DOI: http://dx.doi.org/10.26483/ijarcs.v14i1.6946

Volume 14, No. 1, January-February 2023

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

INCORPORATING A HONEYFARM WITH MLFFNN IDS FOR IMPROVING INTRUSION DETECTION

Dr. Loye L. Ray
Department of Doctorate Studies
Colorado Technical University
Colorado Springs, CO

Abstract: Today's networks must deal with dynamically changing threats each day. Use of static datasets to train and prepare multi-layer feed forward neural network intrusion detection systems (MLFFNN IDS) doesn't address these new threats. The use of real traffic data to train neural network IDSs has been out of reach in organizations due to privacy and concerns. Now the use of a honeyfarm system can provide real-time data to a MLFFNN IDS so that it can adjust to new threats as they begin. This system also removes the privacy and concerns since information about the network is false and acts as a decoy to lure attackers away from the real organizational network. This paper introduces ahoneyfarm architecture one can use with a MLFFNN IDS to improve intrusion detection capability.

Keywords: Honeynet, Honeypot, Intrusion Detection, Neural Network

I. INTRODUCTION

Network intrusion detection systems (IDS) today face dynamically changing threats. Intruders use a variety of techniques to include variations of old attacks and creation of new ones. This forces IDSs to find ways of detecting new and unknown attacks. Incorrect detection of these events can lead to security breaches and data theft. The challenge of IDSs is to detect new threats[1]. An anomaly-based multi-layer feed forward neural network (MLFFNN) IDS can learn and detect about new and unknown attacks. To accomplish this requires a data source to train the MLFFNN IDS. The most common one is the KDD 99 dataset. There are others such as NSL-KDD, ISCX2012, UNSW-NB15 and CICIDS2017. Each of these contain static information on attacks and don't stay updated as variations and new attacks are discovered. Thus, the IDS suffers from poor performance in a real network environment.

The solution is to use real traffic data to detect ever changing security threats[6]. The difficulty is how to collect real traffic data and does it contain enough information to be useful. A possible compromise is to use a dataset and real traffic data to obtain adequate samples for training and testing the IDS[9]. Currently there are no publicly available datasets based on real network traffic due to privacy concerns from their owners. There is no standard method for collecting real traffic for training and testing a MLFFNN IDS[7]. Reference [8] found that a dataset composed of simulated and real traffic is needed. A solution is to use honeypots to collect real traffic.

II. LITERATURE REVIEW

A honeypot simulates real systems that is used as a decoy against intruders and hackers. They are virtual machines (VMs) that can be configured to look like real services on a network. Information that is attractive to the attacker is put on the honeypot. This information is used to record their activity and attacks patterns they used [10]. These include email, database, etc. Honeypots are grouped based on their level of contribution [5]. Low level interaction honeypots emulate a service provided by the network. These can be easily built by spinning up a VM and configure it using Honeyd. Hackers can recognize it as a low association honeypot that doesn't allow the attacker to exploit weaknesses and use the system to attack other systems. Thus, they may not take the bait and disclose how they are interrupting the network.

The high-level interaction honeypots allow attackers to see a more genuine network and servers with vulnerabilities. They also provide a realistic administration for gaining control of the simulated system. These systems can catch data about an attacker while recording how they performed the attack. These honeypots are built as a system called a honeynet. They are used in bigger networks where there are more services that can be emulated. Honeynets operate as an intelligence collection system (Honeynets provide security watch over networks and act like a real system. They lie between the real production systems and the honeypots. During an attack, the honeypot and a log sever collects information about the intruder. The Honeynet controls packet flow and has an IDS to gather and process threat intelligence information [10].

For better interruption recognition systems, one can combine honeypots and honeynets into a honeyfarm. A honeyfarm utilizes the strengths of both honeypots and honeynets to better detect and gather threat intelligence on an attacker. Threat intelligence (TI) in the process of searching for and gathering information and data on various attacks that could be in the wild. The hybrid architecture in this paper is modeled into a honey farm of both types of honey pot devices.

Reference [5] devised a honeypot interruption analyzer called Honey Analyzer. It uses Honeyd and TCPDump to collect data and perform analysis. The Honey Analyzer performs data analysis and extraction first. This involved

capturing data using both honeyd and TCPDump logs. TCPDump provides sniffing data for analysis. The next step is to analyze the data to extractprecise attack signatures. To get the attack signature, a graphical interface uses longest common subsequence (LCS) to obtain the length of the longest sequence between different subsequences in the data collected. A weakness of this system is that only checks HTTP, FTP, and SMTP services. There are many different types of ports and protocols used in network traffic. This means that an attacker using different protocols will not be able to be detected with the Honey Analyzer. An effective system needs to be able to detect many aspects of the traffic. The Honey Analyzer also doesn't use a dataset such as KDD to help detect other network traffic. The combination of both Honey Analyzer and a dataset may improve detection capability.

Reference [4] described a network defense model composed of a hybrid firewall module, IDS, and a virtual honeynet. Each are linked up to conduct real-time monitoring and detection [4]. The hybrid firewall segregates the model from the rest of the network and the internet. It also imposes strict controls over any access attempts by an external attacker. This keeps intruders from using the honeynet as an attack tool against other systems by restricting outbound traffic. The IDS provide real-time monitoring and detection of attacker activity. An IDS creates records and sends alerts on questionable activities. The honeynet acts as the glue that connects the components into an early warning system. The system not only provides threat intelligence information but prevents the intruder from conducting interactive communication with external systems. Data loss prevention is another aspect of the model. The hybrid firewall ensures that connections outside the model is preventing malicious activity. Some issues to be investigatedinclude examining encrypted traffic, secure data communications between the components, and more study into early warnings to improve threat intelligence of the model [4].

Reference [11]describes the use of neural networks (NN) as a anomaly-based IDS. Security events are feed into the NN which classifies the information and alerts when a malicious traffic is detected. The NN can classify traffic to different types of bad behavior. It doesn't rely on any preconceptions which avoids need for preestablished features or thresholds[11]. The learning capability of NN IDS can adapt to different behaviors in the network. The concept of a honeytrap was introduced using a honeynet. In a honeytrap, the network is considered a dedicated sacrificial system to attack intruders. Both NN IDS and honeytrap is implement in software. This makes it easily to configure and quickly restored if compromised. Compared to other honeynet models, it utilizes a firewall to prevent a hacker to use a compromised VM to attack other devices.

A hybrid honeypot framework was devised by reference [2] to better protect networks. The framework is an adaptive IDS based on low and high interaction honeypots. The hybrid framework simulates production systems on the network to fool attackers in believing them as the rest thing. Traffic is sent to the low interactive honeypots (honeyd) where they are diverted to the high-interactive honeypots. The high-interactive honeypots are connected as a honeynet that allows the attacker to interact with real-live services. The framework uses free and static IP addresses. The free IP addresses increase the chances for detecting attackers on the network. Like other honeypot IDSs, the hybrid framework uses a honeywall to isolate the network from the production system. The honeywall contains a bridge and Snort IDS for directing traffic. The Snort portion is a signature-based IDS. The honeyds are passive in nature to capture and record intrude activity. An improvement would have them analyze intruder activities determine attack information for the signature database.

A software honeypot-based intrusion detection and prevention system (IDPS) was developed by reference [3]to detect intrusions to a network. It is composed of a honeypot server application, monitor application and IDS application. Compared to other honeypot-based systems, the honeypot-based IDPS uses only low-interaction honeypots (honeyd). There is a data analysis engine to create attack signatures for the IDS application to use. The IDS application uses both misuse and anomaly-based algorithms for detecting attacks. This allows the IDS application to accept new detection signatures from the honeypot server. The anomaly detection side helps detect normal traffic versus abnormal traffic. The monitor application is used to control the honeypot server and IDS applications. It can make changes to the system to include creating signatures, start/stop honeypots, and create and configure honeypots. One drawback of this system is that intruders can easily determine that the honeypots are fake and may avoid them. Adding the use of high-interaction (honeynet) honeypots would better attract intruders to gain improved detection of malicious activities.

III. PROPOSED ARCHITECTURE

Since public datasets like KDD99 are outdated and don't contain current attack signatures, NNMLFFIDSs need a means to keep up to date against the dynamic threats that attacks networks. A proposed architecture includes adding a honeyfarm system to an anomaly based MLFFNN IDS. This bolt on system will be used to collect, analyze, and create signatures for the IDS to use. Doing so will provide the IDS fresh and updated information for detecting new threats and any variates. AMLFFNN IDS is composed of layers of neurons that operate like the neurons in the brain (Figure 1). There are three layers called input layer, hidden layer, and output layer. The data from the network or honeyfarm would be collected and enter the MLFFNN IDS in this layer. From here the information is compared against malicious activity that the IDS recognizes. In the case of data from the honeyfarm, all data will be considered as new malicious data and used to train the IDS. This malicious data will be moved to the output layer and comes out as specific types of activity. Generally, the output if either normal or malicious activity. The malicious activity can be broken down by its type to alert security personnel what actions they need to do to mitigate the activity.

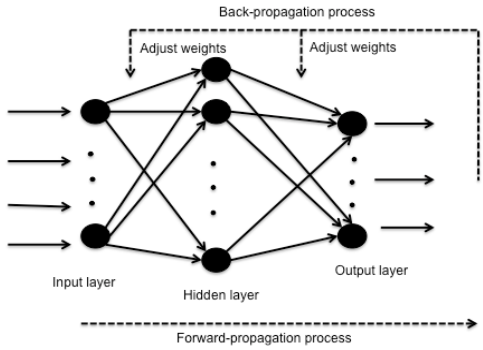

Figure 1. Multi-Layer Feedforward Neural Network Intrusion Detection System

The honeyfarm system is composed of a honeypot subsystem, honeynet subsystem, and honeywall (Figure 2). All three work to entice attackers to take over and use the honeypot decoys for malicious activity. The honeywall acts as a gate to allow the attackers in but filter malicious traffic from getting out. This is to prevent intruders from using the honeypots as a source for conducting attacks on other systems. It acts as a firewall with rules to accomplish this. The honeypot subsystem is a group of VMs built to emulate real system on a network. These include email, application, database, and other services one would find on an organization network. It acts as a low interactive group of honeypots that look like real systems. but the attacker cannot exploit the VMs to attack others. Intruders can see these as fake and will avoid them. To avoid this, the honeypots redirect the intruder to the honeynet.The honeynet is composed of high interactive honeypots that allow intruder to interact with the VMs. Use of high interactive honeypots presents a risk that intruders can use the VMs to exploits other systems. This is prevented by the honeywall.

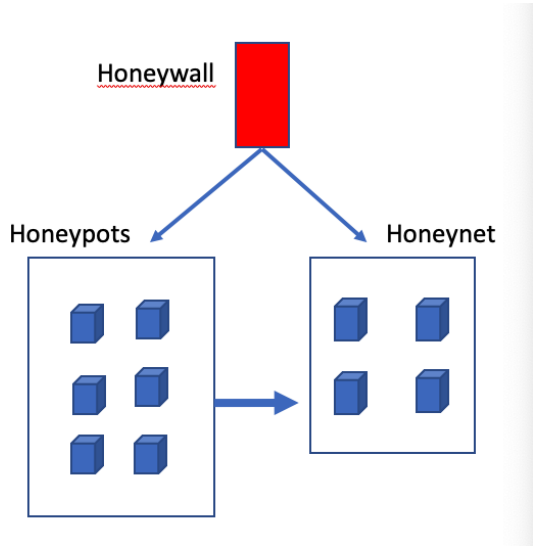

Figure 2. Honeyfarm System

When an intruder tries gaining access to the honeypots and the honeynet, an analyzer collects information to create signatures from the honeypots and stores them in a database. The database feeds the new signatures to the MLFFNN IDS. Putting the MLFFNN IDS in-line with the network traffic will create an IPS that will block malicious traffic. A control station is used to manage the honeypots and honeynet. This includes configuration management of the resources in the hybrid system. Building the hybrid system requires the use of private and free IP addresses that not anyway associated to the productions or other areas of the organization network. This way an intruder cannot sweep the IP range based on an IP address used in the organization network. The honeywall will help prevent the sweeps as well.

IV. CONCLUSION

The use of a honeyfarm can overcome the training and performance of MLFFNN IDSs where public datasets are used. The use of real network traffic can improve detection of malicious traffic. It can also improve the training of the MLFFNN IDS by providing realistic data to train on.
The next step is to construct the honeyfarm system with a MLFFNN IDS and put on an organizational network. Data needs to be collected to see where both the honeypots and MLFFNN needs to be tuned. The honeypots must be configured for what similar systems are used in the organization. Firewall rules in the honeywall will need to be created as part of this tuning effort.

Also, the MLFFNN IDS needs to try various algorithms to tune the training and performance rates to reduce the chances of false positives. This can be done online and reduce downtime of the IDS when trying to train the system. Normally an MLFFNN IDS is training offline with public datasets. Measuring this difference is needed to determine the improvement of the honeyfarm to reduced training rates and increase detection rate.

V. REFERENCES

[1] H.L. Ahmed, N.A. Elfeshawy, S.F. Elzoghdy, H.S. El-sayed, and O.S.Faragallah, “A neural network-based learning algorithm for intrusion detection systems.” Wireless Personal Communications 97, pp.3097-3112, 2017.

[2] H.Artail, H. Safa, M.Sraj,I.Kuwatly, and Z. Al-Masri, “A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks.” Computers & Security, 25, pp.274-288, 2006.

[3] M.Baykara and R. Das, “A novel honeypot-based security approach for real-time intrusion detection and prevention systems.” Journal of Information Security and Applications, 41, pp.103-116, 2018.

[4] Y.C. Cho, and J.Y. Pan, “Hybrid network defense model based on fuzzy evaluation.” The Scientific World Journal, 2014.

[5] P.Kumar, and R.S. Verma, “A review on recent advances & future trends of security in honeypot.” International Journal of Advanced Research in Computer Science, 8(3), pp.1108-1113, March-April 2017.

[6] L. Le Jeune, T.Goedeme, and N.Mentens, “Machine learning for misuse-based network intrusion detection: Overview, unified evaluation, and feature choice comparison framework.” IEEE Access, 9, pp.63995-64015, 2021.

[7]. L. Ray, “Training and testing anomaly-based neural network intrusion detection systems.” International Journal of Information Security Science, 2(2), pp.57-63, 2013.

[8]. L. Ray and H.Felch, “Improving performance and convergence rates in multi-layer feed forward neural network intrusion detection systems: A review of the literature.” International Journal of Strategic Information technology and Applications, 5(3), pp.24-36, 2014.

[9]. L. Ray, "Challenges to multi-layer feed forward neural networks in intrusion detection." Issues in Information Systems, 17(I),pp.89-98, 2016.

[10] K.Rushikesh, "Study on honeypot based secure network system." International Journal of Advanced Research in Computer Science, 10(3), pp.71-72, May-June 2019.

[11]. T. Verwoerd, and R. Hunt, "Intrusion detection techniques and approaches." Computer Communications, 25, pp.1356-1365, 2002.