Neural Network Based Attack Detection Algorithm

ARACELI BARRADAS-ACOSTA, ELEAZAR AGUIRRE ANAYA,
MARIKO NAKANO-MIYATAKE, HECTOR PEREZ-MEANA
ESIME Culhuacan
Intituto Politecnico Nacional
Av. Santa Ana 1000, Mexico City, 04430 D. F.
MEXICO
hmperezm@ipn.mx, hmpm@prodigy.net.mx http://calmecac.esimecu.ipn.mx

Abstract: - The influence of computer technology on the human activities has greatly increased during the last three decades, due to major developments in the VLSI technology. However this widespread use of computer equipments has generated computer a considerable increase of computer crimes. To reduce this problem it is necessary to carried out a network analysis using the computer network traffic. However the increase of network traffic is huge, doing the analysis of traffic data complicated. Thus it is required to develop an effective and automatic algorithm to carry out the traffic network analysis, facilitating I such way the expert forensic work. This paper proposes a network analysis algorithm using recurrent neural network that can analyze computer network attacks facilitating the evidence extraction. Proposed algorithm can reduce time and cost of forensic.

Key-Words: - Computer network, Forensic analysis, Recurrent neural network, Forensic, Attacks

1 Introduction

The computer science has an important impact in fields such as medicine, economy, communications, educational activities, entertainment, etc., which determine the form in which most people conduct their activities. Thus the importance of computer technology in the development process of a country is huge. There are several programs within a computer operating with open network connections that related with business transactions, bank operation, communication, industrial processes, research, security, etc. The correct operation of all of these transactions is highly dependent on a proper development of computer technology, because, although some these programs are developed for valid or legal purposes, other programs are developed by people with criminal motives. As a result of the last mentioned programs have emerged informatics attacks whose goal is to achieve illegal access, compromising the computers security to obtain information regarding the access to the network, identify the access sources, theft of personal identities, disable the online business, generate network traffic, delete or extract business or personal information, etc. However, because not all attempts to access the network can be considered as an attack, is a complicated job to easily discriminate between attacks and not attack, because it has not easily detected symptoms.

As a response to the growing informatics crimes arise the computer forensic science whose goal is to discover, retrieve information about an attack, implementing a review of all attack with several tools and algorithms that can adequately perform this task, generating in such way the evidence to submit it to a personal legal action or to reconstruct attacked actions.

This paper proposes a network analysis method to extract attacks evidence. To this end an automatic and effective algorithm at the forensic analysis stage of the network is proposed, based on recurrent neural networks. Recurrent neural networks (RNN), which intend to emulate the behavior of the human brain, have the ability to learn and extract information from an input sequence to carry out a classification of attacks on the network. Thus with a RNN algorithm it is possible to extract evidence that the network has been compromised or attacked.

2. Network Attack

It is necessary to take into account some concepts before performing network information capture.

2.1. Attack

Due to the great development of computer technology to generate informatics crimes, many people is trying to harm the computer system or network, where the

goal is to attack the computer systems to obtain the desired results such as theft of personal identities, disable online business, generating traffic in a network unexpectedly, delete or extract confidential information, obtain identification of access source, generate viruses or worm without authorization.

An attack is a series of steps from attacker to achieve an unauthorized result. An attack generally is composed of five parts, which form part of a logical algorithm of an attacker.

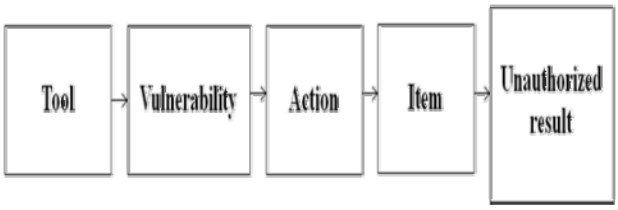

Fig. 1. Attack

In figure 1 shows elements carried out by an attacker. An attacker uses a tool to exploit a vulnerability to perform an action on a target in order to achieve an unauthorized result. To be successful, an attacker must find paths that can be connected (attacks), perhaps simultaneously or repeatedly. [1]

2.1.1 Different types of attacks

There are different types of attacks, which depend on the management of information within computer. One of them is the passive attack in which the attacker listens or monitors the network traffic without altering information. The purpose of such attack may be:

- To obtain information from source and destination, reading headers of packet monitored.
- To control the traffic volume exchanged between entities monitored and data, thus obtaining information about activity or unusual inactivity.
- To control usual hours of data exchange between entities of the communication network allowing the extraction of information about the activity periods.

One disadvantage of passive attacks is the difficulty to detect this attack because it does not generate any change in the data, only with help of encryption of information.

Other type of attacks are the active attacks in which the attacker listens or monitors the network traffic altering information which is subdivided into following categories:

- Interruption. A system object is destroyed, unusable or unavailable.
- Intercept. An element gets unauthorized access to a particular system object.
- Modification. To get access to modify the object.
- Manufacture. An unauthorized element inserted false objects in system.
- Destruction. Modification that disable the object.

3. Forensic Network

The term forensic network was presented by the security expert Marcus Ranum in the early 90's [2] and is taken from the legal and criminology terms. Here the term "Forensic" refers to the crimes investigation. The digital forensics has been defined in 2001, while the network forensic is defined as "the use of scientifically proven techniques to collect, identify, examine, correlate, analyze and document digital evidence from multiple sources and digital transmission in order to discover evidences related to intentions, or measurements of success of unauthorized activities designed to alter, corrupt, and/or system components, providing also information to assist in the recovery or response to these activities [3]. Sources of digital evidence on networks include server logs, contents of networks devices, and traffic on both wired and wireless networks. In forensic network it is necessary to understand how the protocols, configurations and communication infrastructure are combined to result in a specific moment as well as a particular behavior. This combination of words that provides a professional understanding about the operations of computer networks is possible to obtain following the protocols and criminalist training to establish traces of the movements and actions that an intruder has to do to complete its action. When digital investigators do not have access to a key computer, it is necessary to reconstruct the events using only networks evidence.

Computers connected to the global Internet communicate using a set of protocols collectively called TCP/IP (Transport Control Protocol/Internet Protocol). TCP/IP is essentially the common language that enables hosts on these individual, often dissimilar networks to communicate. Each TCP connection bi-directional: one flow for receiving data and a second flow for sending data. A tool can monitor network traffic and maintain logs for later analysis [4].

Unlike the crime in the physical world, a criminal can be in several places of a given network at any time. This distribution of criminal activity and the associated digital evidence does it difficult to isolate a crime scene. Having a solid understanding about networks function, in general, will enable an

investigator to understand many different types of networks and will help determine when and what kind of expert is needed [5].

4. Recurrent Neural Network

Network protocols work as a function of time, therefore, an algorithm that adjust this characteristic is the recurrent neural network.

Recurrent neural networks (RNN) are nonlinear dynamic systems able to detect temporal regularities in the sequences can be processed and applied, therefore, a multitude of processing tasks of such sequences. Neural network with recurrent connections (recurrent network), as its name suggests, describes the recurrence of the information. The signal travels from the input layer to output layer, at the same time, some or all data return from the upper to lower, forming a cycle relay information, not return to input layer. The output signal is sometimes allowed to return to node that produced.

4.1. Recurrent Neural Network Design

In recognition of captures, we used a RNN which identifies temporal relationship between data units of data capture traffic that is responsible for temporarily storing information and somehow manage learning of temporal relations. The RNN has recurrent connections that feed units backward as an additional entry. These recurrent connections can be between any of these network units (input, output or hidden units).

4.1.1. Neural Network Architecture

Neural network consists two layers: input layer and output layer as shown in Fig. 2. The structure of this network differs from simple perceptron network including a recurrence in output layer because it is a kind of supervised classification with goal to separate different classes knowing their belonging [6]. Neurons in output layer of network are full connected to each neurons of input layer and neurons of output layer are connected to themselves. Each connection weight value is changed during training process. Therefore, network output y at cycle t is obtained as follow:

$$y_k(t) = f\left(\sum_{j=1}^{m} v_{jk} y_k(t-1) + \sum_{i=1}^{q} w_{ij} x_i(t) \right) \quad \textbf{(1)}$$

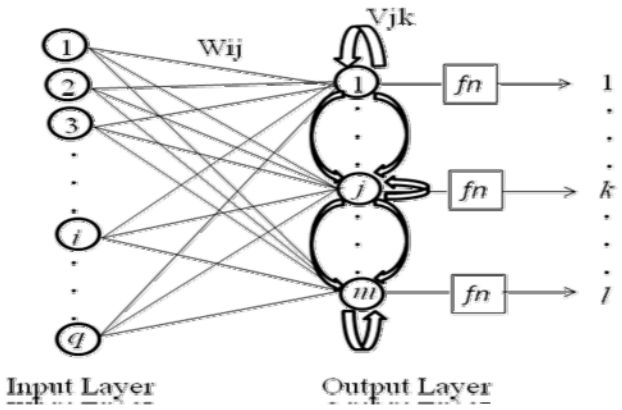

Fig. 2. Architecture of recurrent neural network.

where, X_i is the input value of ith input unit at cycle t, $W_{ij}(t)$ is the weight between ith input unit and jth output unit, V_{jk} is the recurrent value from output units at cycle t-1, q is the number of input units, and m is the number of output units. The activation function is nonlinear. The adaptation of weight uses input value and output value not required for a reference value. The proposed recurrent neural network works as follow:

Initialize weight value W_{ij} and V_{jk}, the weight values W_{ij} initialize with actual values from q input neurons to m output neurons. The weight values V_{jk} initialize with actual values from m output neurons to k output neurons, it is recurrence of same layer that is output. Values initialized with random values ans very small.

1. $Y_k(t-1)$ initialized with values of zeros.
2. Calculate desired output of network applying activation function that is nonlinear.
3. Repite process at cycle t until learning error is zero.

4.2. Learning Algorithm

We use delta rule for training of any neural network, which applies concept of error gradients, which consists mainly of a techical decreased error. The objective is to minimize average square error function between actual output and desired output of network. The error of kth output unit at cycle is denoted by:

$$E = \sum_{k=1}^{N} e_k^2 \quad (2)$$

where $e_k = (d_k - Y_k)$ is the output error, d_k is the desired value of kth output neuron. Y_k is the real output value of the kth output neuron. The output has two possible result,0 if it is a normal flow and 1 if it is an attack. The matrices of weight are adapted as follows:

$$\mathbf{W}_{ij}(t+1) = \mathbf{W}_{ij}(t) - \alpha \frac{\partial E}{\partial \mathbf{W}_{ij}} \quad (3)$$

$$\mathbf{V}_{jk}(t+1) = \mathbf{V}_{jk}(t) - \alpha \frac{\partial E}{\partial \mathbf{V}_{jk}} \quad (4)$$

The variable α is learning constant. Next, consider chain rule which is given by

$$\frac{\partial E}{\partial \mathbf{W}_{ij}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \mathbf{W}_{ij}} \quad / \quad (5)$$

$$\frac{\partial E}{\partial \mathbf{W}_{ij}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \mathbf{W}_{ij}} \quad (6)$$

$$\frac{\partial E}{\partial y_k} = -e_k \quad (7)$$

$$\frac{\partial y_k}{\partial \mathbf{W}_{ij}} = \mathbf{X}_i \quad (8)$$

$$\frac{\partial y_k}{\partial \mathbf{V}_{jk}} = f'\left(\sum_{j=1}^{m} v_{jk} y_k(t-1) + \sum_{i=1}^{q} w_{ij} x_i(t) \right) y_k \quad (9)$$

Defining

$$s_k(t) = \sum_{j=1}^{m} v_{jk} y_k(t-1) + \sum_{i=1}^{q} w_{ij} x_i(t) \quad (10)$$

Therefore, the adaption of RNN weights at cycle *t* becomes:

$$\mathbf{W}_{ij}(t+1) = \mathbf{W}_{ij}(t) + \alpha e_k \mathbf{X}_i \quad (11)$$

$$\mathbf{V}_{jk}(t+1) = \mathbf{V}_{jk}(t) + \alpha e_k f'(s_k) \mathbf{Y}_k \quad (12)$$

5. Forensic System Based in Recurrent Neural Network

In this section, we present the components that support our approach in network forensic analysis. There is no perfect rule that distinguishes attack from network traffic. In this paper, we developed an automated and effective algorithm at the stage of the forensic analysis of the network that can produce similar result like forensic expert's work with the implementation of recurrent neural networks.

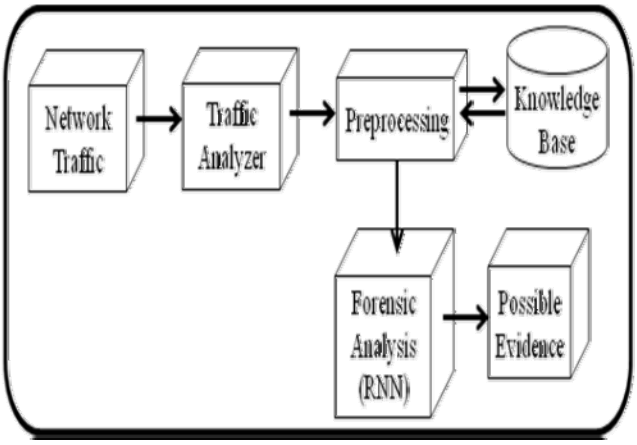

Fig 3. Forensic System Architecture

Figure 3 shows the architecture of our network forensic analysis system. It consists of five components: the traffic analyzer, preprocessing, knowledge base, forensic analysis and possible evidence[8]: a) The Traffic Analyzer reads the network traffic from a sniffer tool (in raw flows format) from networks and hosts under investigation. b) The preprocessing takes the information captured from log file and access the knowledge base that each parameter of a packet as IP source, IP destination, MAC source, MAC destination and protocol changes for an identification as shows in figure 4. Obtaining a data matrix reduced in roder to facilitate the learning of RNN therefore the analysis is fast. c) The Forensic Analysis (RNN) reads the data matrices for training RNN to obtain possible evidence. If the system determines the presence of an attack, it is important that the information available allows the reconstruction the actions affecting the system to be presented to a judge.

5.1 Traffic analyzer

The first step of forensic system is the process of traffic capture where generates log files from the network. They provide the base information to other components of the forensic system. Network forensic system has to capture all information of network traffic and considered all parameter of packets themselves will save in log file in format tcpdump[9].

The traffic analysis selects only important information and the criteria to classify packets are stamp of time, IP source, IP destination, MAC source, MAC destination, Flags (IP), Total frame size,

Protocol, Fragment Offset, Flags (TCP), Options (TCP), Port source and Port destination. This system is considered for the protocols ARP, TCP, UDP and IP and it considers all parameter of these protocols.

5.2 Knowledge Base

Knowledge base stores all IDs network in IP and MAC, and the type of protocol in database with its own identifier to minimize the matrix of a capture.

A matrix without to be minimized contain approximate to 160 digits and a reduced matrix contains 17 digits to training a RNN for only packet and also it presents was trained and the evidence is identified therefore we imagine that a capture contains 30 packets and a traffic capture contain a lot of captures as result will be too many values to training a RNN the learning will be very slow for this motive is necessary to reduce a data matrix to reduce time in stage of forensic analysis and training of RNN.

ID	IP
1	172.16.0.1
2	172.16.0.2
3	172.16.0.3
4	172.16.0.4
5	172.16.0.5
6	172.16.0.6
7	172.16.0.11
8	172.20.8.1
9	172.20.8.2
10	172.20.8.3

(a) IP Identifier

ID	MAC
1	00:1b:77:b0:48:dc
2	00:0f:b0:6a:f9:c7
3	00:0c:29:ab:0e:ef
4	00:0c:29:d9:23:a8
5	00:0c:29:4d:0b:ab
6	00:50:56:c0:00:01
7	00:0c:29:ea:35:3c
8	00:00:00:00:00:01
9	00:00:00:00:00:02
10	00:00:00:00:00:03

(b) MAC Identifier

ID	Protocol
1	ARP
2	TCP
3	UDP
4	IP

(c) Protocol Identifier

Fig. 4 Types of identifiers.

Por example MAC address as AA:BB:CC:DD:EE:FF is a lot of information for one parameter and is not important the detail of each bit because the system takes all content of this parameter to relate with other parameters.

5.3. Preprocessing

Taking into account all traffic information captured to train a neural network is considered as data matrix with its respective values that corresponds only its important parameters but this matrix is a very extensive and complicated matrix to work with RNN, which was necessary to realize tables of extraction capture where its own identifier to reduce data matrix has.

172.16.0.1	172.16.0.2	00:50:56:c0:00:01	ffffffffffff	60	0.0034	arp
172.16.0.2	172.16.0.1	00:0c:29:ea:35:3c	00:50:56:c0:00:01	60	0	arp
172.16.0.2	172.16.0.1	00:0c:29:ab:0e:ef	00:50:56:c0:00:01	42	0.0003	arp
172.16.0.1	172.16.0.2	00:0c:29:ab:0e:ef	00:0c:29:ea:35:3c	42	0.0001	arp
172.16.0.2	172.16.0.1	00:0c:29:ab:0e:ef	00:50:56:c0:00:01	42	1.0042	arp
172.16.0.1	172.16.0.2	00:0c:29:ab:0e:ef	00:0c:29:ea:35:3c	42	0.0004	arp

Fig. 5 Data capture during n attack

The real values of information capture from network traffic access to identifiers table to obtain its own corresponding identifier. Thus, obtaining a reduced data matrix taking into account all possible features, if a captured is not a parameter, the values are zero (see figure 6).

1	2	6	71	60	1	1	0	0	0	0	0	0	0	0	0
2	1	7	6	60	0	1	0	0	0	0	0	0	0	0	0
2	1	3	6	42	1	1	0	0	0	0	0	0	0	0	0
1	2	3	7	42	1	1	0	0	0	0	0	0	0	0	0
2	1	3	6	42	2	1	0	0	0	0	0	0	0	0	0
1	2	3	7	42	1	1	0	0	0	0	0	0	0	0	0

Fig. 6 Reduced data matrix

In the case of ARP protocol does not have all parameters as shows in the figure 6, the criteria to classify packets previously explained, which does not contain data as Flags (IP), Fragment Offset, Flags (TCP), Options (TCP), Source Port and Destination Port. ARP protocol will be filled with zeros. As evidence of the performance of proposed recognition system it was required to identify several attacks. Evaluation results, given latter, show the efficiency of proposed system when required to carry out attack identification tasks.

5.4 Forensic Analysis

The ANN are trained before being used to identify or recognize patterns typical, therefore within the stage of training will use attacked captures and normal flow captures.

For RNN training, different data sets were captured, considering different environments, to include as different possible as situations, as shown in table 1.

Operating System	Tools	Differents IP's networks
Windows XP Professional	Cain & Abel	
Windows Vista Home Premium	Ettercap	
Windows Server 2003	Nemesis	
Back Track 3		
Red Hat Enterprise Linux		

Table 1 Different environments used for evaluation

The captures were realized with 3 attack tools, 5 attacker's operating systems and different IP's networks, then different attack scenarios are generated such as: Attack scenario 1 which was created with ettercap and Windows Xp Professional like attacker's operating system. Next, the attack scenario 2 was created with ettercap and Back Track 3 like attacker's operating system. Subsequently the attack scenario 3 was created with ettercap and Windows Vista Home Premium like attacker's operating system. The attack scenario 4 was created with Cain & Abel and Windows Vista Home Premium like attacker's operating system.

Other attacks analyzed were the scenario 5, which was created using the Cain & Abel and Windows Xp Professional like attacker's operating system. Other attack use is the scenario 6, which was created with the Cain & Abel and the Windows Server 2003 like attacker's operating system. Finally the attack scenario 7 was created with the nemesis and Back Track 3 like attacker's operating system.

The attack captures were created for different attack scenarios. The importance of different settings of attack is to observe the behavior of the RNN to train different values which belong to a given attack, in order to increase the ability of recognition of this algorithm [10].

The design of our RNN system adapts to these features of the Neural Network:
17 neurons in input layer because data matrix contains 17 parameters (stamp of time, Source IP, Destination IP, Source MAC, Destination MAC, Total frame size, Protocol, Fragment Offset, Flags (TCP), Options (TCP), Source Port and Destination Port). One neuron in out has only one out as an attack or not. In this section, we present design of RNN for recognition of attacks. Figure 7 shows a RNN for these types of protocols in a traffic capture.

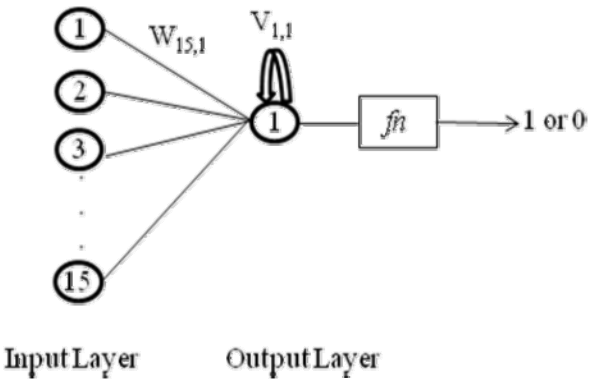

Fig. 7 Forensic Analysis System

5.4.1 Training of the RNN

The implementation of the RNN was in MATLAB R2008a version 7.6.0.324, the training was performed in a processor Intel Core2 Duo CPU 1.50 Ghz, a system type of 32-bit operating system (Windows Vista Home Premium) and a memory (RAM) of 2 GB. 40 attacks captures and 40 captures of normal traffic were used for the training of the RNN structure and they constitute different conditions that were mentioned at the beginning. Already has data matrix which is suitable for training RNN [11].

5.5. Possible Evidence

One of the process should take into account in investigation is obtaining of evidence. The evidence information can be process in analysis stage. The proposed of analyzing the evidences is reduce time and to know as precisely what did it happen in stage of analysis and training of RNN [12]. At output of system, it obtains only result as attack or not, it shows in figure 8.

```
Output is:  1   The file is an attack
Output is:  0   There are not any attack in file
```

Fig. 8 Output of RNN

6. Experiments

Taking into account all related to an attack as its definition, types and categories, first we perform ARP poisoning attack because this attack is a widely used and known to identify an evidence in a RNN algorithm, it recognizes an attack or a normal flow. Session hijacking was performed after of ARP attack with the same features in the development like the capture extraction of sniffer and to reduce data matrix to training of RNN.

Then, we present other attack named as Session Hijacking and in future work we going to present a DoS attack [13] because they have been used in other investigations [14-16].

6.1 ARP Poisoning

The captures of information was generated by different tools to perform ARP poisoning attack as Ettercap, Cain & Abel and Nemesis, all this to obtain several captures in different conditions.

The objective of the ARP poisoning as its name suggests, a communication poison that originate in communication protocol of ARP packets. ARP is responsible for converting high-level addresses (IP) to physical addresses (MAC). Whenever a host sends some information to another IP host must know destination MAC to transmit, then it is necessary to send an ARP request to the network and, the destination host responds with another ARP packet with its physical address. Therefore it is possible to fool a host within network, telling it a MAC address that a third host wants to communicate, producing a redirection of traffic from source host to destination host, all the information comes of source host passing by a third host, and it redirects the traffic to real destination host to operate transparently in the communication of both victims host and the attacker can observe all information of traffic. This type of attack is known as "man in the middle".

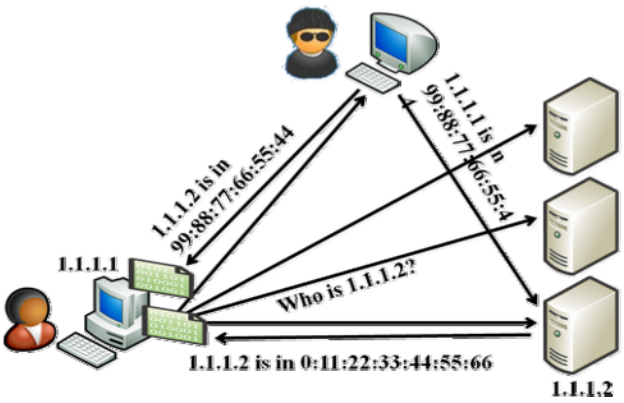

Fig. 9 ARP Poisoning "Man in the middle"

When a third host is between two hosts that are communicating, it monitors all traffic from source host and destination host. Therefore, this allows that all traffic passes through the network, which allows to examine network data and perform capture files. We can analyze information captured with details and sumaries for each package. The classification of packets for ARP protocols are stamp of time, IP source, IP destination, MAC source, MAC destination, Total frame size and Protocol. Next the training is conducted with several tests. The result of RNN it will show two options as in figure 10.

```
What file do you like to test in recurrent neural network?
 1
Output is:  1   The file is an attack
What file do you like to test in recurrent neural network?
 4
Output is:  0   There are not any attack in file
```

Fig. 10 Detection of attack

In this way we can analyze the content of information of the captures.

6.2 Current Work

Actually, we are working about other attack which in it session is hijacking. This attack TCP hijacking was and is one of the most popular techniques for intruders to gain unauthorized access to Internet servers. Even now that almost every systems administrator knows about this potential vulnerability, TCP hijacking is still common because many systems administrators don't understand the principles behind this method. They just can't stop wily hackers who know their stuff. To build an attack to carry out an experiment with Recurrent Neural Network, we explain all of the details about TCP hijacking. It will begin with a brief review of the TCP protocol.

TCP is a connection-oriented protocol, between two extremes and it offers a transmission reliable, designed to fit into a layered hierarchy of protocols that support multiple applications on networks.

The main purpose of TCP is to provide a service of connection or reliable logical circuit and secure peer processes. To provide this service over an unreliable Internet environment, the communication system requires mechanisms related to the following areas: a) Basic Data Transfer, b) Reliability c) Flow Control, d) Multiplexing, e) Connections f) Priority and security

6.2.1 Reliable Communication

A stream of data, sent over a TCP connection is delivered in a reliably and orderly manner at the destination.

The transmission is reliable by using a sequence numbers and acknowledgments. Basically, it is assigned a sequence number to each byte of data. The

sequence number of the first byte of data in a segment is transmitted with that segment and is called the sequence number of the segment. Segments also carry an acknowledgment number, which is the sequence number of the next byte of data expected in the transmission in reverse.

When the module transmits a TCP segment containing data, it puts a copy in a retransmission queue and starts a timer, if the receipt is reached by these data the segment is deleted from the queue. If you do not receive acknowledgment of receipt within a period of expiration, the segment is transmitted. The arrival of the acknowledgment does not guarantee that data has already been delivered to the end user, but only that the receiving TCP has taken the responsibility to do so.

To control the flow of data between TCP modules, uses a flow control mechanism. The TCP receiver returns a window to the TCP sender. This window specifies the number of bytes, counting from the number of the acknowledgment, the TCP receiver is then ready to receive. For TCP packet identification, the TCP header consists of two 32-bit fields that are used as packet counters. They are named Sequence Number and Acknowledgment Number. Another field, named Control Bit, is 6 bits long, and it carries the following command bits:
URG: Urgent pointer field significant
ACK: Acknowledgment field significant
PSH: Push function
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: No more data from sender

6.2.2 TCP Connection

Let is suppose that host A wants to set up to TCP connection with host B. In this case, host A sends host B a packet like this:

A -> B: SYN, SEQ *x*

In the message that host A sent, the command bit SYN is set, and Sequence Number has an initial sequence number *x* that's 32 bits long. After receiving this packet, host B generates a reply:

B -> A: SYN, SEQ *y*, ACK (*x*+1)

This reply sets the SYN and ACK command bits. Host B sets the Sequence Number to its initial sequence number *y*, and the Acknowledgment Number is set to *x* (from host A) and increased by 1. To finish the establishment phase connection, host A sends:

A -> B: ACK (*y*+1)

This packet sets ACK. The Sequence Number *x* is increased by 1, and the Acknowledgment Number *y* is increased by 1. After sending this packet to host B, host A ends the third-level handshake. A TCP connection between these hosts (A and B) is set up. From now on, host A can send data packets to host B through this new virtual TCP link:

A -> B: ACK *x*+1, ACK (*y*+1); DATA

You can follow the entire TCP connection setup scheme as shows in figure 11.

The TCP connection setup described above, the only identifiers of TCP clients and TCP connections are Sequence Number and Acknowledgment Number two 32-bit fields. This protocol does not place any restriction on the fact reused again and again the same connection. A connection is defined by the pair of connectors, a new instance of a connection will be referred to as an incarnation of the connection. To generate a fake TCP packet, the things that an intruder must know are the current identifiers for definite connection: *x* and *y* and of source and destination ports. All an intruder has to do is get the current values of the identifiers of a TCP packet for a defined TCP connection (like a Telnet connection). Then, to have the intruder's packet recognized as valid, the intruder can send a packet from any Internet-connected host, as long as the packet is masked as a packet from any host that's part of the connection.

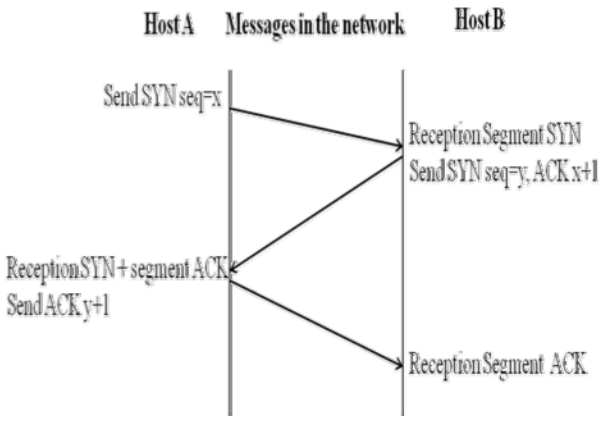

Fig. 11 TCP connection

This sample TCP connection setup scheme details the connection between host A and host B.

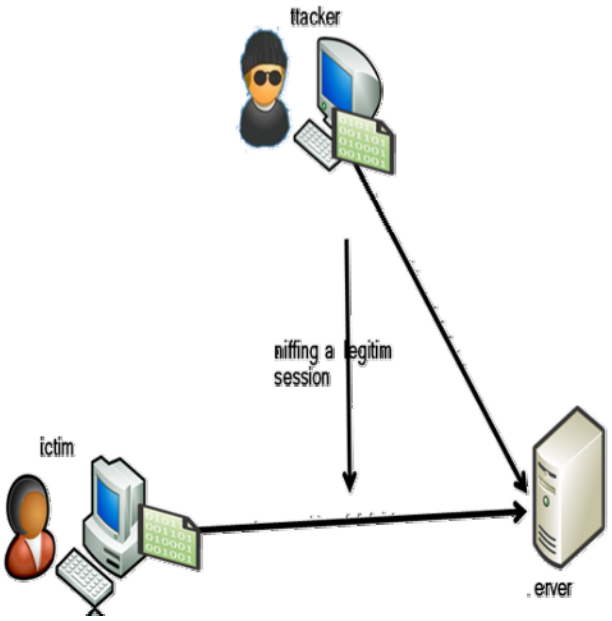

Fig. 12 Session Hijacking.

For an attack like the one depicted above to occur, the only pieces of information that an intruder is required to know are the two current 32-bit parameters (*x* and *y*) that identify the TCP connection When an attacker is in the same segment as the target or when the target's traffic comes through the attacker's segment, the task of getting *x* and *y* numbers becomes trivial. All the hacker must do is grab and analyze network traffic. The TCP protocol allows us to have secure connections only when the attacking side can't grab (sniff) network traffic.

As in the earlier work, now take into account all the parameters of a session hijacking attack, which involves the TCP protocol with 17 values that are: stamp of time, source IP, destination IP, source MAC, destination MAC, Flags (IP), Total frame size, Protocol, Fragment Offset, Flags (TCP) of TCP Options (Maximum segment size, window scale, and SACK permitted), source port and destination Port. In figure 13 shows an array with all the values in this protocol.

Like the previous work of the ARP attacks their real data matrices were reduced to facilitate training and learning of RNN as shows in figure 14.

The next step is to collect multiple captures attacked (session hijacking) and other captures of normal flow related to the TCP protocol to generate the matrices for testing the recurrent neural network, there will observe the efficiency of the recognition algorithm [17].

192.168.10.2	192.168.10.6	00:0c:29:1e:9b:a9	00:0c:29:8a:4d:6a	62	5.4999	tcp	4	1	1043	21	0	2	2	1460	0	1
192.168.10.6	192.168.10.2	00:0c:29:8a:4d:6a	00:0c:29:1e:9b:a9	62	0.0007	tcp	1	1	21	1043	0	12	2	1460	0	1
192.168.10.2	192.168.10.6	00:0c:29:1e:9b:a9	00:0c:29:8a:4d:6a	54	0.0047	tcp	4	1	1043	21	0	10	3	1460	0	1
192.168.10.6	192.168.10.2	00:0c:29:8a:4d:6a	00:0c:29:1e:9b:a9	96	0.0033	tcp	4	1	21	1043	0	18	4	1460	0	1
192.168.10.2	192.168.10.6	00:0c:29:1e:9b:a9	00:0c:29:8a:4d:6a	54	0.1167	tcp	4	1	1043	21	0	10	3	1460	0	1
192.168.10.2	192.168.10.6	00:0c:29:1e:9b:a9	00:0c:29:8a:4d:6a	66	4.245	tcp	4	1	1043	21	0	18	4	1460	0	1
192.168.10.6	192.168.10.2	00:0c:29:8a:4d:6a	00:0c:29:1e:9b:a9	87	0.0032	tcp	4	1	21	1043	0	18	4	1460	0	1
192.168.10.2	192.168.10.6	00:0c:29:1e:9b:a9	00:0c:29:8a:4d:6a	54	0.0002	tcp	4	1	1043	21	0	10	3	1460	0	1
192.168.10.2	192.168.10.6	00:0c:29:1e:9b:a9	00:0c:29:8a:4d:6a	68	3.1769	tcp	4	1	1043	21	0	18	4	1460	0	1
192.168.10.6	192.168.10.2	00:0c:29:8a:4d:6a	00:0c:29:1e:9b:a9	74	0.0034	tcp	4	1	21	1043	0	18	4	1460	0	1

Fig. 13 An TCP attack capture with real data.

36	40	78	76	62	6	2	4	1	1043	21	0	2	2	1460	0	1
40	36	76	78	62	1	2	1	1	21	1043	0	12	2	1460	0	1
36	40	78	76	54	1	2	4	1	1043	21	0	10	3	1460	0	1
40	36	76	78	96	1	2	4	1	21	1043	0	18	4	1460	0	1
36	40	78	76	54	1	2	4	1	1043	21	0	10	3	1460	0	1
36	40	78	76	66	5	2	4	1	1043	21	0	18	4	1460	0	1
40	36	76	78	87	1	2	4	1	21	1043	0	18	4	1460	0	1
36	40	78	76	54	1	2	4	1	1043	21	0	10	3	1460	0	1
36	40	78	76	68	4	2	4	1	1043	21	0	18	4	1460	0	1
40	36	76	78	74	1	2	4	1	21	1043	0	18	4	1460	0	1
36	40	78	76	54	1	2	4	1	1043	21	0	10	3	1460	0	1
36	40	78	76	78	5	2	4	1	1043	21	0	18	4	1460	0	1
40	36	76	78	84	1	2	4	1	21	1043	0	18	4	1460	0	1

Fig. 14 Reduced data matrix for a TCP attack

7. Results

In development of neural network takes into account the minimized data matrices to train, they generate iterations to calculate the learning of error, the error is zero for the reason that learning constant is less than 1 [18]. An example is training of 42 information captures which included 21 captures with attacks and 21 captures of normal flow. The total number of captures in these instances are different combinations of operating systems, attack tools, IP's networks, which they create different environments. At the iteration 5, the error is zero, thus the training was learning very fast for these 42 captures. When it is an attack, output is equal to 1 and when it is a normal flow value is equal to 0.

The order of captures to train was ordered in this way can be random or in sequence, explanation of this order is to immediately identify attack; is first attack capture after normal flow capture that successively up

to ending with 42 captures to train. In figure 16 shows trained outputs of the network, which they registered in points.

Number of captures	Average of attacked captures	Tools	Attacker's Operating System
12	6.66	Cain & Abel	Windows XP Professional & Windows Server 2003
1	5.45	Cain & Abel	Windows Server 2003
5	4.001	Cain & Abel	Windows Vista Home Premium
1	3.65	Cain & Abel	Windows Vista Home Premium
3	2.78	Ettercap	Windows XP Professional
1	2.74	Ettercap	Back Track
2	2.5	Ettercap	Windows XP Professional & Back Track
1	2.22	Ettercap	Back Track
1	1.09	Cain & Abel	Windows Vista Home Premium
1	0.34	Ettercap	Back Track
1	0.29	Ettercap	Back Track
3	0.28	Ettercap	Windows XP Professional & Back Track
2	0.26	Ettercap	Windows XP Professional & Back Track
3	0.25	Ettercap	Windows XP Professional & Back Track
2	0.24	Ettercap	Windows XP Professional
1	0.14	Ettercap	Back Track
5	0.011	Nemesis	Back Track
2	0.01	Nemesis	Back Track

Table 2. 11 Average of time since previous displayed packet to next packet of a capture

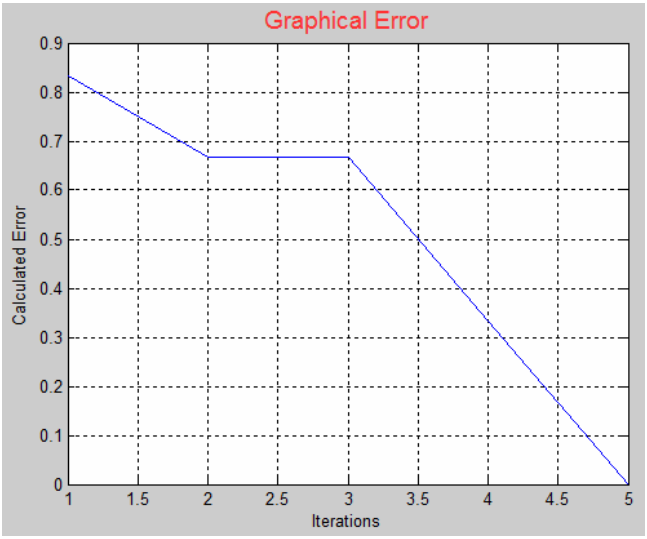

Fig. 15 Graphical error.

In each capture has its existence of attack, for example in the number 1 of capture indicates that in the existence of attack is one, it contains an attack. In the number 4 of capture is zero, it does not contain an attack, it is normal flow capture. In table 2 shows the comparison of the time it takes to create a package to another within a capture.

All averages of time since previous displayed packet to next packet that contain a capture, they were taken of each capture (captures with attacks and captures of normal flow). It shows time generated by each packet into attack capture, time generated in the attack captures are similar with other attack captures for example there are 12 captures that they generated in the same time with cain & abel tool and Windows Xp Professional & Windows Server 2003. Time generated in normal flow captures are too differents and they do not have similar time between them.

The ARP poisoning attack creates its own packages quickly and that IP's of each host have the same MAC address of attacker which no realize any search of MAC addresses in an ARP cache, however, in normal traffic are updated looking for the addresses and MAC addresses of each IP.

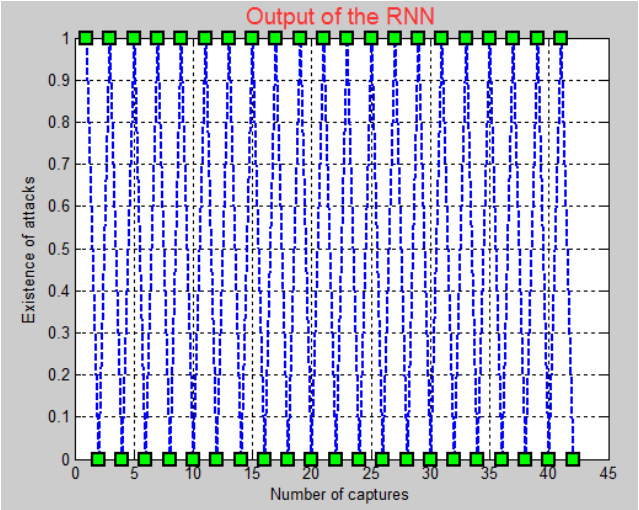

Fig. 16 Output of the Network

8. Conclusions

Network forensic is based on methodologies that will help to forensic expert to resolve the procedure followed by an attacker as performed action. Within network forensic science, there is a part, which requires more time of the forensic expert that is extraction of evidences. But forensic science is a recent creation in networks and is now at the stage of development.

In this paper, we present an algorithm of recurrent neural networks for the implementation of evidence classification for network forensics. This analysis reduces time and cost of forensic expert's work in an affective manner. We did a series of experiments for the identification of evidence, using a groups of attacks capture and normal flows capture for training and testing. The results indicate that a network trained with RNN algorithm improves the system performance analysis in which use network preceptron simple.

The time of training delay between 44.87 seconds to 57.87 seconds for 42 captures. In the future works we are going to realize analyses with other algorithms and other types of attacks to find the efficiency and automatic alghoritms for evidences extraction.

References:

[1]Howard John and Longstaff Thomas. A Common Language for Computer Security Incidents. Sandia National Laboratories.USA, 1998.

[2]Marcus Ranum, Network Flight Recorder, http://www.ranum.com/

[3]Digital Forensic Research Workshop. A Road Map for Digital Forensic Research, 2001.

[4] Casey Eoghan. Digital Evidence and Computer Crime.2nd ed.Oxford:2004.

[5] Peisert Sean and Bishop Matt. Computer Forensics In Forensis. University of California, San Diego: 2008.

[6]Schalkoff Robert. Artificial Neural Network. McGraw Hill,1997.

[7] Lee Seong-Whan and Kim Young-Joon. A New Type of Recurrent Neural Network for Handwritten Character Recognition.Seoul, korea: 1995.

[8]Ajith Abraham. Neuro Fuzzy Systems: State-of-the-art Modeling Techiques.Australia.

[9]Wang Wei and Daniels, Thomas. Diffusion and Graph Spectral Methods for Network Forensic Analysis.Germany: September, 2006.

[10]Kang Seung-hoon and Kim, Juho. Network Forensic Analysis Using Visualization Effect. Sogang University, Korea: 2008.

[11]Rong-Chang Chen and Lai Chih-Yi. Back Propagation Networks for Predicting Credit Card Fraud with Stratified Personalized Data. WSEAS Transactions on Computers.Isue 3, Volume 6, March 2007.

[12]Kim Jung-Sun; Kim Doug-Geun and Noh Bong-Nam. A Fuzzy Logic Based Expert System as a Network Forensic. Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference,Volume 2, Issue , 25-29 July 2004 Page(s): 879 – 884.

[13]Stefan Axelsson. The Base-Rate Fallacy and the Difficulty of Intrusion Detection. ACM Transactions on Information and System Security, Vol. 3, No. 3, August 2000, Pages 186–205.

[14]Wagener Gérard; Dulaunoy; Alexandre and Engel Thomas. Towards an Estimation of the Accuracy of TCP Reassembly in Network Forensics. University of Luxembourg:2008.

[15]Pawalai Kraipeerapun, Chun Che Fung and Kok Wai Wong. Uncertainty Assessment using Neural Networks and Interval Neutrosophic Sets for Multiclass Classification Problems. WSEAS Transactions on Computers. Isue 3, Volume 6, March 2007.

[16]Genge Bela and Losif Ignat. An Abstract Model for Security Protocol Analysis. WSEAS Transactions on Computers. Isue 2, Volume 6, February 2007.

[17] Shahbaz Pervez, Iftikhar Ahmad, Adeel Akram and Sami Ullah Swatti. A Comparative Analysis of Artificial Neural Network Technologies in Intrusion Detection Systems. WSEAS Transactions on Computers. Isue 1, Volume 6, January 2007.

[18]Yang Xiang; Wanlei Zhou. Mark-aided distributed filtering by using neural network for DdoS. Defense Global Telecommunications Conference, 2005. GLOBECOM apos;05. IEEE Volume 3, Issue , 28 Nov.-2 Dec. 2005