

52 In Network and Distributed Systems Security Symposium (NDSS) 2018, San Diego, February 2018. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. Yisroel Mirsky, Tomer Doits, Haiman, Yuval El ovic i and Asaf Shabtai. arXiv:1802.09089v1 [cs.CR] 25 Feb 2018. Ben-Gurion University of the Negev {yisroel, tomerdoi}@post.bgu.ac.il, {el ovic i, shabtai} @ bgu.ac.il Abstract Neural networks have become an increasingly popular solution for network intrusion detection systems (NIDS). Their capability of learning complex patterns and behaviors make them a suitable solution for differentiating between normal traffic and network attacks. However, a drawback of neural networks is the amount of resources needed to train them. Many network gateways and routers devices, which could potentially host an NIDS, simply do not have the memory or processing power to train and sometimes even execute such models. More importantly, the existing neural network solutions are trained in a supervised manner. Meaning that an expert must label the network traffic and update the model manually from time to time. In this paper, we present Kitsune: a plug and play NIDS which can learn to detect attacks on the local network, without supervision, and in an efficient online manner. Kitsune's core algorithm (KitNET) uses an ensemble of neural networks called autoencoders to collectively differentiate between normal and abnormal traffic patterns. KitNET is supported by a feature extraction framework which efficiently tracks the patterns of every network channel. Our evaluations show that Kitsune can detect various attacks with a performance comparable to offline anomaly detectors, even on a Raspberry PI. This demonstrates that Kitsune can be a practical and economic NIDS. Key words Anomaly detection, network intrusion detection, online algorithms, autoencoders, ensemble learning.

I. INTRODUCTION The number of attacks on computer networks has been increasing over the years [1]. A common security system used to secure networks is a network intrusion detection system (NIDS). An NIDS is a device or software which monitors all traffic passing a strategic point for malicious activities. When such an activity is detected, an alert is generated, and sent to the administrator. Conventionally an NIDS is deployed at a single point, for example, at the Internet gateway. This point deployment strategy can detect malicious traffic entering and leaving the network, but not malicious traffic traversing the network itself. To resolve this issue, a distributed deployment strategy can be used, where a number of NIDSs are connected to a set of strategic routers and gateways within the network. Permission to freely reproduce all or part of this paper for non commercial purposes is granted provided that copies bear this notice and the full citation on the first page. Reproduction for commercial purposes is strictly prohibited without the prior written consent of the Internet Society, the first-named author (for reproduction of an entire paper only), and the author's employer if the paper was prepared within the scope of employment. NDS'18, 18-21 February 2018, San Diego, CA, USA Copyright 2018 Internet Society, ISBN 1-891562-49-5 <http://dx.doi.org/10.14722/ndss.2018.23204> Over the last decade many machine learning techniques have been proposed to improve detection performance [2], [3], [4]. One popular approach is to use an artificial neural network (ANN) to perform the network traffic inspection. The benefit of using an ANN is that ANNs are good at learning complex non-linear concepts in the input data. This gives ANNs a great advantage in detection performance with respect to other machine learning algorithms [5], [6], [7], [8]. The prevalent approach to using an ANN as an NIDS is to train it to classify network traffic as being either normal or some class of attack [6], [7], [8]. The following shows the typical approach to using an ANN-based classifier in a point deployment strategy: 1) Have an expert collect a dataset containing both normal traffic and network attacks. 2) Train the ANN to classify the difference between normal and attack traffic, using a strong CPU or GPU. 3) Transfer a copy of the trained model to the network/organization's NIDS. 4) Have the NIDS execute the trained model on the observed network traffic. In general, a distributed deployment strategy is only practical if the number of NIDSs can economically scale according to the size of the network. One approach to achieve this goal is to embed the NIDSs directly into inexpensive routers (i.e., with simple hardware). We argue that it is impractical to use ANN-based classifiers with this approach for several reasons: Offline Processing. In order to train a supervised model, all labeled instances must be available locally. This is infeasible on a simple network gateway since a single hour of traffic may contain millions of packets. Some works propose offloading the data to a remote server for model training [9], [3]. However, this solution may incur significant network overhead, and does not scale. Supervised Learning. The labeling process takes time and is expensive. More importantly, what is considered to be normal depends on the local traffic observed by the NIDS. Furthermore, in attacks change overtime and while new ones are constantly being discovered [10], so continuous maintainable of a malicious attack traffic repository may be impractical. Finally, classification is a closed-world approach to identifying concepts. In other words, a classifier is trained to identify the classes provided in the training set. However, it is unreasonable to assume that all possible classes of malicious traffic can be collected and placed in the training data. High Complexity. The computational complexity of an ANN Output Layer Ensemble Layer Map score RMSE RMSE RMSE RMSE RMSE RMSE RMSE RMSE RMSE RMSE The reason we use autoencoders is because (1) they can be trained in an unsupervised manner, and (2) they can be used for anomaly detection in the event of a poor reconstruction. The reason we propose using an ensemble of small autoencoders, is because they are more efficient and can be less noisy than a single autoencoder over the same feature space. From our experiments, we found that Kitsune can increase the packet processing rate by a factor of five, and provide a detection performance which rivals other online (batch) anomaly detectors. In summary, the contributions of this paper are as follows: A novel autoencoder-based NIDS for simple network devices (Kitsune), which is lightweight and plug-and-play. To the best of our knowledge, we are the first to propose the use of autoencoders with or without ensembles for online anomaly detection in computer networks. We also present the core algorithm (KitNET) as a generic online unsupervised anomaly detection algorithm, and provide the source code for download. 1 A feature extraction framework for dynamically maintaining and extracting implicit contextual features from network traffic. The framework has a small memory footprint since the statistics are updated incrementally over damped windows. An online technique for automatically constructing the ensemble of autoencoders (i.e., mapping features to ANN inputs) in an unsupervised manner. The method involves the incremental hierarchical clustering of the feature-space (transpose of the unbounded dataset), and bounding of cluster sizes. Experimental results on an operational IP camera video surveillance network, IoT network, and a wide variety of attacks. We also demonstrate the algorithm's efficiency, and ability to run on a simple router, by performing benchmarks on a Raspberry PI.

Fig. 1: An illustration of Kitsune's anomaly detection algorithm. KitNET grows exponentially with number of neurons [11]. This means that an ANN which is deployed on a simple network gateway, is restricted in terms of its architecture and number of input features which it can use. This is especially problematic on gateways which handle high velocity traffic. In light of the challenges listed above, we suggest that the development of an ANN-based network intrusion detector, which is to be deployed and trained on routers in a distributed manner, should adhere to the following restrictions: Online Processing. After the training or executing the model with an instance, the instance is immediately discarded. In practice, a small number of instances can be stored at any given time, as done in stream clustering [12]. Unsupervised Learning. Labels, which indicate explicitly whether a packet is malicious or benign, are not used in the training process. Other meta-information can be used so long as acquiring the information does not delay the process. Low Complexity. The packet processing rate must exceed the expected maximum packet arrival rate. In other words, we must ensure that there is no queue of packets awaiting to be processed by the model. The rest of the paper is organized as follows: Section II discusses related work in the domain of online anomaly detection. Section III provides a background on autoencoders and how they work. Section IV presents Kitsune's framework and its entire machine learning pipeline. Section V presents experimental results in terms of detection performance and run-time performance. Finally, in section VII we present our conclusion. In this paper, we present Kitsune: a novel ANN-based NIDS which is online, unsupervised, and efficient. A Kitsune, in Japanese folklore, is a mythical fox-like creature that has a number of tails, can mimic different forms, and whose strength increases with experience. Similarly, Kitsune has an ensemble of small neural networks (autoencoders), which are trained to mimic (reconstruct) network traffic patterns, and whose performance incrementally improves overtime. i/sj