Proposal

Abstract

With the rapid growth in the scale of networked computers and existing apps, it has become clear that the possible harm caused by launching cyber-attacks is a great risk that has only increased in recent years. For this reason, the Intrusion Detection System (IDS) is a critical security control which can be used to identify advanced and continuously evolving network attacks. For network protection, a Network Intrusion Detection System (NIDS) allows a user to identify and respond to malicious traffic. The biggest advantage of an intrusion detection system is the ability to generate alerts and forward them to the security operations center. Once identified, incident response teams are alerted to investigate a possible attack or network breach. However, traditional signature based IDS lack the ability to identify evolving network attacks. In addition, security tools such as firewalls and anti-virus applications usually do not have access to encrypted keys and also are traditionally signature based, making it difficult or impossible to specifically detect maliciously encrypted files. This study is conducted in order to capture the network traffic using the Zeek open-source platform that ensures effective capturing of traffic by generating logs. The network logs track key activities of the network which can be used to troubleshoot and perform network forensics. The logs display network-related information, fault, and warning events. The captured logs are then processed using machine learning techniques to train a Deep Neural Network. TensorFlow is then used to create an Auto Encoder architecture which is is used to classify malicious network activity based on a network of IoT devices.

Introduction

A powerful and efficient Network Intrusion Detection System (NIDS) is one of the main network security issues. Despite substantial advancement in NIDS technologies, in comparison to anomaly detection techniques, most methods are still using less capable signature-based techniques.

Detection of intruders plays a big part in network protection by engaging system administrators to warn about suspicious behavior such as assaults and malware. IDS is an important defensive line to defend sensitive networks from the ever-expanding problems of attack. Today, there are numerous obstacles to data security analysis due to rapid development in computer networks and applications. Events that may violate the computer systems standards such as availability, authority, secrecy, and integrity can be described as intrusions/attacks. Security strategies and tools are unable to detect new attack conditions and cannot study in-depth network packets. The amount of malware found is expected to expand recently at more than 90% of small- to medium-sized companies (Shekhawat et al., 2019). Real-time network malware identification can minimize malware spread on the network considerably.

Intrusion detection systems and deep packet inspection are used to detect network-based malware. They aggregate packets and the contents are processed in order to search for signatures or other attributes that may be used as harmful or innocuous data. The effect of these approaches was found quite useful as it decreases the malicious activities in the network for a greater extent. However, encrypted, intelligent, and advanced security network attacks are not detected as effective as necessary that leads to higher security breaches. When a network is breached by an unauthorized user or device, a network security violation occurs and when hackers have access to the network anything can be done (Moustafa & Slay, 2015).

If a networking accident is observed, a network administrator must manually check the occurrence and come up with a plan to avoid the occurrence in the future and avoid such events. The study of the network takes time and needs strong network expertise. Creating a solution that automates the function of the supervisor will accelerate the analyzing process significantly and promote the entire process for less-skilled administrators. The aim of this research is to analyze various features commonly used to distinguish malicious network traffic from authenticated packets. The emphasis is again on an overview of functions focused on qualified models of machine learning. In the approach, the data packets are captured using Zeek that is not an integrated monitoring mechanism such as a firewall or a mitigation mechanism for the intrusion. Instead, Zeek sits on a "sensor," program, virtual device, or the cloud platform, which monitors network activity silently and discreetly. Zeek interprets and produces the lightweight, high-fidelity transaction records, file content, and completely customizable output that can be managed on the drive or in a more analyst-friendly platform like SIEM. The best thing about packet capturing using Zeek is: it is an open-source platform that enables the users to capture the network packets effectively.

Once the packets are captured using Zeek open source then logs are going to be converted into panda's data frames. A table like SQL or Excel is like a panda data frame. It is also similar in form and can be used to combine, process, and pivot related operations. However, since Data Frames are built into Python, more sophisticated operations and manipulations that SQL and Excel can provide can be programmed with Python. Especially helpful are data frames that provide an efficient way to analyze the captured packets. A variety of data structures and directories can be loaded by Data Frames including list and dictionary, CSV files, excellence files, and database records (read more about this here).

As the data set is created now it is the processing part that is going to be done with the help of tensor flow. TensorFlow is a large-scale machine-learning open source library for numerical computers. TensorFlow packages a number of models and algorithms of computer learning and profound learning and in this research, the logs are going to be given as input to the tensor flow. The Edge TPU is going to operate neural networks such as neural convolutional networks (CNNs) in detail. It only supports 8-bit quantized and then compiled model TensorFlow Lite explicitly for Edge TPU. A user may not need to adopt this entire method to build a successful model for Edge TPU. By retraining it utilizing its own data sets, a user would be going to be able to take advantage

of current TensorFlow models compatible with Edge TPU. For starters, Mobile-Net is a common Edge TPU-compliant image classification/detection software architecture.

Training the neural grid from scratch will take days to measure time and requires huge numbers of training data (when there are no measured weights or partialities). Transfer learning is going to help to begin with a model that has already been trained for a task and then begin to train the model using a smaller training dataset (Sharafaldin et al., 2018). This is achievable by re-training the entire model (fitting the weights around the whole network), but also by conveniently eliminating the final calibration and training a new layer that understands the new levels. This ensures that you can produce very detailed outcomes. The ultimate aim is to eradicate the problem of current network security risks and threats that cannot be handled with the help of a common intrusion detection system.

Literature Review

The earlier work on network classification has been performed by Cannady and Nziga. The researcher points out that neural networks are rational solutions because they are equipped with representative training sets for a particular problem area. The model does not accommodate streaming data and thus the person who controls our system must delete the data if they need to train the prototype and run it to the modified representative data collection. In addition, a three-layer control feed-forward system was used by the authors to generate a variety of feedback mappings of the information (Cannady, 2009). Consequently, in accordance with the User datagram protocol (UDP), the specific intrusion detection system (IDS) agent can learn how to detect flood-based Denial of Service (ICMP) assaults. Initially, the design explores how ICMP attacks can be observed and then periodically changes and retrains the model. It will also learn how to detect new attacks based on the UDP protocol. In order to propose an online learning process, Cannady implemented the Cerebellar Articulation Controller Neural Network. In order to construct a series of input and output mappings, the Authors introduced a multilayer process. Initially, the device is trained on how to recognize ICMP infringements and how to distinguish attacks based on the UDP protocol through previous knowledge and preparation. However, only in flood-based Denial of Service attacks is the approach oriented. An inspection of the sequence of device calls uses one approach to identify abnormalities in IDS (Nziga et al., 2012).

A textual and experimental contrast has been conducted between Dong and Wang's use of unique conventional NIDS and deep learning methods (Dong et al., 2016). The investigators concluded that the methods of deep learning increased identification precision in a number of test sizes and forms of traffic anomalies. The writers have also shown the potential to solve issues associated with imbalanced datasets by over-sampling, for which they use the SMOTE methodology.

Random forest and regression trees were used by researchers to classify violations of the method (Rong, 2010). The researchers used outliers and anomaly identification in IDS for Principal Component Analysis (PCA). They compared numerous methods of choosing and analyzing

features of identification of anomalies (Bahl et al., 2015). It is an essential task to choose a proper model that helps to evaluate the relationship between precision and complexity and outlines an optimal solution. The observation of sequences of device calls establishes one method for detecting vulnerabilities of host-based intrusion detection systems.

Many of the publications in the IDS research project can be outlined by using supervised or semi-supervised learning structures as issues with learning classifications (Chandola et al., 2009). Although some authors have tried to achieve unattended instruction, poor quality has been achieved. RL has been widely used in fields related to the computer networks but it was not significantly discussed in the field of intrusion detection or intrusion prevention. The fields of routing protocols, authentication procedures, entry management, and service efficiency mechanisms are greatly fascinating to scientists. Attention is paid to the fact that RL is acceptable in control cases where an environmental reaction exists (Sutton, 1998). We detect feedback, which is interpreted as a reward, in all of the cases described above.

In association with Hidden Markov models (HMM), Xu et al. has incorporated reinforcements in the detection of infringements through knowing the probability of state change. The authors proposed that HMM may provide a clear evaluation of state IDS transformations (Xu et al., 2005). To change the value function, time differential methods and outcomes were implemented using the same training and test sets. Two years back, Xu and Luo designed a temporal difference solution to the behavior of the network (Xu et al., 2007). In that job, improved detection precision was achieved, in line with HMM's previous implementation. The kernel uses the least-squares time difference algorithm (LS-TD) to test the value function and apply the parameter reduction (Xu et al., 2005). In order to validate the consistency of the model outlined, Xu and Luo presented the realistic solution for IDS; they used device call traces from the sending mail program Miller and Inoue used a paradigm called the Reinforcement Perceptual intrusion method, which acts on numerous agents (Miller et al., 2003). A single agent may use a map for the detection of fraudulent activity and the results given by all the agents are black-boarded. When a signal within the system is identified, it is transmitted for collective group analysis by all members. They relay votes to the central blackboard device that measures weights and awards agents according to their achievement.

Some of the early literature concerning the use of data streaming was based on the premise that limited and not total data storage is used. The sliding window principle reflects the notion that the most current and new knowledge in the learning process can be identified at any time in a frame. The choice of window size W is a difficulty, so the simplest solution is to let the user pick the size of the window to address it and preserve it while the algorithm is being implemented (Dong et al., 2003). The problem fixes this problem. A score is used to consider the shift between the present and the related frames. This is accomplished by using the ratings. Other recommendations imply that only composite figures with a 'decay function' can be recalled, which signifies that these aggregates have significance over time (Cohen et al., 2003).

For other methods, rates of three operational thresholds such as precision, accuracy, and recoveries are controlled to create the draftiness detection definition. Its values are evaluated on an average, with reference to the confidence intervals for standard sample errors and relevant predictors (use the current set of examples). The basic goal is to evaluate the window size of the data to minimize the expected error in new instances. The technique uses a data collection without etiquette. This means that complicated equations are not required and it is simple to use. Another approach to distinguish distribution changes is to detect the algorithm's real-time error rate. In this way, analysis is carried out in a sequence of studies. If a new instance of experimentation is open, the existing model is labeled. The authors evaluate a *kW* alert and *KD* drift thresholds that are allowed until the error rate exceeds default thresholds. These thresholds act as a recommendation for the instance delivery change (Gama et al., 2004).

Hansman and Hunt implemented a four-dimensional description (Hansman & Hund, 2005). Their classification scheme contains particular categories of infringements that add to protection by the creation of continuity in the vocabulary identifying multiple forms of assaults. They suggested that the method could be strengthened with a solid style with a thorough explanation of the different attack styles. The second dimension stresses the definition of the perpetrator and the third defines the method of describing the different stages of risk. The second dimension is to aid administrators to identify the violation.

The authors of the (Etienne, 2009) deep packet inspection use the techniques of conventional patterns or signature-based technologies to detect malicious traffic by analyzing the payload content. Snort, an IDS, detects harmful traffic and is using a signature or string matching the packet contents. This analysis uses the same details. Snort also houses a common rule collection for the IPS (Intrusion Security System). However, only a small percentage of TLS-specific rules in Snort suggest that TLS-related malware related pattern matching technique does not function (Bakhdlaghi, 2017).

BotFinder (2012) is a network data flow analyzer used to detect bot infections. It is a network flow information analyzer. In order to identify anomalies in network activity between two endpoints, the system depends on chronologically-ordered streams (or traces). This knowledge is used in a clustering-based algorithm along with other network metadata. A neural network-basic malware detection approach based on various network flow functions (Prasse, Machlica, Pevn'y, Havelka, & Scheffer, 2017) is developed.

The literature review of different research papers summarizes that current IDSs are not good enough for the advanced level threats to the networks. Further investigation needs to be carried out with regard to the study of players' tactics and methods in accordance with the cybersecurity knowledge system. In addition, the theoretical structure and functional implementation of the potential variance pose distinct problems. It is worth noting that the different variables determining the game have several difficulties quantifying. Using the common security measurement of IDSs

analysis and other associated classically valid methods, the applied research on network security is currently limited. However, the increasingly evolving cyber world and creative policies invented by attackers nowadays must be taken into account in introducing modern and sophisticated approaches. Firewalls and other methods for intrusion prevention can be helpful to our fundamental security but new software and hardware technologies and high-tech technologies are required to allow the manager to respond rapidly and appropriately to each threat. As a result, a systemic and advanced approach is going to be followed in this project that would come with all the problems that are highlighted in the literature.

References

Shekhawat, A. S., Di Troia, F., & Stamp, M. (2019). Feature analysis of encrypted malicious traffic. *Expert Systems with Applications*, *125*, 130-141.

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018, January). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP* (pp. 108-116).

Moustafa, N., & Slay, J. (2015, November). 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)* (pp. 1-6). IEEE.

Cannady, J. (2009, October). Distributed detection of attacks in mobile ad hoc networks using learning vector quantization. In *2009 Third International Conference on Network and System Security* (pp. 571-574). IEEE.

Nziga, J. P., & Cannady, J. (2012, September). Minimal dataset for Network Intrusion Detection Systems via MID-PCA: A hybrid approach. In *2012 6th IEEE International Conference Intelligent Systems* (pp. 453-460). IEEE.

Dong, B., & Wang, X. (2016, June). Comparison deep learning method to traditional methods using for network intrusion detection. In *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)* (pp. 581-585). IEEE.

Rong, Y. (2010). Classification and Regression Trees Random Forest Algorithm. *Machine Learning Approaches to Bioinformatics Science, Engineering, and Biology Informatics*, 120-132.

Bahl, S., & Sharma, S. K. (2015, February). Improving classification accuracy of intrusion detection system using feature subset selection. In *2015 Fifth International Conference on Advanced Computing & Communication Technologies* (pp. 431-436). IEEE.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1-58.

Sutton, R. S. (1998). andA. G. Barto,“Reinforcement learningan introduction”.

Xu, X., & Xie, T. (2005, August). A reinforcement learning approach for host-based intrusion detection using sequences of system calls. In *International Conference on Intelligent Computing* (pp. 995-1003). Springer, Berlin, Heidelberg.

Xu, X., & Luo, Y. (2007, June). A kernel-based reinforcement learning approach to dynamic behavior modeling of intrusion detection. In *International Symposium on Neural Networks* (pp. 455-464). Springer, Berlin, Heidelberg.

Xu, X., Xie, T., Hu, D., & Lu, X. (2005). Kernel least-squares temporal difference learning. *International Journal of Information Technology*, *11*(9), 54-63.

Miller, P., & Inoue, A. (2003, July). Collaborative intrusion detection system. In *22nd International Conference of the North American Fuzzy Information Processing Society, NAFIPS 2003* (pp. 519-524). IEEE.

Dong, G., Han, J., Lakshmanan, L. V., Pei, J., Wang, H., & Yu, P. S. (2003, June). Online mining of changes from data streams: Research problems and preliminary results. In *Proceedings of the 2003 ACM SIGMOD Workshop on Management and Processing of Data Streams* (pp. 739-747).

Cohen, E., & Strauss, M. (2003, June). Maintaining time-decaying stream aggregates. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 223-233).

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004, September). Learning with drift detection. In *Brazilian symposium on artificial intelligence* (pp. 286-295). Springer, Berlin, Heidelberg.

Hansman, S., & Hunt, R. (2005). A taxonomy of network and computer attacks. *Computers & Security*, *24*(1), 31-43.

Etienne, L. (2009). *Malicious traffic detection in local networks with snort* (No. STUDENT).

Bakhdlaghi, Y. (2017). Snort and SSL/TLS inspection. *SANS Institute, InfoSec Reading Room*, 24.

Prasse, P., Machlica, L., Pevný, T., Havelka, J., & Scheffer, T. (2017, May). Malware detection by analysing network traffic with neural networks. In *2017 IEEE Security and Privacy Workshops (SPW)* (pp. 205-210). IEEE.