Zero Trust-NIDS: Extended Multi-View Approach for Network Trace Anonymization and Auto-Encoder CNN for Network Intrusion Detection

1st Abeer Z. Alalmaie
School of Electrical and Data Engineering
University of Technology Sydney
Sydney, Australia
abeer.z.alalmaie@student.uts.edu.au

2nd Priyadarsi Nanda
School of Electrical and Data Engineering
University of Technology Sydney
Sydney, Australia
priyadarsi.nanda@uts.edu.au

3rd Xiangjian He
Computer Science
University of Nottingham
Ningbo, China
sean.he@nottingham.edu.cn

***Abstract*—As the enterprise networks are being constantly targeted by sophisticated cyber threats, Zero Trust Security has been suggested to address existing threats. Zero Trust Security models have been recently proposed for outsourcing network security monitoring to third-party analysts. Therefore, the current trends of security monitoring needs to shift to "Never Trust, Always Verify". There are no concerns about analysis accuracy, if a zero trust model is resistant against security attacks. In this paper, a modified multi-view approach is proposed to preserve privacy in network traces, emphasizing the challenges needed to be tackled. We then extend the multi-view approach for the features that are not in the known list of the analyzer and extend the partitioning methods to a more balanced approach. In addition, in order to send any data to the analyzer, we propose to use an Auto-Encoder Convolutional Neural Network, which has the ability to receive any type of input attributes for detecting intrusive behavior. Our proposed multi-view approach outperforms existing works and improves efficiency by improving indistinguishability and preserving privacy for any attributes. The proposed Intrusion Detection System also outperforms existing works by up to 1% higher accuracy without any need for feature engineering.**

***Index Terms*—Zero Trust Security, Cyber Security, Network Trace Anonymization, Network Intrusion Detection, Deep Neural Networks**

I. Introduction

As network security monitoring grows more sophisticated, there is an increasing need for outsourcing such tasks to third-party analysts, e.g., Managed Security Service Providers (MSSPs) [1]. Organizations are usually reluctant to share their network traces due to lack of trust with the analysts because of concerns over sensitive information appearing in the traces, e.g., compromising network and system configuration, which may potentially be exploited for attacks [2]. Networks are vulnerable to data security breaches and unauthorized access due to their "Implicit Trust" or "Trust but Verify" characteristics. Today, the cloud is the "New Network Edge" and it cannot be structured under this old implicit trust model. Implicit trust networks do not work in a climate where network "edges" have broken down and disintegrated.

Specifically, such security breaches result in breaches of confidential databases, with the number of breaches growing both in number and severity. Therefore, the current trend of security monitoring with a "Trust but Verify" approach is not sufficient in any scenario when third parties are involved. In this work, we address such issues when a data owner outsources his/her private network traces to a third-party Intrusion Detection Systems (IDS) to monitor and detect any suspicious incident in real-time or retroactively. This approach assumes that networks are segmented, and that data centre architecture could create a boundary, or "Demilitarized Zone," between trusted and untrusted portions of networks.

Consequently, some anonymization techniques are required to be applied over traces when data owners share their network traces [1]. On the other hand, making the cloud and networks more reliable is a monumental task, and the best course of action would be to eliminate the idea of trust in itself, called Zero Trust Security [3]. In this context, we first clarify Zero Trust Security methods and then clarify anonymization methods, which are used for enhancing security.

Zero Trust is a security model, a set of system design principles, and a coordinated cybersecurity and system management strategy based on an acknowledgment that threats exist both inside and outside traditional network boundaries. The Zero Trust security model eliminates implicit trust in any one device, user, or application and instead requires continuous verification through real-time information fed from multiple sources to determine access and other system responses. Embracing Zero Trust means shifting this perspective from "Never Trust, Always Verify" to treating every user, device, application/workload, and data flow as untrusted. Authenticate and explicitly authorize each for the least privilege required using dynamic security policies. The Zero Trust security model assumes that a cyber attack is inevitable or has likely already occurred, so it constantly limits access to only what is needed and looks for anomalous or malicious activity. In a Zero Trust Security model, none of the pillars, such as users, devices, applications, and packets, are trusted-no matter what type of entity-even if it is part of the network itself [4]. Deny by default and heavily scrutinize all users, devices, data flows, and requests for access are some of the schemes applied in prac-

tice. Log, inspect, and continuously monitor all configuration changes, resource accesses, and network traffic for suspicious activity. Zero Trust Security therefore offers an opportunity to provide a scalable security architecture across an organization to protect against dynamic threat environments.

A lot of solutions have been proposed to address such security issues pertaining to ignoring the zero-trust model (e.g., encryption, perturbation, and poisoning). Most of them immolate the utility (the accuracy of the outsourced data) to provide better privacy guarantees, while others immolate the privacy to preserve the utility. CryptoPAn (Cryptography-based Prefix-preserving Anonymization) is a well-known technique for anonymizing network traces and anonymizing IP addresses while preserving their subnet structure. It replaces real IP addresses inside network flows with a prefix preserving pseudonyms.For this method, the hierarchical relationships among IP addresses with the same prefixes are preserved to facilitate analysis [5]. In other words, any two IP addresses that share a prefix in the original trace will also do so in the anonymized trace [1]. It has been demonstrated that there is a trade-off between the level of attack mitigation achieved by a zero-trust model and the level of the utility that can be preserved about the original data (zero-trust/utility trade-off) [1].

However, CryptoPAn is vulnerable to fingerprinting and injection attacks, where some network flows in the original traces are already known by adversaries or some forged flows have been deliberately injected into traces [6]–[9]. The knowledge of adversaries can be extrapolated to recognize other flows based on the shared prefixes by recognizing known unchanged fields of the flows, namely, fingerprints (e.g., timestamps and protocols) in the anonymized traces [6]. In the following paragraphs, some examples of these attacks are explained in detail.

A. Example 1

Consider the upper part of the table at Fig. 1 as original trace, while the anonymized trace using CryptoPAn is shown in the lower table. In this example, only source IPs are anonymized, which are highlighted, based on their similar prefixes. So, similar prefixes have similar shading.

Using injection or fingerprinting attacks, the entire CryptoPAn methodology can be known by probing all subnets of the network [4].

An Example of an Injection attack [1], as shown in Fig. 1, has the following steps. (1) Three network flows have been injected by an adversary, shown as the first three records of the original trace. (2) The adversary can recognize three injected flows in the anonymized trace by unique and unchanged attributes such as start time and source port. (3) The adversaries' knowledge can be extrapolated from the injected flows and according to the anonymized flows. As an example, since prefix 159.61 is shared by the second (injected), fifth (real) and sixth (real) flows, the adversary knows that all three must also share the same prefix in the original trace. Such identified relationships between flows in the two traces will

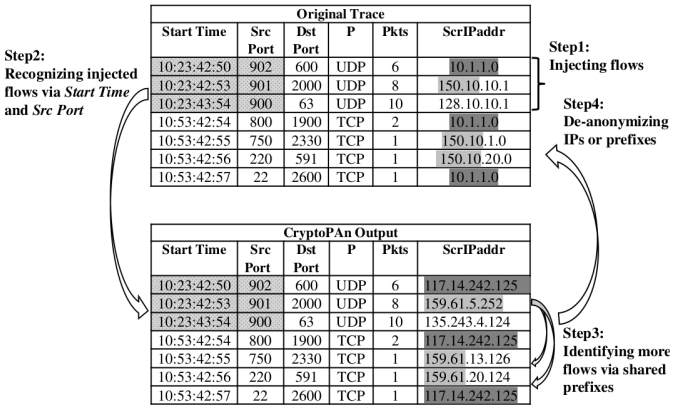


Fig. 1: Injection attack of CryptoPAn [1]

be called "matches" from now on. (4) Since the original IPs of the injected flows are known by the adversary, he/she can infer the prefixes or entire IPs of those anonymized flows in the original traces, e.g., the fifth and sixth flows must have the prefix *150.10*, and the IPs of the fourth and last flows must be *10.1.1.0*.

B. Example 1 Solutions

Most solutions proposed to handle the problem mentioned in Example 1 (section I-A), either require heavy data sanitation or can only support limited types of analysis [1].

The multi-view approach is proposed to handle these limitations [1]. In multi-view, IPs are anonymized by a prefix-preserving anonymization method (i.e., CryptoPAn) and are named" real views, hidden among $N-1$ other fake views. As a result, even with prior knowledge or semantic attacks, adversaries are unable to distinguish the true IPs (views) among those N views. We believe that a more powerful adversary, who can probe all the views of the subnets of a network using injection or fingerprinting, can potentially deanonymize the entire output via a more sophisticated frequency analysis attack.

Consequently, we propose a zero-trust security model, that can pass the three concepts of least privilege & access control, inspect & log traffic and ensure secure access, and also accurate enough. The proposed zero-trust model is based on an extended Multi-View (MV) approach and can be used for outsourcing network traces for Network Intrusion Detection (ZTA-NID). A novel deep neural network is used as an NID to improve fine-grained and automatic authorization processes, which improves efficiency. To the best of our knowledge, this is the first time MV is used as a part of Zero-Trust Security.

Our main contributions are as follows. 1- We extend the multi-view network trace anonymization approach from only IPs to other attributes, and try to make partitions more balanced. 2- We propose a new deep neural network model for Network Intrusion Detection, including an Auto-Encoder and a Convolution Neural Network. The proposed method outperformed other related work.

The rest of the paper is organized as follows. In Section 2, we describe the Zero Trust Security Model. In section 3, the proposed method and contributions of this paper are described in three sub-sections. In section 4, experiments and results are reported as well as training information and testing conditions. Finally, in section 5, the results and conclusions are discussed.

II. Zero Trust model

The Zero Trust Security model was proposed in 2010, and the main idea is to eliminate trust [3]. Zero Trust Architecture totally redefines the approach to resource segmentation – a core principle where resources that must be kept protected are grouped together and isolated safely or kept separately to keep unauthorized access at bay. According to Assuncao, Zero Trust models also offer the opportunity to micro-segment networks, enabling organizations to adapt to their needs without restructuring their entire network [10].

Assuncao [10] pointed out that, by redefining the network, Zero Trust models create new opportunities for the segmentation gateway. It concentrates all the resources of the modern network – from content to filtering, access control, firewalls, cryptography, and package forwarding. The Zero Trust segmentation gateway model can be considered the next generation model of a firewall, increasing micro segmentation of the networks, while offering versatility, scalability, and providing the benefits of virtualization-friendliness. No model is perfect, nonetheless, it is possible to reduce the impacts of attack by using Zero Trust methods [10].

Puthal [4] posits that there are three tenants of Zero Trust Security: all resources must be accessed regardless of location; Access control is on a need-to-know basis and it is strictly enforced; organizations must inspect and log all traffic to verify that users are doing the right thing [4].

A number of Zero Trust Security models have been designed in recent years in an attempt to create a more mirco-segmented network infrastructure in the face of demilitarization, cloud computing, the Internet of Things (IoT), and Bring Your Own Device (BYOD) capabilities. In the following sections, we will review four of the most promising works and compare them in terms of least privilege & access control, inspect & log traffic, ensure secure access.

Puthal [4] proposed a Zero Trust Security model based on Software Defined Perimeters (SDP). Based on this model, SDP counteracts cyber threats in networks by creating a dynamic safety perimeter in any part of the data center. The SDP does not allow any availability or visibility to the users until they have been authenticated to use authorized applications [4].

DeCusatis [2] describes a model for creating Zero Trust networks through first packet-based authentication, defending a Software Defined Network (SDN) controller against cyber-attacks. The author's rationale behind this Zero Trust model revolves around how fine grain segmentation can help improve management visibility, making it possible to stop attacks at the earliest instance possible. Their research identifies how this approach is compatible with micro segmentation [2].

Eidle [11] proposed Autonomic Security for Zero Trust Architecture paper posit that automation or "Autonomic," offers an effective solution. Further, a number of studies leading up to this study show the requirement for security systems that reduce response times and improve the effectiveness of cyber security defences [11].

Eidle [11] conducted trials that identified management where automated threat response and packet-based authentication were linked with the dynamic management of eight distinctive trust models. Then, log parsing orchestration software is tested with open-source log management tools, ultimately coordinating threat responses from network devices like firewalls and authentication gateways. Coordination and integration of responses from multiple network devices. Eidle [11] argue that this particular approach is effective at streamlining the detection and mitigation of cyber threats, therefore, offering a more effective response, specifically to DDoS attacks and a host of other major cyber threat use cases [11].

Puthal and DeCusatis [2], [4] used similar components – controller, client, and gateway. Both approaches also authenticate first before giving users access. However, Puthal focuses on the transport layer, while DeCausatis focuses on the network layer. They also used different technologies. While DeCusatis [4] used Transport Access Control (TAC) technologies, Puthal [2] focused more on the network perimeter. Table I illustrates the benefits of each model, in relation to the core Zero Trust principles outlined by Modderkolk [12]. Based on these principles, the Autonomic Security and Fine-Grained Big Data Security models were the most effective, due to their ability to inspect and log traffic at various different levels of security. Both models also tick all three crucial boxes of important Zero Trust core principles: secure access to all resources within the network; following a least privilege approach and carrying out strict access control; as well as working on the principle of logging and inspecting all traffic.

According to Modderkolk [12], there are three main challenges in creating Zero Trust Security Models. (1) The extremely diverse nature of enterprises. Enterprises are dynamic and can consist of dozens, or even hundreds, of employees with different roles. Therefore, there is no one- size-that-fits-all approach for ensuring cyber security against cyber threats. (2) Comparison between models. It will be challenging to analyze current models to locate cyber security measures that fit into the Zero Trust philosophy. (3) Finding enterprises who wish to participate a case study embracing a Zero Trust Security model is a large undertaking for any enterprise.

Based on the Zero Trust Security models researched, there is a further possibility to consider as to how to build a more efficient authentication system that can be used on a wider scale. None of the models were able to block all kinds of cyber attacks, despite their major improvement on traditional cyber security philosophies, indicating that new models can offer significant improvements.

We propose to leverage a recently proposed reliable notion called "the multi-view" approach for implementing a

TABLE I: Comparison of Software-Defined Perimeter, Transport Access Control and First Packet Authentication, Fine-Grained Big Data Security and Autonomic Security models in relation to the core Zero Trust principles

Concept	Software-Defined Perimeter (SDP)	Transport Access Control and First Packet Authentication	Fine-grained Big Data Security	Autonomic Security
Least Privilege & access control	x	x	x	x
Inspect & log traffic			x	x
Ensure Secure Access	x	x	x	x

Zero-Trust scenario which preserves both the security and the accuracy of an analysis. We consider the importance of implementing an effective zero-trust solution for the scenarios, where organizations outsource their data to third-party intrusion detection systems.

In the following, we propose a new Zero-Trust model based on a multi-view approach and use a deep Auto-Encoder and convolutional neural network for intrusion detection.

III. Proposed Method

Our proposed Zero-Trust Security model provides enhanced security and accuracy wherein it does not trust any inside or outside request (like third-party analysis) by using MV and NID.

In this section, we introduce the proposed methods to enhance the multi-view approach for network trace anonymization and a robust and efficient deep neural network for Network Intrusion Detection (NID).

A. Enhance multi-view network trace anonymization

We propose to enhance the multi-view approach for network trace anonymization using any fields that do not need to be known by the analyzer, as well as balancing the partitioning step of multi-view. The multi-view approach previously applied to IPs has shown good performance. However, we believe that extending the multi-view approach to the other fields can enhance the privacy against very strong adversaries and enable all data to be sent to the analysts. The key idea is to give the data owner the ability to send any available information to the third-party analysts such that they can analyze the data, which has been anonymized and is sufficiently indistinguishable.

The changed building blocks of the modified multi-view mechanism includes the iterative multi-field CryptoPAn and a mixed partition-based prefix preserving anonymization.

1) The iterative CryptoPAn: The multi-view approach applies a prefix - preserving function on the IPs iteratively to generate each view of the multi-view. In addition, we propose to apply different prefix-preserving functions over multiple fields iteratively. Moreover, the analyst will invert these functions in order to obtain the real views (against fake views). Therefore, how these functions can be reversed iteratively is an important issue which is discussed in [1].

Since the result of applying this prefix-preserving function iteratively yields valid values, we can apply cryptoPAn over any data field. For simplicity, we only consider ports as the second field of anonymization.

Specifically, let us denote by PP_{ij} (a,b,K_i,K_j) ($i > 1$, $j > 1$) the iterative application of PP on IP address a and port b using key K_i and K_j correspondingly, where j is the number of iterations for ip anonymization and i is the number of iterations for port anonymization. For example, if we set $i = 2$ and $j = 3$, we have (1).

$$PP_{2,3}(a, b, K_i, kj) = [PP(PP(a, K_i), K_i), PP(PP(PP(b, K_j), K_j), Kj)] \quad (1)$$

It can be easily verified that given any two IP addresses a_1 and a_2 sharing k-bit prefix will always result in two IP addresses that also share a k-bit prefix (i.e., PP_i is prefix-preserving). More generally, the same also holds for applying PP under a sequence of indices and keys (for both IPs), e.g., $PP_i(PP_j(a, K_0), K1)$ and $PP_i(PP_j(b, K_0), K_1)$ will also share k-bit prefix. Finally, for a set of IP addresses S, iterative PP using a single key k satisfies the following associative property.

2) Partition-based prefix preserving anonymization using balanced partitions: The main idea of multi-view is to divide the network traces into partitions where anonymization is iteratively applied to each partition with a different number of iterations. In multi-view approach, traces are only partitioned based on source IPs in two ways: IP-based Partitioning Approach, and Multi-view Using N Key Vectors and both have some weaknesses [1]. In our implementation, we have found that the second method (N Key Vectors) is not balanced in the number of items in each partition since there are many more IPs in some ranges and there are none in others.

We propose to use a mixed partitioning method that is nearly balanced to mitigate the mentioned drawback. In our approach, traces are partitioned not only based on source IPs but also based on all other attributes that are going to be anonymized. In the following, we first mention two partitioning methods and their weaknesses that have been proposed previously.

1. *IP-based Partitioning Approach:* in this method, the original traces are divided in a way that all the IPs sharing prefixes will always be placed in the same partition. Although, adversaries cannot identify the real view, they may choose to live with this fact and attack each partition inside any (fake or real) view instead.

This approach is only designed to prevent attacks across different partitions, and each partition itself is essentially still the output of CryptoPAn and thus still inherits its weaknesses.

2. *Multi-view Using N Key Vectors:* This method is used to mitigate the weakness of an IP-based Partitioning approach; however, it sacrifices some indistinguishability. In this approach, the total IP ranges are divided into five equal parts.

However, since IP addresses are not evenly distributed, many IPs with the same prefixes may get into the same partition. While there may be some parts without any IP,in the following, we attempt to make it more balanced.

3. *mixed Balanced partitioning using multiple Key Vectors* This method is used to further mitigate the weakness of Multi-view Using N Key Vectors approach. We propose to use N_a key vectors for partitioning traces based on attribute a. For example, if we have two sets of vectors (N_{ip} and N_{port}), we divide all traces based on both IPs and ports. As a result, we have partitioned and anonymized both attributes. However, dividing the partitions using any attribute (i.e., IP) can still be unbalanced. For example, we may have a lot of IPs in the range of (100, 1000) and no IPs are available in the range of (1000,10000). In order to handle this issue, we propose to use Na percentiles over attribute a in training dataset for dividing each attribute. As a result, we have balanced partitions where the number of traces in each partition are nearly the same and all acceptable values are covered.

The original trace should be evenly divided in such a way that all the anonymized attributes sharing prefixes will always be placed in the same partition. This will prevent the attackers from identifying the real view by observing multiple shared prefixes across one partitions [1].

We can use the modified MV without worrying about feeding any personal data to the NID, which is introduced in the next subsection.

B. Network Intrusion Detection

We propose to use a deep Auto-Encoder (AE) Convolutional Neural Network (CNN) for NID. Although CNNs and Recurrent Neural Networks (RNN) have been proposed for NID [13]–[15], CNNs work well with data that has a spatial relationship (e.g. images) [16], and RNNs work well when predicting sequences of the same source (e.g. speech) [17], [18]. However, since network trace attributes have different meanings (e.g. ip, port, etc.) and different structures (string, binary, numerical), they are not spatially related. On the other hand, since they may not come sequentially from the same source, they may not be related sequentially. As a result, using a method that can combine and compress attributes in a record of data can be beneficial.

We propose to use a deep AE to extract a combined and compressed feature from network trace attributes. Then, we can use these integrated features for classification, using CNNs. Since these features are higher level features and continue, we can consider them spatially related.

1) Auto-Encoder feature extraction: Network trace related bottleneck features are extracted using the deep AE. The bottleneck layer in AEs contains the latent variables, which forces a compressed knowledge representation of the original input. An AE consists of two components: the encoder and the decoder, as well as a loss function to compare the output with the ground truth (target). The output of the encoder part is named ”bottleneck” and can be used as a compressed feature,it represents a compressed view of the knowledge

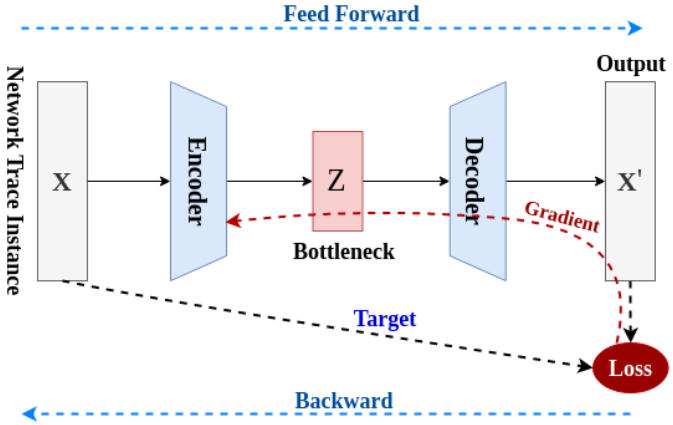


Fig. 2: Auto-Encoder for network traces

of the input. Consequently, a deep auto-encoder is used to extract a combined and compressed feature from network trace attributes.

In AEs, both the encoder and the decoder consist of fully-connected layers, where the activation for each layer can be different. The encoder function ψ maps the original data X to a latent space z, which is named bottleneck. The decoder function ϕ maps the latent space z to the output $\hat{X}$, where the output is expected to be the same as the input. Thus, the algorithm tries to reconstruct the original trace after some generalized nonlinear compression, as shown in (2).

$$\begin{aligned} \psi = X \Rightarrow z \\ \phi = z \Rightarrow \hat{X}, \\ \psi, \phi = argmin||\hat{X} - (\phi(\psi(X)))||^2 \end{aligned} \tag{2}$$

Auto-Encoders are trained to minimize the loss function, which is a reconstruction error, such as mean squared error (mse) (3)

$$L(X, X') = \|X - \hat{X}\|^2 = \|X - \sigma(W_0(\sigma(WX + b)) + b_0)\|^2 \tag{3}$$

where $X - \hat{X}$ is usually averaged over a mini-batch input training set. W, W_0 are weight matrices and b, b_0 are bias vectors for encoder and decoder, respectively. The encoder and decoder parameters can be trained independently, or the decoder parameters can be considered as encoder transposed parameters. In this work, we consider these parts independently and train the whole network at once. Since we want to aggregate input features into the bottleneck layer, our encoder part does not have any bias parameters.

The proposed network structure is shown in Fig. 2, where dotted lines implement the parts that are only used for training the Auto-Encoder. The structure of the proposed Auto-Encoder is shown in Table II.

The bottleneck features of the trained Auto-Encoder, which are spatially related, are used as input to the CNN classifier.

TABLE II: Auto-Encoder structure for network trace feature extraction

Part	Layer	Size
Input	-	196
encoder	Linear + Sigmoid	128
	Linear + Sigmoid (Bottleneck)	64
decoder	Linear + Sigmoid	128
	Linear + Sigmoid	196

TABLE III: Convolutional Neural Network structure for Network Intrusion Detection from bottleneck features

Layer	#Filters	Filter size	Output Shape
Input	1	-	64
Convolution	128	11	54
Pooling	-	2	27
Convolution	128	9	19
Convolution	128	7	13
Convolution	128	5	9
Linear	512	-	512
Linear	128	-	128
Linear	64	-	64
Linear (Classifier)	2	-	2

2) Convolutional Neural Network classifier: In this section, we propose to use a CNN to classify network traces into intrusion and not intrusion from bottleneck features. The Convolutional Neural Network has been proven to be able to automatically detect network intrusions. However, as previously mentioned, CNN works well with data that has a spatial relationship, whereas network trace instances do not present spatial relationships. Consequently, we propose to use a CNN on the bottleneck features extracted from the trained Auto-Encoder.

Our proposed CNN structure is shown in Table III, and $LeakyReLU$ with 0.2 negative slope is considered as activation function for hidden layers. Obviously, $SoftMax$ is used in the output layer that outputs the classifier results.

In the following, we will review the training and test conditions along with the evaluation results.

IV. Experiments and Results

In this section, we first report the experimental results for the modified MV anonymization approach and compare it with previous works. Then, we report the NID accuracy and compare it with the corresponding work. The combination of these two modules can be used as a Zero-Trust model, where the network traces are secured and anonymized by the proposed multi-view approach, and NID can be used by the analyzer to detect intrusions more accurately.

A. Modified multi-view approach results

To validate the modified MV anonymization approach, we use the whole UNSW-NB15 dataset [19]. Since this work is a balanced version of the N Key Vector approach and can affect mostly on indistinguishability, we focus on this metric [1]. We believe that applying MV on the other attributes does not have any effect on the results and only gives the ability to send these attributes to the analyzer.

Indistinguishability is important because it can help us prevent intrusions. Whenever an analyst (adversary) receives N different traces with identical attribute values and different attribute values, his/her goal can be to identify the real view among all the views. For example, he/she may attempt to observe his/her injected or fingerprinted flows, or he/she can launch the aforementioned semantic attacks on those views, hoping that the real view might respond differently to those attacks. Therefore, the main objective of multi-view is to satisfy the indistinguishability property, which means the real view must be sufficiently indistinguishable from the fake views under semantic attacks. We use the ϵ-indisinguishablity property to compare the indistinguishability of different methods [1]. In the following paragraphs, we first review the data properties and configuration, followed by review the evaluation conditions and results.

We have used UNSW-NB15,a network intrusion dataset. The UNSW-NB15 includes 2540047 records, including 49 attributes. It contains nine different attacks, including DoS, worms, backdoors, and fuzzers. For simplicity, the source IP attribute is intended for evaluation and can be expanded to other attributes.

As mentioned by Meisam [1], "the number of views N is an important parameter that determines both the privacy and computational overhead. The data owner could choose this value based on the level of trust in the analysts and the affordable computational overhead.

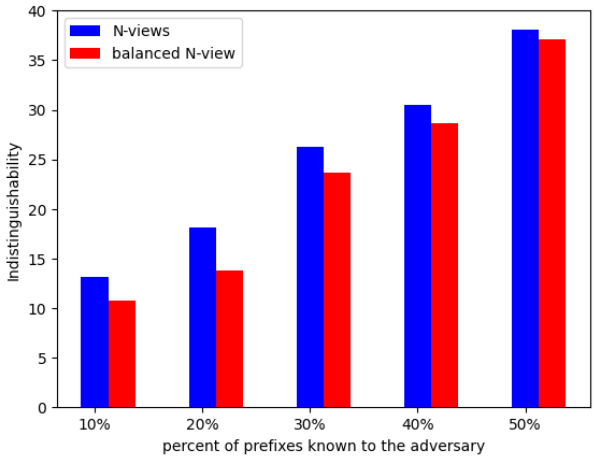
The number of real view candidates is shown to be $e^{\epsilon} * N$. The data owner can first estimate the adversary's background knowledge α (number of prefixes known to the adversary) and then calculate ϵ [1].

The results of indistinguishability for different number of groups are represented as 10, 50 and 136 (the first octet) and different adversary knowledge are shown in Fig. 3. We cannot achieve the results of Meisam's work [1] since our dataset is different. Obviously, using a balanced partitioning method for a small number of groups outperforms the original approach as shown in Fig. 3a, except when the percent of prefixes known to the adversary is high (40% and 50%). Obviously, this improvement gets lower as the number of groups increases (Fig. 3b and Fig. 3c). The balanced N Key Vector approach does not give any improvement for 136 IP groups (first octet), although both methods seem to work in the same way.

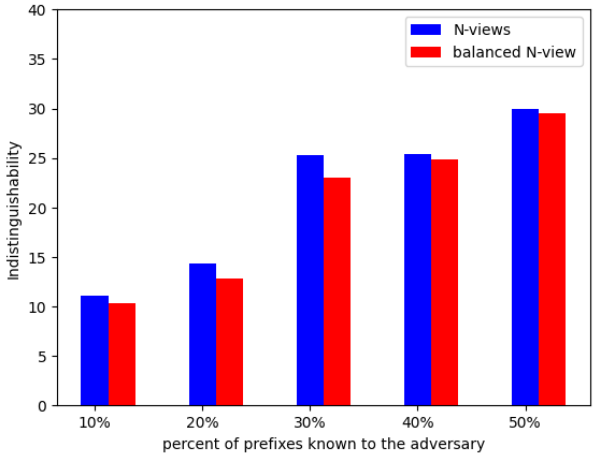
B. Network Intrusion Detection results

To validate the proposed NID, we use the train and test subsets of the standard UNSW-NB15 dataset [19]. As mentioned in Parts II and III-B, we can feed the multi-view generated data to the model. However, we use the standard training and testing subsets for NID validation to compare the results with previous work.

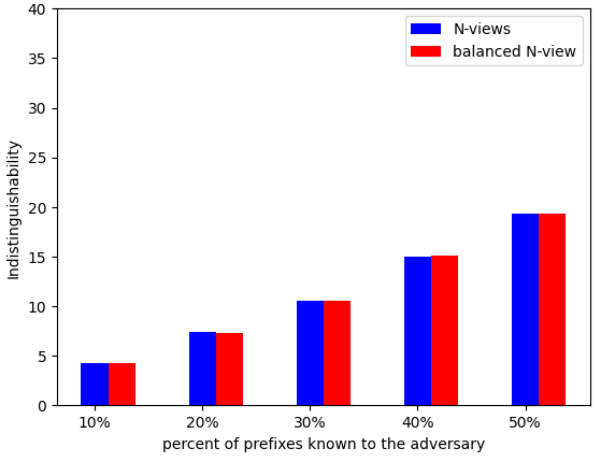
Both AE and CNN models are trained on a single 3060s GPU and are implemented in pyTorch. The auto-encoder is trained with a batch size of 32, an initial learning rate of 1e-4 and mean squared error (mse) loss, and Adam optimizer. The models are trained until no more improvement is possible,



(a) 10 IP Groups



(b) 50 IP Groups



(c) 136 IP Groups

Fig. 3: Indistinguishability comparison of balanced and original N Key Vector approaches for different number of groups and adversary knowledge.

TABLE IV: Accuracy results of Network Intrusion Detection (binary classification) on test data

Method	Accuracy
CNN (raw input)	84%
AE-CNN (proposed method)	92.23%
Decision Tree + XGBoost [21]	90.85%
Tow-stage ensemble [22]	91.27%

according to the validation results. All data attributes are normalized to numerical values between 0 and 1. Thus, non-numerical attributes are converted into numerical values using one-hot encoding.

Bottleneck features extracted from the trained AE are fed into the CNN model for classification. The classifier is trained with a batch size of 32, an initial learning rate of 1e-6 and cross-entropy loss for the binary classifier and Adam optimizer. The accuracy result of the proposed method is reported in IV, and is compared with the best previous work. We tried to train a model like [20], but we could not achieve the reported results. Also, this work has never been mentioned or compared in later works.

1) Network Intrusion Detection Performance: Since the number of parameters of the proposed neural network is low, it is fast and can be used for low - resource scenarios. The proposed AE model has only 33K parameters and the CNN model has 1.01 M parameters. Totally, the NID model has 1.03 M parameters, and it takes 7 milliseconds to infer a sample record on a 1.60 GHz core CPU.

C. Zero Trust Model Comparison and Conclusion

In this subsection, we compare our proposed Zero Trust security model with four Zero Trust models discussed in Section II including Software-Defined Perimeter, Transport Access Control and First Packet Authentication, Fine-grained Big Data Security, and Autonomic Security.

Based on the principles outlined by Modderkolk [12], our proposed model is able to inspect and log traffic automatically using NID. It also provides secure access to the resources within the network by anonymizing any fields that do not need to be known by the analyzer using the proposed multi-view approach. Moreover, it is able to log and inspect all traffic using the combination of both proposed components. Consequently, our proposed model ticks all three crucial boxes of important Zero Trust core principles available in Table I. However, we tried to build a more efficient authentication system that can which can be used on a wider scale using machine learning and artificial intelligence technology.

In our proposed model, we have attempted to improve on fine-grained and automatic authorization processes, which can improve efficiency dramatically.

None of the models were able to block all kinds of attacks, despite their major improvement on traditional cybersecurity philosophies, indicating that new models can offer significant improvements. Therefore, further experimentation with these models opens up the possibility for future improvements is latency and dynamic optimization of other network equipment.

V. Conclusion and Future Works

In this article, we have defined the philosophy and need for a Zero-Trust model. First, we have reviewed the multi-view approach for Network Trace anonymization, which has been proposed previously, and mentioned some drawbacks. Then we have proposed methods to reduce them.

Zero-Trust needs a robust and accurate network intrusion detection system. Consequently, we have proposed a Deep Neural Network model to detect network intrusion.

For validating the proposed methods, we have used the UNSW-NB15 dataset. We have seen that our proposed method for Network Trace anonymization outperformed the original N Key Vector approach for a small number of groups. In addition, our proposed Deep Neural Network, including Auto-Encoder and CNN as feature extractor and classifier respectively for Network Intrusion Detection have outperformed previous works.

For future work, we plan to test the proposed Deep Neural Network for multi-class Network Intrusion Detection. Also, we propose to use Recurrent Neural Network base models to enhance the detector.

References

[1] M. Mohammady, L. Wang, Y. Hong, H. Louafi, M. Pourzandi, and M. Debbabi, "Preserving both privacy and utility in network trace anonymization," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 459–474. [Online]. Available: https://doi.org/10.1145/3243734.3243809

[2] C. DeCusatis, P. Liengtiraphan, A. Sager, and M. Pinelli, "Implementing zero trust cloud networks with transport access control and first packet authentication," in *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, 2016, pp. 5–10.

[3] J. Kindervag, *Build security into your network's dna: The zero-trust network architecture*, 2010.

[4] D. Puthal, S. P. Mohanty, P. Nanda, and U. Choppali, "Building security perimeters to protect network systems against cyber threats [future directions]," *IEEE Consumer Electronics Magazine*, vol. 6, no. 4, pp. 24–27, 2017.

[5] J. Xu, J. Fan, M. Ammar, and S. Moon, "Prefix-preserving ip address anonymization: measurement-based security evaluation and a new cryptography-based scheme," in *10th IEEE International Conference on Network Protocols, 2002. Proceedings.*, 2002, pp. 280–289.

[6] T. Brekne, A. Årnes, and A. Oslebo, "Anonymization of ip traffic monitoring data: Attacks on two prefix-preserving anonymization schemes and some proposed remedies," 05 2005, pp. 179–196.

[7] T. Brekne and A. Årnes, "Circumventing ip-address pseudonymization." 01 2005, pp. 43–48.

[8] T.-F. Yen, X. Huang, F. Monrose, and M. Reiter, "Browser fingerprinting from coarse traffic summaries: Techniques and implications," vol. 5587, 07 2009, pp. 157–175.

[9] M. Burkhart, D. Brauckhoff, M. May, and E. Boschi, "The risk-utility tradeoff for ip address truncation," in *Proceedings of the 1st ACM Workshop on Network Data Anonymization*, ser. NDA '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 23–30. [Online]. Available: https://doi.org/10.1145/1456441.1456452

[10] P. Assunção, "A zero trust approach to network security," in *In Proceedings of the Digital Privacy and Security Conference*, 2019.

[11] D. Eidle, S. Y. Ni, C. M. DeCusatis, and A. Sager, "Autonomic security for zero trust networks," *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 288–293, 2017.

[12] M. Modderkolk, "Zero trust maturity matters: modelling cyber security focus areas and maturity levels in the zero trust principle," *Masters, Department of Information and Computer Science, Utrecht University*, 2018.

[13] V. Ravi, S. Kp, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," 09 2017, pp. 1222–1228.

[14] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep recurrent neural network for intrusion detection in sdn-based networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 202–206.

[15] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for iot intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, p. 102031, 2020, modeling and Simulation of Fog Computing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X19301625

[16] K. Swingler and M. Bath, "Learning spatial relations with a standard convolutional neural network," in *12th International Conference on Neural Computation Theory and Applications*. SCITEPRESS-Science and Technology Publications, 2020, pp. 464–470.

[17] N. Naderi, B. Nasersharif, and A. Nikoofard, "Persian speech synthesis using enhanced tacotron based on multi-resolution convolution layers and a convex optimization method," *Multimedia Tools and Applications*, 2021.

[18] P. Farajiparvar, H. Ying, and A. Pandya, "A brief survey of telerobotic time delay mitigation," *Frontiers in Robotics and AI*, vol. 7, 12 2020.

[19] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[20] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–6.

[21] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset," *Journal of Big Data*, vol. 7, no. 105, 2020.

[22] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "Tse-ids: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94 497–94 507, 2019.