f_{model}(x_s, x_t)} = \\ = \sigma\left(\mathrm{w^{(L+1)^T}}\left(\phi \circ \ldots \circ \phi\left(\mathrm{W^{(1)}h_c} + \mathrm{b^{(1)}}\right)\right) + \mathrm{b^{(L+1)}}\right). \quad (18)$$

Where:

- f_{model} represents the composition of the LightGBM processing of structured data, the DistilBERT processing of textual data, and the subsequent fully connected layers leading to the final output.
- ∘ denotes function composition.
- ϕ is the activation function applied at each hidden layer.

This architecture essentially fuses two lightweight models that concern separate tasks to produce a probabilistic answer whether the data is a part of an automation-based attack. Messages that arrive at the server's endpoints are often structured and have multiple sensical fields. Such fields often hold textual, categorical, and numerical data types. Different models perform better on different data types and provide corresponding accuracy rates. DistilBERT is used to extract complex features from textual data, while LightGBM is used for categorical and numeric data.

The concrete applied definition of the model involves specifying a number of neurons in each fully connected layer through a tuple, such as $(512,256,128)$. The shown structure defines three layers with 512, 256, and 128 neurons. In this case, the dimensions of the weight matrices and bias vectors would be:

- $W^{(1)} \in R^{512 \times d}$, $b^{(1)} \in R^{512}$;
- $W^{(2)} \in R^{256 \times 512}$, $b^{(2)} \in R^{256}$;

- $W^{(3)} \in R^{128\times256}$, $b^{(3)} \in R^{128}$;
- $w^{(4)} \in R^{128}$, $b^{(4)} \in R$.

The combined vector h_c is passed through L fully connected layers with ReLU activations, computing outputs in a cycle for $l = 1$ to L:

$$\begin{cases} \boldsymbol{z}^{(l)} = \boldsymbol{W}^{(l)}\boldsymbol{a}^{(l-1)} + \boldsymbol{b}^{(l)} \\ \mathbf{a}^{(l)} = \phi\left(\mathbf{z}^{(l)}\right) = \max\left(0, \mathbf{z}^{(l)}\right) \end{cases}. \quad (19)$$

The final output is computed using a linear transformation followed by a sigmoid activation (Thakur & Dhawale):

$$\begin{cases} z^{(L+1)} = w^{(L+1)^{\mathrm{T}}} a^{(L)} + b^{(L+1)} \\ \hat{y} = \sigma\left(z^{(L+1)}\right) \end{cases}. \quad (20)$$

The model parameters, including the weights and biases of the fully connected layers and any trainable parameters within f_{LGBM} and $f_{\mathrm{DistilBERT}}$, are optimized during training to minimize the binary cross-entropy loss. This enables the model to make accurate predictions based on the input data and learning from both structured and textual information that could be passed to the system through the common communication structures, such as JSON.

Description of a target distributed system. Having defined the protection model against distributed gradual degradation attacks based on continuous monitoring and semantic sampling, let us now discuss its applied integration into contemporary distributed systems. With the theoretical context, parameters, and model properties in place, we first outline the architecture of the distributed system depicted on Fig. 1:

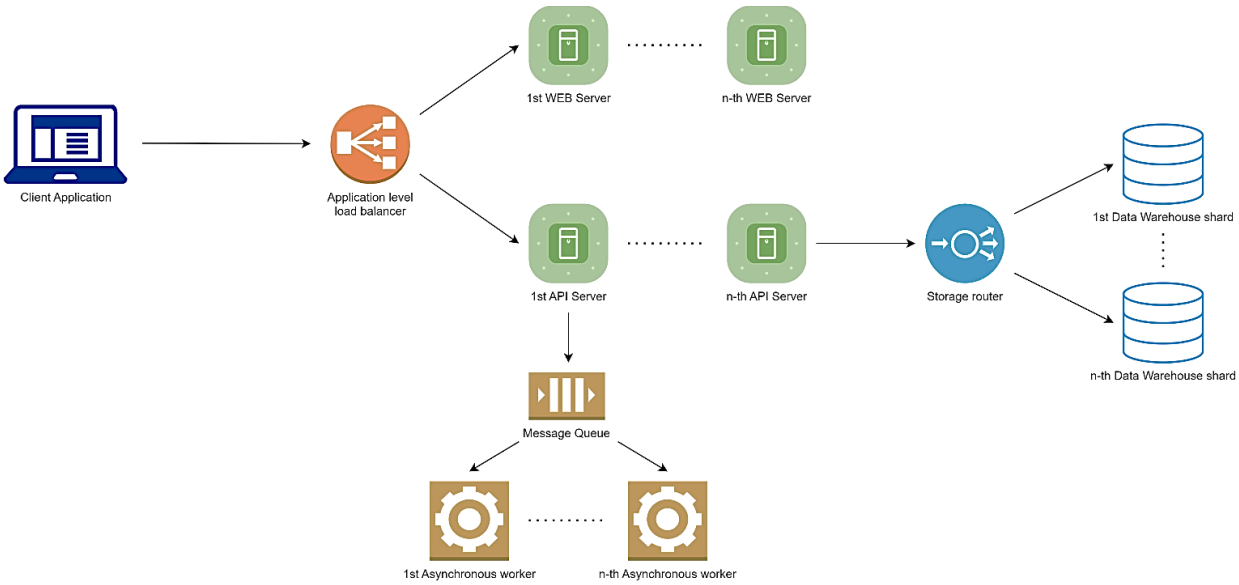

Fig. 1. Architecture of the target distributed system

This architecture represents an abstract system that uses the most common data flow methods: storage and stream-oriented. It consists of the application-level load balancer that routes the requests to the API and WEB servers based on the request paths. Subsequently, API servers could either perform a state-mutating operation in the data warehouse or initiate an asynchronous task through the means of queueing services. The common approach for building such workflows involves AMQP protocol (Prajapati, 2021).

Protection model application for data warehouse services. The data warehouse service is responsible for persistent storage, processing, and retrieval of information. It is often the case that such services are extremely hard to scale and are also the backbone of the stateful distributed system. Having said that, due to the extreme requirements for availability and consistency, it is important to integrate the protection model without impacting the response times and minimize influence on the overall performance. Fig. 2 shows the integration architecture with the persistent storage service:

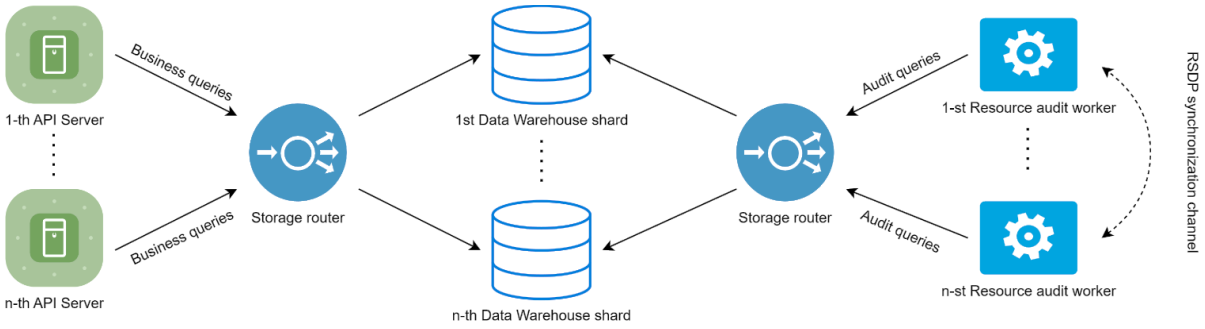

Fig. 2. Proposed model's integration as an asynchronous monitoring solution

ISSN 2707-1758

The architecture reflects an asynchronous, batch-processing approach towards resource utilization audits and protection gains distributed through gradual degradation attacks. The key principle is to allow flowless execution of requests and after that schedule retrospective tasks to verify the authenticity and validity of the mutations. For that purpose, the external "resource audit workers" perform the algorithm described within the protection model. Since the sliding window $W(t)$ is itself a parameter, it is possible to initialize such batch tasks in the required context.

Moreover, contemporary production systems tend to grow extensively in size. For that purpose, the service clusterization could be performed based on the Replica State Discovery Protocol (RSDP). RSDP provides a lightweight and efficient framework to coordinate cluster-wide operations execution and state management. Each node within such a cluster could leverage the deterministic parameters and based on its cluster position, schedule its sliding window accordingly (Kotov, Toliupa, & Nakonechnyi, 2024, p. 102; p. 156). Such an approach is efficient in terms of processing power and yet allows for scaling if needed.

Protection model application for data queueing and streaming services. The message queuing services primarily work in an "eventually transient mode". That is, these services do not save data for too long, often until the message processing is confirmed by a connected worker. These services are most frequently utilized for asynchronous and deferred operations. Which implies that the response time is not an issue, allowing for the middleware-oriented architecture. Figure 3 shows the integration architecture with the message streaming service:

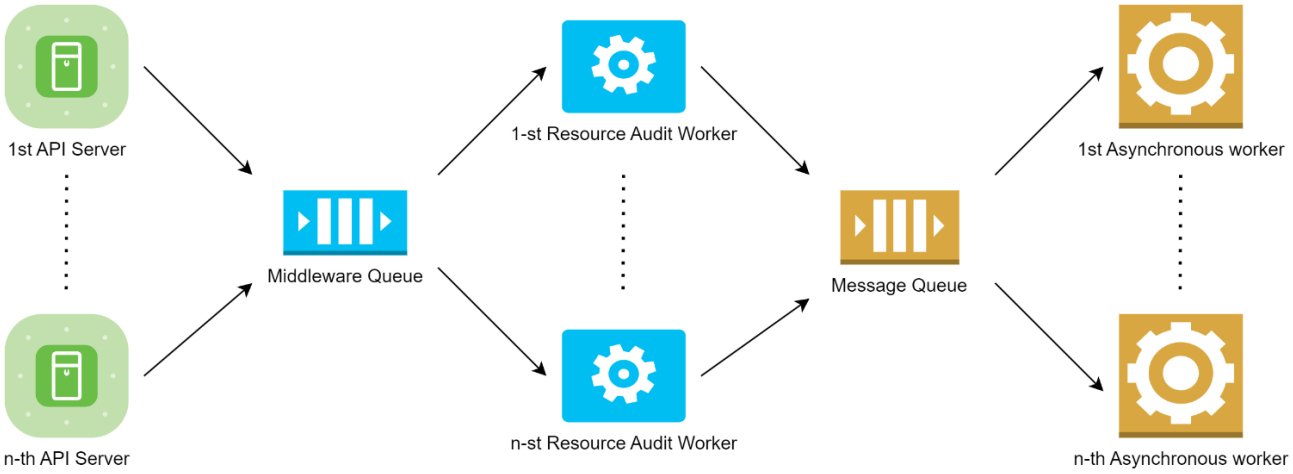

Fig. 3. Proposed model's integration as a middleware monitoring solution

Before the target message queue, this architecture implies the middleware queue for analytical and security processing. The entire model described earlier would be executed inside the "resource audit worker". Scalability in this case is trivial and does not require any coordination mechanism since the logic would entirely mirror the target queue's message processing infrastructure. Upon receiving new messages, the sliding would operate on a real-time basis and directly control the stored buffer size during the runtime. The message processing is often coordinated and distributed between the workers in a round-robin manner, thus effectively integrating load balancing capabilities for both the protection model's worker execution and the target logic itself (Prajapati, 2021).

Discussion and conclusions

Since the distributed attacks could be potentially equipped with intelligent means to bypass existing security measures, the development of a protection model against potential resource leaks is gaining relevance. The recent success in the development of artificial generative intelligence leads to raising concerns about the safety and adequacy of the current security measures against automation-based distributed attack vectors. It is often a case that the protection models are inclined towards prevention of the attack rather than recovery. This approach, while targeting the source of risks, often leads to complacent design decisions without considering the potential outcomes of a successful breach. The proposed model provides a theoretical foundation for building systems that both react to the active execution of threats and perform recovery mechanisms, assuming that the attack may potentially bypass initial security measures.

While developing the protection model against the gradual degradation attacks, the primary concern was twofold: ensuring that the system is capable of recovering from unexpected resource loss and ensuring that the detection and monitoring processes themselves don't require a superfluous amount of processing power. The primary focus was on the integration of the model within the operation context of streaming and static storage services. As a result, a set of diagrams, mathematical models, and theoretical descriptions is provided to simplify implementation of the said model in modern distributed systems.

The proposed model heavily relies on the semantic feature extraction capabilities of both decision trees and the neural networks. Even though LightGBM and Distilbert are both regarded as extremely fast models, tuned towards performance and memory usage, it is still a case that the execution of the trained model takes a significant amount of resources. That is why the proposed protection model focuses on the logical optimization based on global and internal timeframe discretization and statistical methods such as EWMA to reduce as much as possible the number of execution calls of the pretrained models for the semantic sampling.

To address the issue of efficiency with static data warehouse analysis, the proposed model utilizes global and internal discretization of the timeframes. This approach allows it to coordinate its operation in a batch-oriented way. The scalability of such a solution is hence

ISSN 2707-1758

possible by leveraging deterministic scheduling properties. Each monitoring node, knowing its own position in a replica, can effectively determine its assigned timeframes. In that context, the integration of RSDP provides an essential capability to organize a deterministic approach by synchronizing states between cluster nodes.

While addressing streaming services, this paper considers thoroughly the capabilities provided by queueing protocols such as AMQP. The introduction of middleware queues allows us to analyze suspicious messages in real-time. Additionally, the proposed scaling model involves logical extensions for exchanges that allow for statistical analysis and avoid redundant verification logic invocation. This approach significantly enhances the potential efficiency of the proposed model.

To summarize, the proposed model provides a solid and efficient theoretical foundation for managing intelligent threats towards digital and processing resources. Its implications necessitate continuous monitoring of emerging zero-day attacks that may easily bypass modern security measures. It is the intention of this article to inspire further empirical research, impact assessment of distributed gradual degradation attacks and their mitigation methods.

Authors' contribution: Maksym Kotov – conceptualization, methodology, formal analysis, development of software; Serhii Toliupa – analysis of sources, preparation of a literature review and theoretical foundations of research, editing and reviewing.

References

Adoma, A. F., Henry, N.-M., & Chen, W. (2020). *Comparative analyses of Bert, Roberta, Distilbert, and Xlnet for text-based emotion recognition*. *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP),* 117–121. IEEE. https://doi.org/10.1109/ICCWAMTIP51612.2020.9317379

Arora, R., Basu, A., Mianjy, P., & Mukherjee, A. (2018). *Understanding deep neural networks with rectified linear units.* https://doi.org/10.48550/arXiv.1611.01491

Box, G. E. P., & Pierce, D. A. (1970*).* Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, 65(332), 1509–1526. https://doi.org/10.1080/01621459.1970.10481180

Buchyk, S., Shutenko, D., & Toliupa, S. (2022*).* Phishing attacks detection. *CEUR Workshop Proceedings*, 3384, 193–201. https://ceur-ws.org/Vol-3384/Short_7.pdf

Buchyk, S., Toliupa, S., Buchyk, O., & Shevchenko, A. (2024). *Method for detecting phishing sites*. In A. Luntovskyy, M. Klymash, I. Melnyk, M. Beshley, & A. Schill (Eds.). *Digital ecosystems: Interconnecting advanced networks with AI applications. TCSET 2024. Lecture notes in electrical engineering*, 1198. Springer, Cham. https://doi.org/10.1007/978-3-031-61221-3_15

Büyüköz, B., Hürriyetoğlu, A., & Özgür, A. (2020). Analyzing ELMo and DistilBERT on socio-political news classification. *In Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020* (pp. 9–18). European Language Resources Association (ELRA).

Cox, D. R. (1961). Prediction by exponentially weighted moving averages and related methods. *Journal of the Royal Statistical Society. Series B (Methodological*), 23(2), 414–422. https://doi.org/10.1111/j.2517-6161.1961.tb00424.x

Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Computer Networks*, 44(5), 643–666. https://doi.org/10.1016/j.comnet.2003.10.003

Hernández-Castro, C. J., R-Moreno, M. D., Barrero, D. F., & Gibson, S. (2017). Using machine learning to identify common flaws in CAPTCHA design: FunCAPTCHA case analysis. *Computers & Security*, 70, 744–756. https://doi.org/10.1016/j.cose.2017.05.005

Hunter, J. S. (1986). The exponentially weighted moving average. *Journal of Quality Technology*, 18(4), 203–210. https://doi.org/10.1080/00224065.1986.11979014

Kotov, M., Toliupa, S., & Nakonechnyi, V. (2024). Replica state discovery protocol based on advanced message queuing protocol. *Cybersecurity: Education, Science, Technique*, 3(23), 156–171. https://doi.org/10.28925/2663-4023.2024.23.156171

Kovács, Á., & Tajti, T. (2023*).* CAPTCHA recognition using machine learning algorithms with various techniques. *Annales Mathematicae et Informaticae*, 58, 81–91. https://doi.org/10.33039/ami.2023.11.002

Leevy, J. L., Hancock, J., Zuech, R., & Khoshgoftaar, T. M. (2020). Detecting cybersecurity attacks using different network features with LightGBM and XGBoost learners. *IEEE Second International Conference on Cognitive Machine Intelligence (CogMI)*, 190–197. IEEE. https://doi.org/10.1109/CogMI50398.2020.00032

Lucas, J. M., & Saccucci, M. S. (1990*).* Exponentially weighted moving average control schemes: Properties and enhancements. *Technometrics*, 32(1), 1–12. https://doi.org/10.1080/00401706.1990.10484583

Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *SIGCOMM Computer Communication Review*, 34(2), 39–53. https://doi.org/10.1145/997150.997156

Na, D., Park, N., Ji, S., & Kim, J. (2020). CAPTCHAs are still in danger: An efficient scheme to bypass adversarial CAPTCHAs. In I. You (Ed.). *Information security applications. WISA 2020. Lecture Notes in Computer Science*, 12583. Springer, Cham. https://doi.org/10.1007/978-3-030-65299-9_3

Nelson, B. K. (1998). Time series analysis using autoregressive integrated moving average (ARIMA) models. *Academic Emergency Medicine*, 5(7), 739–744. https://doi.org/10.1111/j.1553-2712.1998.tb02493.x

Prajapati, A. (2021). AMQP and beyond. *In 2021 International Conference on Smart Applications, Communications and Networking (SmartNets).* Glasgow, United Kingdom. https://doi.org/10.1109/SmartNets50376.2021.9555419

Srivastava, A., Gupta, B. B., Tyagi, A., Sharma, A., & Mishra, A. (2011*).* A recent survey on DDoS attacks and defense mechanisms. In D. Nagamalai, E. Renault, & M. Dhanuskodi (Eds.). Advances in parallel distributed computing. PDCTA 2011. *Communications in computer and information science*. *Vol. 203*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24037-9_57

Toliupa, S., Buchyk, S., Shabanova, A., & Buchyk, O. (2023). The method for determining the degree of suspiciousness of a phishing URL. *CEUR Workshop Proceedings*, 3646, 239–247.

Zargar, S. T., Joshi, J., & Tipper, D. (2013*).* A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys & Tutorials*, 15(4), 2046–2069. https://doi.org/10.1109/SURV.2013.031413.00127

Zhang, B., Zhang, T., & Yu, Z. (2017*). DDoS detection and prevention based on artificial intelligence techniques*. In 2017 3rd IEEE International Conference on Computer and Communications (ICCC) (pp. 1276–1280). IEEE. https://doi.org/10.1109/CompComm.2017.8322748

Zhao, G., Wang, Y., & Wang, J. (2023). Intrusion detection model of Internet of Things based on LightGBM. *IEICE Transactions on Communications*, E106–B(8), 622–634. https://doi.org/10.1587/transcom.2022EBP3169

Отримано редакцією журналу / Received: 17.10.24
Прорецензовано / Revised: 15.10.24
Схвалено до друку / Accepted: 30.11.24

ISSN 2707-1758

Сергій ТОЛЮПА, д-р техн. наук, проф.
ORCID ID: 0000-0002-1919-9174
e-mail: tolupa@i.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Максим КОТОВ, асп.
ORCID ID: 0000-0003-1153-3198
e-mail: maksym_kotov@ukr.net
Київський національний університет імені Тараса Шевченка, Київ, Україна

МОДЕЛЬ ЗАХИСТУ ВІД РОЗПОДІЛЕНИХ АТАК ПОСТУПОВОГО ВИСНАЖЕННЯ РЕСУРСІВ, ЗАСНОВАНА НА СТАТИСТИЧНИХ І СЕМАНТИЧНИХ ПІДХОДАХ

В с т у п . *Нині кожен критично важливий сектор соціальних інституцій виконує свої операції на основі розподілених систем оброблення. Сучасна цифрова інфраструктура у своїй роботі значною мірою покладається на дані, що надають користувачі. В результаті, розподілені атаки на основі ботнетів перебувають у безперервних "перегонах озброєнь" із методами захисту, які фільтрують надходження шкідливих даних. Методи протидії часто покладаються на евристичні способи перевірки, орієнтовані на людину. З появою нових досягнень у сфері штучного інтелекту, такі атаки набувають додаткові шляхи досягнення своїх цілей. Успішне виконання зазначеного плану може призвести до поступового виснаження ресурсів цільової системи. Метою цього дослідження є намагання уникнути таких загроз за допомогою поєднання статистичних і семантичних підходів.*

М е т о д и . *Це дослідження проводить теоретичний аналіз і систематизацію розподіленої атаки поступового виснаження ресурсів у розподілених системах і її значення в контексті технологій штучного інтелекту, що розвиваються. Математичне моделювання використовують для визначення властивостей запропонованої моделі захтсту, процесу її інтеграції та виконання. Запропонована модель значною мірою покладається на статистичні методи для аналізу часових рядів та їхніх відхилень, а також класифікаційні нейронні мережі для семантичного виявлення підозрілої поведінки.*

Р е з у л ь т а т и . *У результаті цього дослідження розроблено нову модель, яка використовує статистичну та семантичну перевірку для виявлення аномалій. Процес безперервного моніторингу оптимізований для високонавантажених систем із постійним шквалом потоків даних.*

В и с н о в к и . *Оскільки розподілені атаки можуть бути оснащені інтелектуальними засобами для обходу існуючих заходів безпеки, то розроблення моделі захисту від потенційних витоків ресурсів набуває актуальності. Відомий нещодавній успіх у розробленні штучного генеративного інтелекту викликає занепокоєння щодо безпеки й адекватності поточних заходів безпеки проти векторів розподілених атак на основі автоматизації. Часто буває так, що моделі захисту налаштовані на запобігання нападу, а не на відновлення. Цей підхід, що орієнтований на джерело збитків, часто призводить до проєктних рішень без урахування потенційних результатів успішного порушення. Запропонована модель забезпечує теоретичну основу для створення систем, які одночасно реагують на активне виконання загроз і виконують механізми відновлення, припускаючи, що атака потенційно може обійти початкові заходи безпеки.*

К л ю ч о в і с л о в а : *розподілені системи, атаки поступового виснаження ресурсів, виснаження ресурсів, статистичний аналіз, семантичні підходи, стійкість, LightGBM, Distilbert, EWMA.*

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758