Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

Botnet detection based on Markov chain and Fuzzy rough set

Aziz Ezzatneshan [1], Seyed Reza Kamel Tabbakh Farizani[2*], Maryam Kheirabadi[3], Reza Ghaemi[4]

1 Department of Computer Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran(ezzatneshan@gmail.com)

2 Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran(drkamel@mshdiau.ac.ir)

3Department of Computer Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran(m.kheirabadi@iau-neyshabur.ac.ir)

[4]Department of Computer Engineering, Quchan Branch, Islamic Azad University, Quchan, Iran (r.ghaemi@iauq.ac.ir)

Abstract

Botnets now make up a wide range of cyber-attacks, which are a network of infected computers connected to the Internet, with remote control. So far, a lot of research has been done in this field, the proposed methods are based on the signatures of discovered botnets, anomalies, traffic behavior, and addresses. Each method has both advantages and disadvantages. This research proposes a structure for performing identification operations, which is presented in this research based on the Markov chain and is based on behavioral analysis. A disadvantage of the past methods is the inability to receive network information at a very high speed. In this research, it has tried using a solution to receive traffic at a very high speed of about 40 Gbps and analyze it. To be able to perform the analysis with a lower overhead. The proposed method can investigate the behavior of botnets by examining the area of behavior better than the previous solutions, and as a result, during the solutions used by botnets to hide their behavior, it can counter and identify suspicious flows. The accuracy of the proposed method was found to be 96.170%.

Keywords: Markov chain, botnet detection, network flow, feature extraction, fuzzy rough, Analytic hierarchy process

Introduction

Today, the use of botnets as a tool for large-scale criminal activities in computer networks against large targets such as a country has greatly increased. The bot is a distributed environment that is used for various attacks with a large volume, for this reason, the detection of this type of attack is considered one of the important issues in the security of computer networks.

Attackers, after capturing the victims' vulnerable systems, install their malicious software that can be controlled remotely on those systems, and then they can use these victims to perform various

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

Internet attacks, such as sending spam and preventing attacks. From the distributed service, identity theft and other criminal activities can be carried out on a massive scale, while the real identity of the attacker remains hidden[5].

The main problem with botnets is that they perform these actions secretly, that is, until we specifically look for them, we will be unaware of their presence in the system, and as long as they remain in the victim's system, the victim's system can resist. In contrast, failure to execute the instructions of the owner of the botnet will not be the case [6].

Considering that the past methods each have problems either problems in terms of processing overhead or in terms of accuracy in detecting attacks, therefore, in this research, a solution has been attempted to deal with and detect botnet attacks. To be determined. In this proposed solution, the Markov chain is used and it has been done in such a way that in addition to increasing the accuracy of the detection, attacks can be detected with relatively lower processing overhead and the information about the victims can be prevented. Alternatively, identifying bot attacks is a very difficult task because bots try not to show their original behavior easily and try to act in such a way that their behavior is mixed with the behavior of nonsuspicious streams so that the original behavior is hidden and Therefore, in this research, an attempt has been made to use a part of the flow in which the part has shown its original behavior with very high probability, and in this way, attacks can be identified with high accuracy.

In the following, the basic concepts are stated and necessary terms are explained. In section 4; a review of previous works is made. In section 5, the evaluation of the proposed method is described. In the end, a general summary of the work is given and future suggestions are made.

BASIC CONCEPTS

Malware

Malware is referred to by various names such as malware and malCode. There are countless definitions of malware. Christodorescu, for example, describes malware as a program that has a sinister purpose, while Maurice identifies any code that is added, modified, or removed to a software system with the intent of causing a malfunction [1][Miller 17].

There are several types of malware such as viruses, botnets, worms, and trojans. According to the Internet Security Company [3], published in April 2018, more than 317 million new malware has been created compared to previous years, which indicates that it has grown by 26% more than in 2017. Of this ratio, more than 28% are aware of the virtual machine environment and in the virtual machine, the environment either stops altogether or shows other behavior [21].

Botnet

Each botnet contains a group of machines infected with the same malicious code driven by an attacker through a command and control channel. A bot is malware that performs its malicious behavior by receiving commands sent by a central element [7]. Malware is software that is

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

motivated to infiltrate, modify, or destroy other software secretly. A set of bots that are coordinated and run by a single command (bot-master) is called a bot network. What can distinguish a bot as malware from other types of malware is the coordination of the bots and their command-ability of the same element. Other threats that botnets can create include data theft, tampering with social polls, and the use of bot-infected computer resources [29]. For example, recently, bot resources have been used to process and extract bitcoins, send spam, steal confidential information, as well as infect other computers and spread on the network [8]. The spread of botnet networks and the presence of a large number of infected computers is the main reason why networks are dangerous. As a result, an attacker could take advantage of the bandwidth, storage, and processing power of a large number of computers and pose an even greater threat so the need to create a system to detect bot-infected computers has become increasingly important today. There are three tasks in a bot network: Zombie, Relay, and C&C Server [23].

Botnet detection criteria

Botnet detection has different criteria. If a method can detect botnets in the early stages of their life cycle (formation, command, and control stages), it can disable them before engaging in a cyber-attack [9]. In contrast, methods that distinguish botnets from their life cycle during the attack phase are more accurate [28].

Botnet Detection levels

Botnet detection methods can use two different levels of correlation analysis including individual level and group level [10]. In individual-level analysis, the focus of botnet detection methods is on identifying each bot-infected host individually on the network, without considering the behavior of infected hosts on another bot [11]. The individual-level analysis is usually performed by adapting the observed activities to the known patterns in the database. Therefore, prior knowledge of botnets is required [27].

Botnet Categories

Botnets can be distinguished based on the type of structure they implement, as well as the type of technology they use and the communication channels they use to connect [22]:

- Centralized bots: Some networks are based on one or two C&Cs, and each bot is in direct contact with the command and control server. Sorting and managing these types of architectures is simple but very vulnerable, and when C&C shuts down, the entire botnet network crashes.
- Decentralized or peer-to-peer botnets: They are structured in such a way that they are not dependent on one or more C&C servers. In this type of architecture, recognizing one bot cannot disable the entire network. In decentralized botnets, also called peer-to-peer (P2P) botnets, bots are not necessarily connected to C&C servers, but are sent from one zombie to another by creating a mesh structure [26].

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

- Hybrid botnets: Decentralized architecture management is very complex. For this reason, hackers prefer to use hybrid structures. Therefore, they use both centralized and decentralized structures to make themselves more resistant to discovery.

Markov Chain

The Markov chain is a random model for describing a sequence of possible events in which the probability of each event depends only on the state of the previous event. The Markov chain, named in honor of Russian mathematician Andrei Markov, is a mathematical system in which the transition between states takes place from one state to another. The Markov chain is a random process without memory, which means that the conditional probability distribution of the next state depends only on the current state and is independent of its past [32]. This type of memorylessness is called the Markov property. The Markov chain is a time-discrete random process with the Markov property. Markov's property states that the conditional probability distribution for the system in the next step depends only on the current state of the system and does not depend on the previous states. Because the system changes randomly, it is generally impossible to predict the state of the Markov chain at certain points in the future. However, the statistical features of the system are predictable in the future, which is very important in many applications [32].
Changes are called transmission system states, and the probabilities attributed to these state changes are called transfer probabilities. A Markov process is characterized by a numeric state space, a transition matrix to describe the probabilities of each transfer, and an initial state (or initial distribution) in the state space [12].

RELATED WORKS

Zeng and colleagues developed a hybrid approach based on observations at the network and host levels. Their framework is such that it first examines and analyzes streams in the network and discovers botnets by their similar traffic [13].
Strayer and colleagues proposed the use of machine learning techniques to detect infected IRC traffic. What they did is summarized in two layers: in the first layer, IRC and non-IRC traffic are separated, and then a distinction is made between botnet traffic and legal IRC traffic [14].
T.Ha and colleagues focused on kademila (a type of P2P botnet). They introduced a new perspective on the discovery of botnets using strategic points. They introduced factors such as degree of centrality, degree of density, the centrality of the Eigen-vector, and centrality based on routing. For example, the centrality of the vector Eigen says that not all connections to a particular host are equally important, but rather the ones that are connected to the most important nodes in the network. By identifying these factors in the network, better choices can be made for monitoring points in the network, because as mentioned earlier, it is not possible to monitor the entire network. It is also possible to determine the bottlenecks that have a significant impact on the life of the bot network [15].

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

In [Shang , 18], it operates on a host-based basis. This method examines the parameters and arguments of the summoned functions of a system that can even be encrypted. This method has inspired the method [17] [18] but with the difference that, unlike Stinson's model, it does not focus on arguments but a series of system call functions, especially those used for biased operations. This method like the techniques used by [19] [20] works on protocols based on Internet chat. This host-based method uses a virtual protector for the analysis process to aggregate host reports to detect bots. Alternatively, [20] using the host-based method, tries to identify one or a set of bots from the report aggregation method. The idea of aggregating reports has been proposed in [24], which has claimed that by aggregating reports from different sources, bots can be identified more accurately. The BotSwat method works on bot behavior. It monitors programs running in the Win32 library list in the host. A key part of this method is the use of Detours, which performs the process of monitoring the calling functions of the programming interface. These methods are all host-based and cannot be used in all operating systems and network architectures.
In [10], aggregation of reports obtained from scanning incoming and outgoing traffic is performed. This method only detects on a network scale and has no sensitivity to detecting key loggers or file-scaling processes at the host scale.

PROPOSED METHOD

The method presented in this research works in the offline mode at the beginning of the work, that is, it only performs learning for a period and creates a prediction model and builds its dataset, then it is put in the online mode. It makes predictions according to the created dataset, and at each stage, this detected behavior of the bot is stored in the database if it does not exist in the dataset before and is a new behavior. These stored behaviors can be used for future predictions. In this way, the proposed method is gradually trained and can work more strongly.
The proposed method described in this section to detect botnet attacks includes different parts, which are:

1. Extraction of characteristics of the current
2. Selection of effective features
3. Network flow aggregation
4. Extraction of off-stream features
5. Removal of noisy data
6. Model production
7. Specifying the threshold value
8. Modeling and extracting models
9. System training
10. Review of new trends

In step 1, streams are continuously received, and steps 2 to 8 are repeated for all received streams. An understandable model is created for each detected flow. Step 9 is done in offline mode to create the overall work and proper training of the system, that is, in this mode, there is no checking of

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

new flow at all, but only training is done and it is determined whether the incoming flow of each is a bot or not. In the online stage, i.e., stage 10, the review is done, and upon receiving the new flow of information, it is extracted, its model is built (stages 1 to 8) and then stage 10, i.e. review is done. If this created model does not already exist in the database, it will be stored in the database according to the prediction made whether or not it is based on a bot.
To be able to perform the diagnosis with more power. This research has been worked on multi-processors, that is, the program works on multiple processors and two threads are placed in each processor. A separate processor is also considered for dpdk, which can only perform detection. The detected traffic is stored in the Redis database, which is a database in primary memory or RAM. Every time the processor processors complete the processing of a stream, they receive the information on a new stream from this Redis database. In other words, step 1 of the proposed method is performed in a completely separate process and the received information is stored directly in Redis. In this way, the processing power also increases. In this research, the tested system must have at least 3 cores, one core is used for the operating system, one core is used for detection and monitoring by dpdk, and one core is used for processing and diagnosis.

Extraction of characteristics of currents

In this part of the proposed method, which is directly related to the network card, the network traffic is received in real-time. Based on the network flow standard, considers a set of packets as a flow. Generally, in this research, we have two types of input information, information that can be extracted using various features in the stream, such as source address, source port, a destination address, destination port, and protocol, and another category that can be extracted from outside the stream such as the size of the flow, the duration of the start to the end of a flow, and the periodicity of a flow, for which the preparation must be done for each, in this step, the features of the first category or information within the flow are extracted. These features include various items that will be introduced in the following process of identifying effective features, and it will be explained how high-impact features are identified. This part was conducted by dpdk. dpdk is a library provided by Intel. Using it, you can work on specific cards specified by this library, such as i40e, ixgbe, ice, etc. In this research, the i40e network card is used, which can receive about 40Gbps. To detect all currents in the network, it has been attempted to turn on the promiscuous mode of the network card. Here, even the traffic whose destination address is not this machine or the network card will be received and the network level can be well monitored. Note that there are currently two main ways to receive traffic and flow from a network with high volume, which is the use of dpdk or netmap libraries. Netmap can receive up to 25Gbps, while dpdk can receive up to 40Gbps. Netmap is a project presented by a team, and it was initially a doctoral thesis and was later expanded, and one of its main problems is the lack of proper documentation. dpdk has a lot of documentation and is also completely stable, for this reason, this library has been used to receive streams in this research. By using this library, all the necessary features can be extracted. This step is performed in two threads, one thread is for receiving network information by dpdk and the second thread is for extracting information from detected traffic and storing it in Redis.

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

Select effective features

All the features extracted from the previous part are not effective; so we should select effective features. Because these features can be of different importance under different conditions and streams, the feature must be selected for different streams each time.
Bots have behavior that is hidden from the detection system. Figure 1 shows bot sality traffic that tunnels its traffic in various network protocols to communicate with its command and control center.

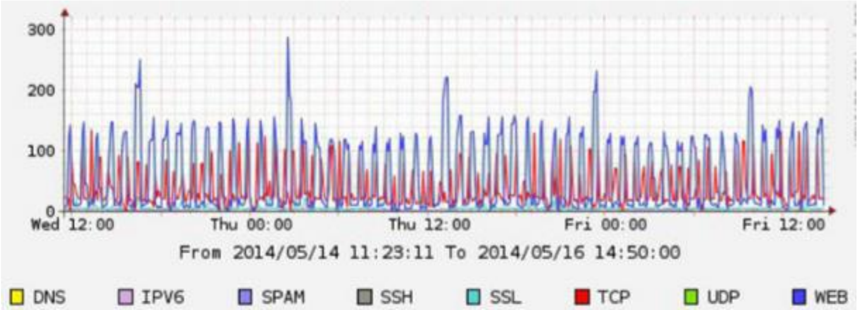


Figure 1: Communication of sality bot tunnel with different communication protocols

As Figure 1 shows, sality uses various network protocols to communicate with its command and control center in encrypted form.
Feature extraction is a technique for removing irrelevant and redundant features and selecting the best subset of features that produce better results of patterns belonging to different classes. Generally, feature extraction reduces the volume of calculations and also reduces noise in some cases, which in turn reduces the error rate and increases the accuracy of predictions.
We use fuzzy rough merged with AHP to extract effective features for low overhead and high accuracy.
A Rough set is a tool that can be used for ambiguity and uncertainty, first introduced by Pawlak (1982). Roughs theory is used in various fields, including decision analysis, and decision support systems. After Mr. Pawlak, three other researchers, Jay, Kho, and Zhang, introduced rough numbers in 2008. A rough has a low limit (L), a high limit (U), and an intermediate limit known as the rough limit distance. Rough numbers are used in matters in which the opinions of experts are involved and in some way create uncertainty and ambiguity.
Suppose that in a decision-making set, set U includes all the members. Y is an arbitrary member of the U and R sets. It is a set of t classes that covers all members of the U. If these classes are in the same order as G1<G2<…<Gt, then the lower, upper, and boundary limits of class G are defined as (1):

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

$$\underline{Apr}(G_q) = \bigcup \{ \mathrm{Y} \in \mathrm{U} \mid \mathrm{R(Y)} \le \mathrm{G}_q \} \qquad (1)$$
$$\overline{Apr}(G_q) = \bigcup \{ \mathrm{Y} \in \mathrm{U} \mid \mathrm{R(Y)} \ge \mathrm{G}_q \}$$
$$Bnd(G_q) = \bigcup \{ \mathrm{Y} \in \mathrm{U} \mid \mathrm{R(Y)} \ne \mathrm{G}_q \} =$$
$$\{ \mathrm{Y} \in \mathrm{U} \mid \mathrm{R(Y)} \succ \mathrm{G}_q \} \cup \{ \mathrm{Y} \in \mathrm{U} \mid \mathrm{R(Y)} \prec \mathrm{G}_q \}$$

This class G can then be represented as a rough number at the lower and upper limits as (2):

$$\underline{Lim}(G_q) = \frac{1}{M_L} \sum R(Y) \mid Y \in \underline{Apr}(G_q) \qquad (2)$$
$$\overline{Lim}(G_q) = \frac{1}{M_L} \sum R(Y) \mid Y \in \overline{Apr}(G_q)$$
$$RN(G_q) = \left\lceil \underline{Lim}(G_q), \overline{Lim}(G_q) \right\rceil$$

Rough's boundary distance is also calculated in Figure 2. This boundary distance expresses ambiguity so that a bigger number, means greater ambiguity, and a smaller number means better accuracy.

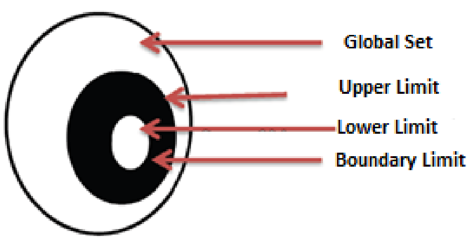


Figure 2: Calculate the limit in the rough set

The AHP (hierarchical analysis process) method is one of the most widely used methods in multi-criteria decision-making. In this model, first, pairwise comparisons are formed and provided to experts to express their views on the comparison two criteria, which are a range of 1 to 9. To use Rough numbers in the AHP (rough AHP) method, we proceed as follows.

1. First, check the pairwise comparisons of the experts in terms of the incompatibility rate, and if the incompatibility rate is less than 0.1, that is, the paired comparison is consistent, and if it is greater than 0.1, the pair comparison numbers should be corrected.
2. Create Rough numbers from expert numbers using the relationships mentioned in the theory.
3. Calculate the distance weight of the criteria using the geometric mean method

Network flow aggregation

In this stage, the source port is one of the influential features, and therefore we try to choose this definitely, and this theorem is obtained using the experimental background. From the input stream under consideration, according to the extracted features of the previous step and the source port, which will be one of the effective features, the common source port is removed in this step so that

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

more suitable inputs can be achieved. The purpose of removing the source port is that a bot mainly uses different ports to communicate with its command and control center. Therefore, the output of this section will be a set of streams with effective properties. As a result, each of these records extracted from the stream with effective properties is called a connection. These connections are used in the later stages of the work.

Extraction of out-of-stream properties

A botnet traffic behavior analysis has three parts: activities, connections, and patterns. Most bots use several different modules for their activities, each of which can generate different patterns of behavior. To analyze all these behaviors, it is necessary to examine the botnet and its generated traffic for a long time, so that during this period, the botnet can update all its behavioral patterns. To do this, we examined the behavior of bots for 57 days, the results of this experiment can be seen in Figure 3. This section is just used for the training part and in detection time it can be sufficient to have just 3 seconds.

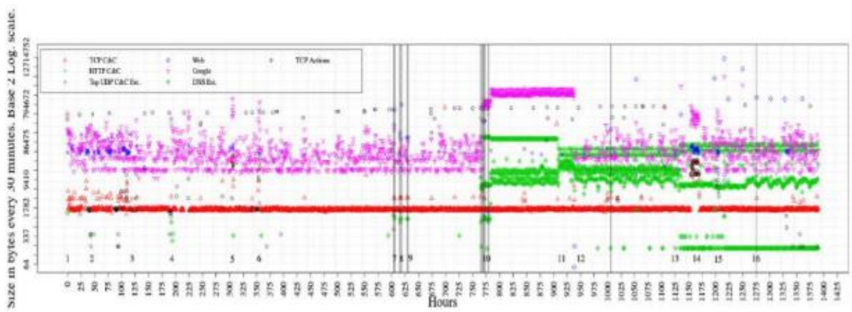


Figure 3: Behavior extracted from botnets during 57 days

In this figure, the vertical column shows the amount of traffic generated, and the horizontal column represents the time. The red line indicates that the botnet continuously generates TCP traffic with a certain volume. This situation is also shown in other protocols with a slight deviation. Therefore, the proposed framework uses the three characteristics of flow size, start-to-end time of a flow, and periodicity of a flow to model the behavior of botnets. These three characteristics have been obtained because of studies and assistance from the article [30].
Since the property of periodicity does not have a specific solution for the calculation. To calculate this feature, we have proposed a solution that can be used to analyze bot traffic and identify attacks. We use (3) to investigate the periodicity of bot behavior based on the flow time characteristics and the volume of transmitted data:

$$Correlation(flow) = \frac{(T_{flow} - \overline{T})(S_{flow} - \overline{S})}{\sqrt{(T_{flow} - \overline{T})^2} + \sqrt{(S_{flow} - \overline{S})^2}} \qquad (3)$$

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

In this regard, T_{flow} the flow time, S_{flow} is the flow volume, $\bar{T}$ the average input time of the previous streams, and $\bar{S}$ the average volume of the previously imported streams. Using this relationship, we can examine the periodicity of the behavior of bots. Considering that this case has already been investigated, a table has been extracted that can be used to examine the results more clearly, which can be seen in Table 1.

Tag	**Correlation Result**
Periodically very weak	0 to 0.19
Weak periodicity	0.2 to 0.39
Intermediate periodicity	0.4 to 0.59
Strong periodicity	0.6 to 0.79
Very strong periodicity	0.8 to 1

Table 1: Classification of flows based on correlation relationship

Noise deletion

In this proposed method, noise data must be deleted, which in this case increases the accuracy. Noise data means data with noise records that records must be removed from the input stream to reduce the overhead and increase accuracy. To be able to detect noise data using the proposed method, the input stream is divided into two parts at the beginning of the work, and their center is specified. After dividing the data into two equal parts, each of these parts is also found in the middle of them, in which case the input stream is divided into four parts by three quarters

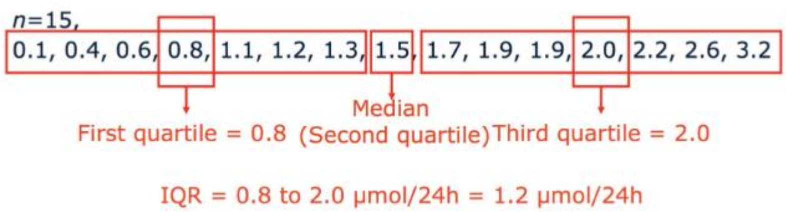


Figure 4: An example of data division in the proposed method

It can be seen in Figure 4 that the data are divided into equal parts and its median is 1.5. the squares on both sides of this median are calculated, which are 2 and 0.8.
To be able to extract better data from all streams and reduce processing time, we select the effective range and call it "the quadratic" value. It should be noted, that this does not mean all values out of these ranges are noise data. Although we identify more efficient ranges that are much more likely to show their original behavior in them and Outside these ranges, the probability of bot behavior is much lower.

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

To be able to calculate the selected ranges between quartiles, (4) was used:

$$IQR = Q_3 - Q_1 \quad (4)$$

To better understand this formula, we can see Figure 5. As it is obvious, the red range is the main behavior range of the bot, which is located in the effective range identified in the proposed method, and the range outside this range is not considered.

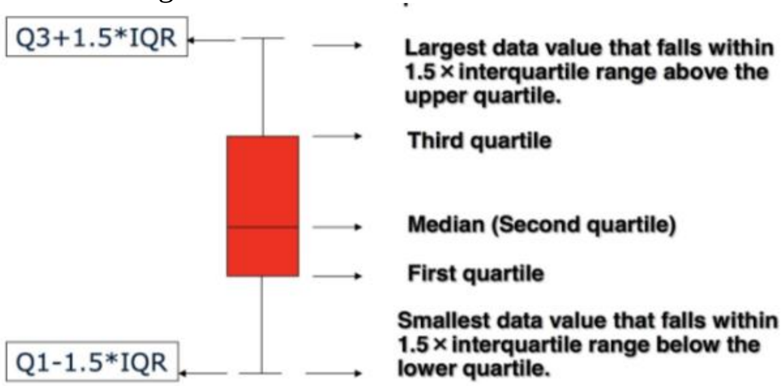


Figure 5: How to calculate the effective interquartile range

It can be seen that here, IQR is calculated using (4), in this example this value is equal to 2-0.8 = 1.2 and on the other hand, the range is equal to the first quarter minus the multiplication of IQR and the third quarter plus the middle Multiply the IQR by which the range is specified ((5)). Most bots do not show their original behavior from the beginning. Rather, they engage in their main behavior in the middle of the stream, so this manner will make it difficult to identify. In Figure 5, the red range is the main behavior that is considered in the domain and as a result, is examined in the range.

As shown in Figure 5, data that are outside the scope of (5) will be removed from the set of training streams so that we can build an accurate model of bot behavior.

$$(utliers > Q_3 + M * IQR) or (outliers < Q_1 - M * IQR) \quad (5)$$

In this formula, M is the same as the middle and outliers are the same outside the range that should be removed from the input stream

Model production

At this stage, behavioral patterns for analyzing network traffic are obtained. Therefore, using the three characteristics of flow volume, flow time, and alternation, the incoming traffic from the network is modeled. The size properties are in bytes and the time interval is in seconds, and these properties can be extracted from any stream, hence we called these properties out-of-stream. Periodicity is present in most botnets due to its automatic nature, and in methods [2], [3], and [4] the same feature has been used to detect botnets. But we should explain a little more about the third characteristic, which shows the periodicity of a flow. The periodicity of each stream can be

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

calculated according to the previous streams. There must be at least three streams in the network to decide on the periodicity of the third stream. In other words, to calculate this property, we calculate the difference between the end time of the first stream and the beginning of the second stream, calling it T1. Similarly, we calculate the time difference between the second and third streams, calling it T2. Equation (5) is used to check the periodicity.

This continues until all streams have been examined and all periodicities have been extracted. This section itself is stored in a format for review. As explained earlier, the proposed method uses two categories of information to identify a bot. this section is the second category of information or the same as out-of-stream information.

Specify the threshold value

In this step, the threshold value is determined to check the out-of-stream properties, which are the values of the quartiles, from which the values are selected as the middle or quarters. Generally, there are two modes for selecting squares used in this research: static and dynamic.

In the static threshold mode, the number of squares is determined using preset values, for example, the value of the first threshold should include 33% of the samples and the value of the second threshold should include 66% of the samples. This issue is also observed in the characteristics of period and periodicity.

Due to the dynamic nature of botnets, static threshold values cannot be efficient enough, and this causes botnets to behave variably. So the best way to deal with behavior dynamics in botnets is to use dynamic threshold values.

In the dynamic mode, unlike the static quantification method, where the threshold values are the same for all botnets, the threshold values will be different for each bot and will be taught in the proposed framework during the training phase.

In this method, by using the training data and examining each feature, as well as classifying them in squares and specifying the available outliers, the threshold values for that bot will be set.

Modeling and extracting models

At this stage, the information obtained from the input is transformed into comprehensible models for review. Models here are created for understanding and training the system. Table 2 uses values for out-of-stream propSerties such as periodicity. For each connection, a model is created or it can be said that we use Table 2 to create a signature for each connection. We use these signatures to detect a bot and in this way, we can save and compare easily

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

Size	Very Big				Big				Middle				Small			
timespan	Very Long	Long	Medium	Short	Very Long	Long	Medium	Short	Very Long	Long	Medium	Short	Very Long	Long	Medium	Short
Strongly periodic	p	n	m	o	l	i	h	g	k	f	e	d	j	c	b	a
Weak periodicity	P	N	M	O	L	I	H	G	K	F	E	D	J	C	B	A
Poor periodicity	‘	[	-	:	?	z	y	x	/	w	v	u	q	t	s	r
Strong non-periodic	“	]	=	;	~	Z	Y	X	\	W	V	U	Q	T	S	R
Lack of sufficient information	&	^	%	$	#	9	8	7	@	6	5	4	!	3	2	1

Table 2: Connection coding table

As can be seen, Table 2 has three characteristics of size, time interval, and periodicity.
In the first line of the table, the dimensions are stated, and in the second line, the time interval is specified, which is between short, medium, long, and very long.
Code “a” means small size, short interval, and strongly periodic. For example, an input stream, which is the connection described earlier, is modeled as follows:

11RrrrrarrrsrrArsrrrrrrRrArrArArrrRArArAArRRrrrr

As can be seen, this model exactly follows the coding table, and in this way, better analysis can be done on the flow and better knowledge can be extracted.

System training

The proposed system is trained using a set of coded streams that have been analyzed using two separate sets, namely, the set of healthy streams and the set of unhealthy streams.
For each set of healthy streams as botnet attacks, the coding is done and the stream is extracted as a code, which is labeled healthy if it is from the white list, and labeled unhealthy if it is from the blacklist, so in this way, the database can be created.
In this stage, we train the model by many attack streams as blacklists and normal streams as a white list to train our model and create a database of signatures. This database can be used to identify new stream categories. That new stream is attacked or not?

Check for new streams

To be able to predict a new stream set, first, all the previous steps must be performed on it to obtain the coding for this new stream, and then the similarity value is calculated using the Markov chain because the Markov chain can subset a set Examine and calculate the degree of similarity.

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

The two-state Markov chain is the most basic model of the Markov chain. In which the Markov chain will have only two modes. This model is commonly used to describe the Markov chain.
Random matrix: Random matrices are used in sciences such as computers, chemistry, and economics. Stochastic matrices can be used to predict different problems in different scientific fields. A random matrix or Markov matrix is a matrix used to describe the transfer of states in a Markov chain. The random matrix is also known as the transition matrix and the probability matrix. Any value stored in this matrix is a nonnegative value between zero and one and represents the probability vector. In other words, the transition from one state to another occurs based on a probability that is located at the corresponding location in the table. Generally, and regardless of the number of states in a Markov chain, there are three types of probability matrices for each Markov chain:

- Right probability matrix: In this type of matrix, the sum of probabilities per row will be equal to 1
- Left probability matrix: In this type of matrix, the total probabilities of each column will be equal to 1.
- Two-way probability matrix: The sum of probabilities in this matrix will be equal to 1 from both the column and row.

We should examine the proposed solution for all behaviors in the database that we have already created during the training, using the Markov chain. Generally, the Markov chain determines the degree of similarity between the two disciplines. If there is a similarity higher than 70% from input by an item in the database we created before, then the input should be a category we determined in the database.
This is because the trained Markov chain may differ in length from the Markov chain observed in traffic, and there may be a situation in the sample we are monitoring that does not exist in the trained chain. In these cases, the probability of error may be calculated or zero, respectively, so the nonexistent and additional case is not considered in calculating the probability of producing the main case chain. This technique in our method enables the detection of polymorphic bots that attempt to mislead the system by creating different traffic. For example, in the following two sets, it can be seen that the modeled string in the database includes 81aaabababababa, and while examining the second string, IE:

81aaaabf0fababababababafxzfxfx

It can be seen that this second string includes the first string, but to be able to hide, he has tried to divide his main behavior, but in the Markov chain, due to the nature of not paying attention to the past, it can easily be able to resemble this type of strings.

Train Chain: 81aaabababababa
Monitor Chain: 81aaaabf0fababababababafxzfxfx

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

Evaluation

The network that is considered for the investigation in this research is an internal network and is created by two machines, that is, one machine is used to send traffic and read from the list of different traffic, and the second machine is used to receive from the card. The network performs the processing. These two machines are connected using a switch and the connection cable is 40 gigabytes so that the maximum capacity of the network cards can be used. Each of these machines had an i40e card. According to figure 6, machine 1 starts sending a set of traffic that is already stored and has the extension pcap. For sending, the dpdk library is used, which can send traffic at a very high speed. 4 rings are used for sending and receiving by network cards. The ring is for increasing the speed that is available in the network cards. The i40e network card can support up to 16 rings, although note that this is not the case, the higher the number of rings, the faster the speed will be because this calculation is also performed to divide between the rings. Based on experience and the use of tools such as ethstatus, 4 rings were considered in this section. Machine 1 transmits using 4 rings and sends it to the switch. The second machine receives all traffic from the switch because its promiscuous is turned on, and in this case, all traffic is received even if its destination is not this machine.

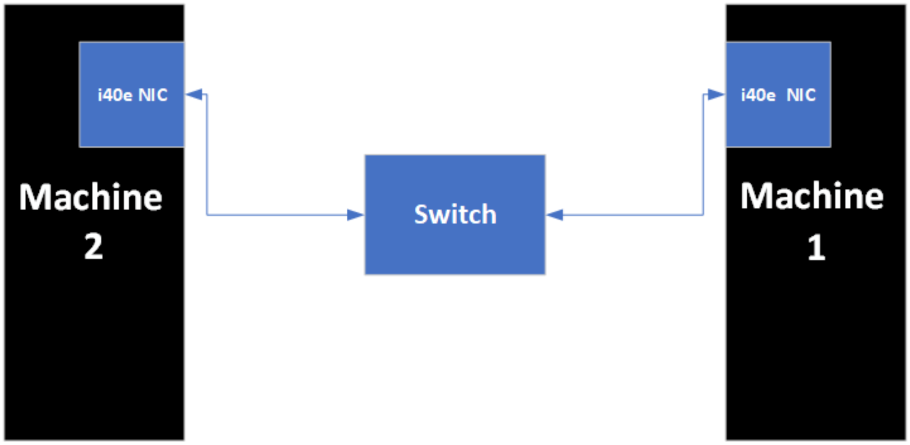


Figure 6: Architecture of considered machines

The implementation was performed in Visual Studio 2017 and with C# programming language. In this implementation, the help of Weka [31], ZedGraph, and Dpdk libraries has been taken.

Datasets

We use the following two datasets to instantiate our model. 1) The SysNet Dataset: The SysNet trace was generated by the SysNet lab in July 2013 and contains traffic from ten infected hosts. It was generated by running ten bot binaries in separate virtual machines: four variants of Pushdo, and two variants of Sality, Kolabc, Virut, Dorkbot, and Bobax. This covers HTTP, IRC, and P2P-

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

based bots that engage in a range of attacks, including sending spam and outbound scanning. Full packet traces from the virtual machines were captured for 24 h using DPDK on host machines. 2) The ISCX Botnet Dataset: This trace was collected by researchers at the ISCX Laboratory at UNB, Canada, and made available publicly. It contains traces of 30 infected machines, nearly half of which are infected by IRC Botnets and the remainder by various P2P and HTTP Botnets, including variants of Zeus, Virut, NSIS, Storm, and Zero Access among others.

The data is stored as a pcap file, and considering how many of them are attacks and how much healthy data is known, the strength of a method can be measured. In this implementation, dpdk is used to send pcap packets at high speed.

In this implementation, the input streams were first collected using the proposed solution and data analysis was generated using the flow analysis, which is then given as input to the system. With these inputs, we create a database. In this database, it has been operated in a supervised manner, input streams have been categorized as attacks and non-attacks, as well as the type of attack subcategory has been precisely specified. In this section, the purpose of training the system and then using the cross k-fold solution is to examine the proposed solution. The resulting database contains 55,940 records and was saved as a CSV file, which is a completely standard format for storing high-volume records. The data saved in the database are labeled so the work is supervised. Here, the value of k for the method using the cross k fold method is set to 10, which is a normal value in the field of data mining. In the past solutions, the split method was used, in which 66% of data were usually considered as training data from the beginning and the rest as test data, but in this case, behaviors may be seen in the test set that did not exist in the training set at all. This cross-k fold is more popular because all different parts of the dataset are used for training and testing, and this is ultimately the average of the various stages of the study of this method, and it has been experimentally proven that k = 10 is the best value for k. In this study, k is considered equal to 10.

Analysis of results

In this section, the proposed method is compared with the methods [9], [25], and [30] stated in the third chapter.

In [9], Gu et al. presented a method based on clustering to detect botnets in the attack phase. In this method, first, similar communication traffic and similar malicious traffic are clustered, and then an inter-cluster correlation is performed to identify hosts with similar malicious activity patterns. The above method works offline, which is a major weakness in botnet detection systems. Also, if the bots of a botnet perform a new malicious activity during the attack phase, this method will not be able to detect that botnet.

In [25], Wang et al. presented a behavior-based system for detecting botnets, which is based on fuzzy pattern recognition techniques and uses individual-level analysis. The main idea of their method is based on the identification of malicious domain names and IP addresses used by botnets, which are obtained by inspecting network flows. This method consists of three stages: traffic reduction, feature extraction, and fuzzy pattern recognition. First, the network traffic enters the traffic reduction stage, and after filtering, it is sent to the feature extraction stage. Finally, the phase

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

of fuzzy pattern recognition identifies botnet-related activities based on the extracted feature vectors belonging to several membership functions. These membership functions include:

1. Generating failed DNS requests
2. The similarity in the intervals of sending DNS requests
3. Generation of failed network connections
4. They all had the same body size in their network connections.

In [30], the method is based on the network and is independent of the structure and protocols of command and control servers. These methods cluster bots based on their traffic similarities. After the completion of the clustering process, the way the clusters communicate with each other is also examined to determine whether there is a connection between the biased activities of one cluster with another cluster or not. If the attacks are mixed and diverse, due to the use of pattern-based methods, after clustering, the probability of bots not being discovered increases.

After running for a certain period of about 1 min, the outputs of Figure 7 to Figure 10 are created in the program, which is related to each method.

```
--------Article 9----------
Correct Prediction Percent = 61.1440829460136%
InCorrect  Prediction Percent = 38.8559170539864%
MeanAbsoluteError(MAE) = 0.475162871420202
MeanSquaredError(MSE) = 0.487422756288312
RelativeAbsoluteError(REA) = 100
Correct Prediction Number = 34204
InCorrect Prediction Number = 21736
```

Figure 7: Output of article[6]

```
--------Article 27----------
Correct Prediction Percent = 68.6145870575617%
InCorrect  Prediction Percent = 31.3854129424383%
MeanAbsoluteError(MAE) = 0.313495376602283
MeanSquaredError(MSE) = 0.556911737300665
RelativeAbsoluteError(REA) = 65.9764041885575
Correct Prediction Number = 38383
InCorrect Prediction Number = 17557
```

Figure 8: Output of article[25]

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

```
--------Article 30----------
Correct Prediction Percent = 81.0904540579192%
InCorrect  Prediction Percent = 18.9095459420808%
MeanAbsoluteError(MAE) = 0.191982043405808
MeanSquaredError(MSE) = 0.404881303106203
RelativeAbsoluteError(REA) = 40.4034184808917
Correct Prediction Number = 45362
InCorrect Prediction Number = 10578
```

Figure 9: Output of article[30]

```
--------Proposed Algorithm----------
Correct Prediction Percent = 96.1708973900608%
InCorrect  Prediction Percent = 3.82910260993922%
MeanAbsoluteError(MAE) = 0.0665537182695683
MeanSquaredError(MSE) = 0.174667276671453
RelativeAbsoluteError(REA) = 14.0065064575898
Correct Prediction Number = 53798
InCorrect Prediction Number = 2142
```

Figure 10: Output of proposed method

According to the received results, it can be seen that the proposed algorithm has the highest accuracy with 96.170% accuracy and the lowest error rate with 3.829% error. A better ratio can be seen in table 3. The proposed algorithm performed much better than the rest of the methods.

	Prediction error percentage	**Percentage of predictions**
[Research [6	38.855%	61.144%
[Research [25	31.385%	68.614%
[Research [30	18.909%	81.090%
Proposed method	3.829%	96.170%

Table 3: Ratio of prediction accuracy and prediction error

As can be seen from the diagram in Figure 11, the proposed method has a more accurate detection accuracy than the other algorithms under consideration, namely, [6], [25], and [30]. This is because, in our prediction method, we considered only those items of test data that had a greater impact on the output, and therefore did not use data that did not affect the output, thus reducing the analysis time. Also, we get a better signature to check streams.

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

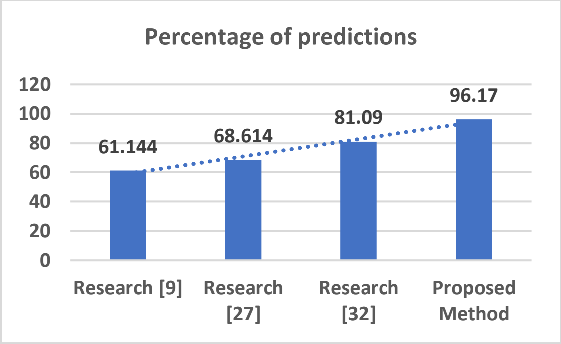


Figure 11: Percentage of true predictions

In the diagram presented in Figure 12, the percentage of misdiagnosis can be seen. Figure 12. Percentage of incorrect prediction among test data for the proposed algorithm and other algorithms under consideration.

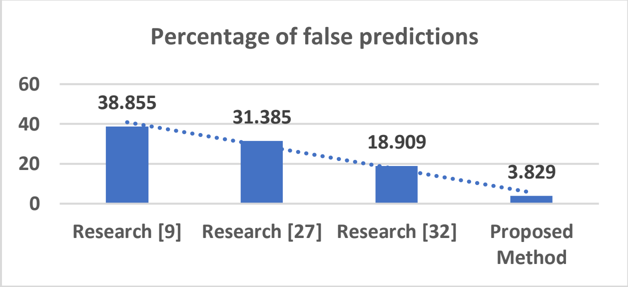


Figure 12: Percentage of false predictions

In Figure 13, the mean squared error (MSE) is calculated for the proposed method and other methods being compared. This criterion obtains the average square error using (6) and using this criterion can accurately measure the prediction and calculate the prediction error rate. In this implementation, this relation is used to calculate the prediction error to determine to which extent the method we introduced for predicting the botnets can perform the predictions correctly:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\mathrm{r}_i - \mathrm{p}_i)^2 \qquad [35](6)$$

where N is the total number of test data, r is the actual value and p is the predicted value.

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

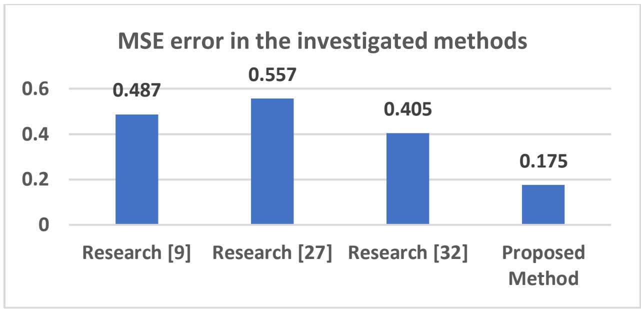
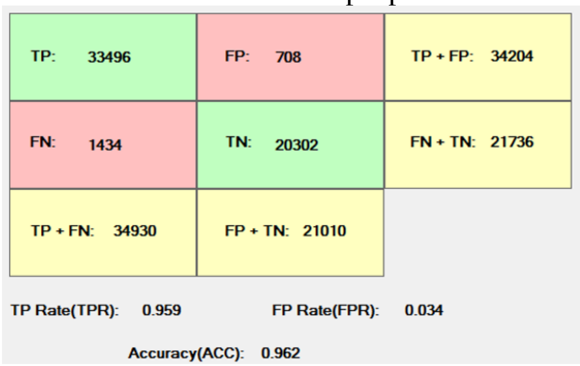


Figure 13: MSE error in the investigated methods

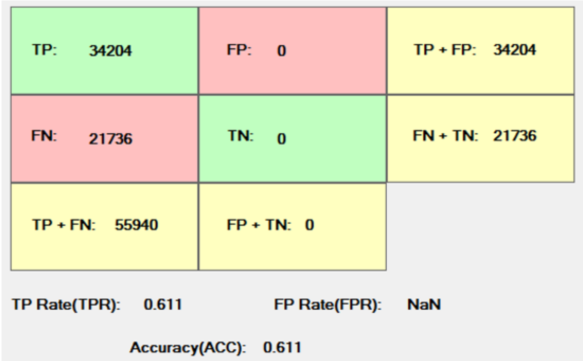
The following is the Confusion Matrix related to the proposed method and others.



TP: 33496	FP: 708	TP + FP: 34204
FN: 1434	TN: 20302	FN + TN: 21736
TP + FN: 34930	FP + TN: 21010	

TP Rate(TPR): 0.959 FP Rate(FPR): 0.034
Accuracy(ACC): 0.962

Figure 14: Confusion matrix of the proposed method



TP: 34204	FP: 0	TP + FP: 34204
FN: 21736	TN: 0	FN + TN: 21736
TP + FN: 55940	FP + TN: 0	

TP Rate(TPR): 0.611 FP Rate(FPR): NaN
Accuracy(ACC): 0.611

Figure 15: Confusion matrix of research [6]

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

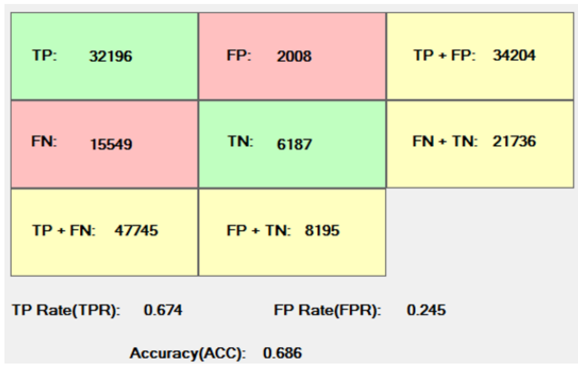


Figure 16: Confusion matrix of research [25]

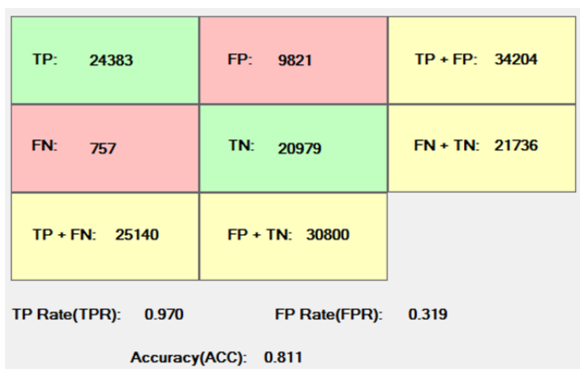


Figure 17: Confusion matrix of research [30]

The proposed method has a higher accuracy because Figures 14-17 it has more TP and TN than other algorithms and less FP and FN than the other algorithms studied here. The higher the TP and TN of an algorithm, the more correctly it detects the attack or not in the test data set, and FP and FN indicate the opposite, which means that it is incorrect.

Conclusions

In this research, a method for evaluating and predicting attacks was presented and it was observed that the presented method has a good performance and a relatively high improvement compared to the algorithms of articles [6], [25], and [30].
In the proposed solution, dpdk was used to send and receive traffic at a very high speed, and a mode was activated in the receiving card that can receive all traffic, in which case the traffic is received even if it is not related to this machine. . This information is stored in a database that works in RAM, and then information from this database is received and processed by computer threads. With this work, traffic can be monitored and checked at very high speeds. It is also

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

considered that for each traffic if the behavior has not been seen before, storage is done so that the system can be gradually trained. In the processing part of the proposed solution, Markov is used, so that the diagnosis can be performed with a small amount of memory, and to check the traffic correctly, the middle solution is used. In this research, a table is presented, using which a label is extracted for each stream, and according to that extracted label, the diagnostic process is performed using Markov.

Future Work

In near future we can work on some suggestions:

1- In the solution used here, some features have been selected and by using them, an attempt has been made to create a signature for each stream, while it is also possible to select a feature at this stage, IE, from the available features. Choose more efficient features and create signatures using these features to reduce memory and computational overhead.
2- Instead of using the Markov chain, decision tree-based solutions can be used, and finally use the decision tree to examine and build the model, as a result of which very useful decisions can be extracted in which cases it is possible. Whether it is an attack or not, very useful information can be obtained

References

[1]. Mohammad Khanjani, "Software Blurring by Multi-Yarn Petri Nets", 20th Annual National Conference of the Iranian Computer Association, March 2015.

[2]. P. Miller, “Hybrid Analysis and Control of Malware,” Computer Sciences Department, 2017.

[3]. J. B. a. Z. Bosnić, "Extending applications using an advanced approach to dll injection and API hooking," Practice and Experience Journal, Volume 40, pp. 567-584, 2010.

[4]. M. Vaziri, “Finding Bugs with a Constraint Solver,” MIT Laboratory for Computer Science, Massachusetts, 2018.

[5]. Hex-Rays. IDA Pro. https://www.hex-rays.com/products/ida/, 2022 Last access: March 18, 2022.

[6]. Ang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2015. Introduction to Data Mining, (First Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[7]. Yu, X., Dong, X., Yu, Ge, Qin, Yuhai, Yue, D., "Data-adaptive Clustering Analysis for Online Botnet Detection", IEEE Third International Joint Conference on Computational Science and Optimization, Vol. 1, pp. 456-460, 2017.

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 *Revised: 12-02-2024* *Accepted: 07-03-2024*

[8]. Shahrestani, Alireza, Feily, Maryam, Ahmad, Rodina, Ramadass, Sureswaran, "Discovery of Invariant Bot Behavior through Visual Network Monitoring System", IEEE Fourth International Conference on Emerging Security Information, Systems and Technologies, pp. 182-188, 2017.

[9]. Gu G., R.Perdisci, J.Zhang, and W.Lee, "BotMiner: Clustering Analysis of network traffic for Protocol- and Structure- Independent Botnet Detection", in Proceedings of the 17th USENIX Security Symposium, San Jose, CA, USA, 2018.

[10]. Ha Duc T., Yan Guanhua, Eidenbenz, Stephan, Ngo, H.Q. "On the Effectiveness of Structural Detection and Defense Against P2P-based", IEEE dependable systems and networks conference, pp. 297-306, 2019.

[11]. Kira, Kenji and Rendell, Larry (2016). The Feature Selection Problem: Traditional Methods and a New Algorithm. AAAI-92 Proceedings.

[12]. M. Antonakakis, C. Elisan, D. Dagon, G. Ollmann و E. & Wu, “The Command Structure of the Aurora Botnet,” Damballa, 2010.

[13]. Zeng, Y., Hu, Xin, G. Shin, K., "Detection of Botnets Using Combined Host- and Network-Level Information".IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), pp. 291-300, 2017.

[14]. Livadas, C., Walsh, R., Lapsley, D., Strayer, W.T., "Using Machine Learning Techniques to Identify Botnet Traffic", IEEE Internetwork Research Department BBN Technologies, proceeding 31th IEEE conference, pp. 967–974, 2016.

[15]. Foladi, Zeinab; Seyed Hassan Hani Zavareh Tabaei; Yaghoub Farjami and Jalal Rezaei Noor, Discovery of botnets based on network traffic behavior, the first national conference on new approaches in computer engineering and information retrieval, Rudsar, Islamic Azad University of Rudsar and Amlash Branch, 2021.

[16]. S., W. Y.Shang, "Botnet Detection with Hybrid Analysis on Flow-Based and Graph-Based Features of Network Traffic," International Conference on Cloud Computing and Security, pp. 612-621, 2018.

[17]. E. Stinson و J. C. & Mitchell, "Characterizing bots' remote control behavior", detection of Intrusions & Malware, and Vulnerability Assessment, 2022.

[18]. Z. Chi, z.Jin. Ch.Zheng, "botnet detection based on behavior analytics," pp. 612-621, 15 03 2018.

[19]. W. T. Strayer, R. Walsh, C. Livadas و D. & Lapsley, “Detecting botnets with tight command and control,” In Local Computer Networks, Proceedings 31st IEEE Conference on, 2016.

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

[20]. J. Goebel, T. & Holz, “Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation,” HotBots, 2017.

[21]. B. Qi, J. Jiang, Z. Shi, R. Mao, and Q. Wang, " Detecting DGA-Based Botnet Using Two-Stage Anomaly Detection," in IEEE, New York, NY, USA, 2018.

[22]. P. G. Efthimion, S. Payne and N. and Proferes, "Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots," SMU Data Science Review, vol. 1, p. 52, 15 03 2018.

[23]. A. Karasaridis, B. Rexroad و D. Hoein, “Wide-Scale Botnet Detection and Characterization”, Workshop on Hot Topics in Understanding Botnets, 2017.

[24]. T. Cochran, J. Cannady, “Not so fast flux networks for concealing scam servers”, in Risks and Security of Internet and Systems (CRiSIS), 2010.

[25]. Wang K., C. Huang, S. Lin, and Y. Lin, "A fuzzy pattern-based filtering algorithm for botnet detection", Computer Networks, Vol. 55, No. 15, pp. 3275–3286, 2021.

[26]. B. A. AlAhmadi and I. Martinovic, "Malware family classification using network flow sequence behaviour", in APWG Symposium on Electronic Crime Research, San Diego, CA, USA, 2018.

[27]. A. Cabri, F. Masulli, S. Rovetta, and G. Suchacka, "A Quantum-Inspired Classifier for Early Web Bot Detection," in IEEE Transactions on Information Forensics and Security, vol. 17, pp. 1684-1697, 2022, DOI: 10.1109/TIFS.2022.3170237.

[28]. Z. Ellaky, F. Benabbou, S. Ouahabi, and N. Sael, "A Survey of Spam Bots Detection in Online Social Networks," 2021 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA), 2021, pp. 58-65, DOI: 10.1109/ICDATA52997.2021.00021.

[29]. Giang L. Nguyen, Braulio Dumba, Quoc-Dung Ngo, Hai-Viet Le, Tu N. Nguyen, “A collaborative approach to early detection of IoT Botnet”, Computers & Electrical Engineering, 2022, pp. 107525, vol. 97, 10.1016/j.compeleceng.2021.107525.

[30]. V. I.Ghafir, "A System for Real-Time Botnet Command and Control Traffic Detection," Cyber-Threats and Countermeasures in the Healthcare Sector, vol. 6, pp. 38947 - 38958, 2018.

[31]. Ian H. Witten, Eibe Frank, Mark A. Hall, Christopher J. Pal, “Data Mining: Practical Machine Learning Tools and Techniques”, fourth edition, 2016.

[32]. O. Dallas. Inc, “The Role of DNS in Botnet Command and Control”, 2021.