IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ICRTTC '23

International Conference on Recent Trends and Technologies in Computing

Department of Computer Science and Engineering, Arasu Engineering College

Vol. 12, Special Issue 1, April 2023

Data Leakage threats and malware in web development

S. SelvaShanthi[1], S. AntonyRaja[2], K. Vijayaprabu[3]

Department Of Computer Science &Engineering, M.I.E.T. Engineering College, Trichy[1-3]

Abstract: In recent years, privacy data leakage has become a hot topic of security. After malware controls the target client, it needs to bypass the existing security defense to transmit the data to the controller.DNS is a common communication protocol in the network, thus traditional defense methods will not place strict restrictions on DNS traffic. Researchers have found various domain requests for covert data transmission in DNS traffic. In the past ten years, people have only noticed the communication of DNS tunnels, but the new kind of DNS data leakage has a more covert transmission mode through sub-domain name coding and other ways. DNS data leakage exhibits low traffic volume and periodicity, which is totally different from DNS tunnels with bi-directional data exchange and high traffic volume. In this paper, a detection model named as LSTM-AE is proposed. LSTM-AE integrates LSTM-based time-series characterization and unsupervised auto encoder to detect data leakage malware through DNS traffic. Experimental results show the detection performance of LSTM-AE is better than other ML-based methods and several unknown malicious domains related data leakage have been detected with real-world DNS traffic.

Index Terms: Data Leakage, Malware detection, Anomaly Detection

I. INTRODUCTION

Personal computers and computer networks have always been the targets of data theft attacks. These attacks usually use man-in-the-middle attacks or malicious software that divulges data through secret channels. Usually, in the case of malware, a remote server acting as a command and control(C&C) waits for communication from the malware and records the data transferred to it. However, in a protected network (private or organizational), the target host can reside in a restricted network segment with limited access to the outside world. In this case, even if connections are allowed, it is usually the security solution that monitors suspicious behavior.

Therefore, in this case, it is necessary for malware to find a covert channel to leak data to the remote server, and the existing security solutions will not block or detect the channel. One way to achieve this goal is the domain name system (DNS) protocol. From the attacker's point of view, DNS protocol is a convenient covert communication channel for data leakage. In the past decade, people have carried out extensive research on covert channel detection [1-8]. However, as far as we know, the previous work mainly focused on detecting communication with DNS tunnels and generally used a ML-based detection model. The model would consider the tunnel communication characteristics of bidirectional data exchange and high throughput. However, it may fail on detecting DNS data leakage with low traffic volume. In fact, our research shows that atleast seven known malware families use DNS for covert data communication in real-world traffic, but traditional methods completely failed in detecting them. In recent attacks, data leakage caused by malware can lead to many serious data problems, such as privacy information and password of credit cards were stolen, crucial and com-promised machines were controlled, personal health data of the medical system was stolen. And it also may force an installation of other malware, so its importance should not be underestimated. Quite a few data leakage malware detected in recent years use Internet domains that are purchased, registered, and operated specifically for network activity. The same is true for DNS tunneling tools, where users need to provide and configure Internet domains. Therefore, rejecting requests from these domains is equivalent to preventing data leakage without affecting normal network operation. Our main contributions are summarized as follows:

· The LSTM-AE model is proposed for detecting DNS data leaks. The model integrates LSTM-based time-series characterization and unsupervised auto encoder, which can adapt to the dynamic variability of DNS traffic and reduce the need for manual labeling.

· Previous DNS data leaks have mainly targeted DNS Tunnel, where the data volume is transmitted more, and less research has been done on DNS ex-filtration, where the data volume is transmitted less. In this paper, we use time-series features to target DNS ex-filtration detection, which can detect a transmission volume of four queries per hour accurately.

· Combining time-series features with domain features, LSTM-AE can outperform other ML-based methods and find unknown hidden DNS data leakage in practical deployments.

·

II. RELATEDWORK

The DNS protocol is a naming system for hosts and an important part of Internet infrastructure. A large number of domains and sub domains on the Internet will not only have the storage capacity of a small, simple database. The designers of DNS foresaw this, and the system was designed as a hierarchical distributed database. The DNS is designed as a stateless protocol for exchanging very short and specific types of information. The original intention of this protocol is not to transmit information from the client to the server in an interactive manner. However, the AuthNS(authorized name server) sees all queries from the clients to the domains delegated to it, and if the queries follow a certain pattern, there quested sub domain can be interpreted as data. In addition to the ability to send data to the AuthNS as described above, one can also leverage the response to form a bidirectional interactive data channel. Therefore, for the low cost of purchasing a domain and configuring a server as its AuthNS, an attacker can abuse the DNS, enabling it to serve as an interactive communication channel between a querying machine and the server.

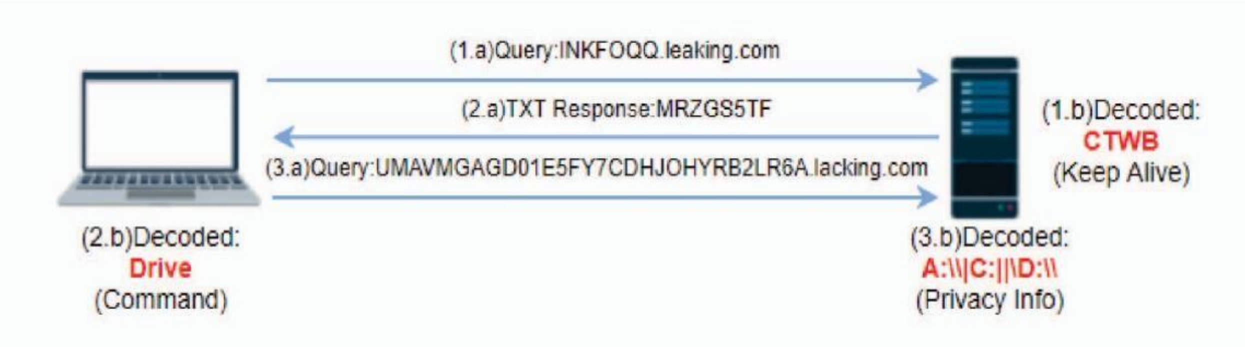

(1.a)Query:INKFOQQ.leaking.com
(2.a)TXT Response:MRZGS5TF
(3.a)Query:UMAVMGAGD01E5FY7CDHJOHYRB2LR6A.lacking.com
(1.b)Decoded:
CTWB
(Keep Alive)
(2.b)Decoded:
Drive
(Command)
(3.b)Decoded:
A:\\|C:||\D:\\
(Privacy Info)

Fig.1. Data leakage between client and attacker's server over DNS covert channel

For example, if the domain was queried, the AuthNS for string and can interpret it as incoming data. (see the illustration in Fig. 1).And this is the so-called DNS tunneling. Malicious DNS tunneling is a bridge for data exchange between malicious malware and malicious server. This type of abuse of the DNS protocol for the sake of data exchange has been thoroughly studied in previous research along with its unique attributes [9-11]. The most commonly investigated unique attributes are: long queries and responses different resource record distribution, and a high volume of requests and encoded data rather than plain text. While these abnormalities may capture the entire landscape of data exchange over the DNS, they are insufficient for accurate data leakage detection, since not all data exchange is malicious.

Other works have leveraged anomaly detection to overcome the problems encountered in the above-mentioned works. Cambiaso et al. [12] refer to the entire DNS communication as a whole and look at sliding windows of requests and responses, extract features and then reduce dimensionality using Principle Component Analysis(PCA).

Engelstad et al [13] evaluate two anomaly detection techniques to detect tunneling over mobile DNS. Their findings show that OCSVM (one-class SVM, used when there is only one type of data, and the goal is to test the new data and detect whether it is similar to the training data.) out performs k-means in this scenario. While offering a generic detection approach for one-class anomaly detection, only two techniques Were considered (clustering and distance). McCarthyetal.[14] suggest a Markov decision process to infer what network nodes sensors should be activated for the detection of DNS exfiltration. The difference between legitimate and malicious DNS tunneling is emphasized in Wang et al. [15]. In this work, Wang points out that the detection of tunneling is simple enough since the volume of requests is high, however, the distinction between types of tunneling is hard. He suggests a white list in the form of a legitimate tunneling directory in which reputable DNS tunneling services must register in advance and otherwise would be denied. Although above approaches deal well with the detection of some DNS tunneling tools they are limited in two aspects:

(1) data leakage with low throughout, which causes a high false negative rate; (2) share similar behavior features between benign and malicious DNS channel, which causes a high false positive rate. In section III, we proposed our detection method using an anomaly detection model assisted with time-series features. Anomaly detection model can detect low throughput data leakage which was hard for supervised model to detect. Time-series features help us separate channels that have similar behavior on basic features and reduce false positive rate.

III. DETECTING DATA LEAKAGE

The architecture of the detection system is illustrated in Fig. 2. First, our passive DNS traffic data are collected from Shanghai Jiao Tong University DNS server. Then we use a sliding window of λ minutes to separate the traffic into window slides for statistic features extracting. In every window slide, we filter out benign queries by Alexa top 10,000 domains. The query of these domains makes up a time-series. We trained a LSTM based Auto encoder on time-series with reconstructing loss as a loss function. After training this LSTM based Auto encoder, the final features of time serials is output. Meanwhile, traditional domain features is also extracted from sliding window. Then with an auto encoder model, Data leakage channels will be detected according to anomaly scores.

A. Extracting Time Series Features with LSTM based Auto encoder

We use LSTM based auto encoder to extract the time-series features of domain name resolution requests automatically.

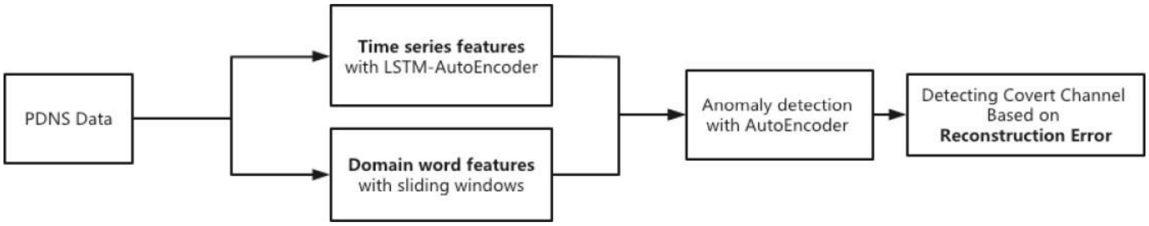


Fig.2. Architecture of detection system. DNS traffic is separated and grouped every λ minutes.

LSTM is a variant of traditional RNN, which retains the time dimension data of sequence data by connecting neurons to a network with direct circular input shows the basic structure of the LSTM storage unit, which has two different components: the long-term state component c(t) and the short-term state component h(t). The storage unit consists of three control door input, output and forgetting gates, which perform write, read and reset functions in each unit. The multiplication gate allows the model to store information for a long time, thus eliminating the gradient disappearance problem observed in RNNs. As illustrated in Fig. 3, the time sequence information such as the number of domain name resolution requests is counted as a time-series according to the time window, which is firstly input into the LSTM network of the first layer. During the training process, the LSTM network will output the LSTM hidden layer information of the short-term state component of the specified dimension, and this layer information will be input into the hidden layer of the whole auto encoder through the full connection layer. Then, through the same structure of all join layers and LSTM layers, we get the same length sequence as the original sequence. The loss functions of the whole LSTM based Auto encoder is defined as the MSE of the output sequence and the input sequence trained using gradients of reconstruction loss. After training, the algorithm will output an encoder and decoder. We feed all traffic data into encoder, and finally get all-time series features. Using such features, it is easier to separate at a leakage channels from legitimate ones.

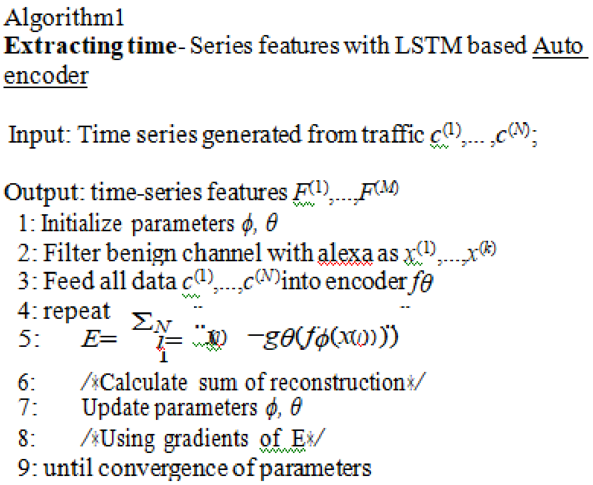
In the training phase, we train the normal traffic input model, and the model performs gradient descent training according to the reconstruction error of the reconstructed time-series. When the model converges, the hidden layer information of the model is taken as the time-series feature of the domain name.

B. Domain Feature Selection with sliding window

We use a sliding window of λ minutes to separate the traffic in to window slides for statistic domain features extracting. Λ is determined according to the realistic traffic rate. Based on a number of related literature [16-18], we choose a number of characteristics ,as shown in Table I.

a) Domain Entropy : In information theory, the entropy of a random variable is the expectation of information that represents uncertainty in the possible outcome of the variable. Domain entropy expresses the randomness of each character in the domain name. The more random the domain name, the higher the entropy value. Domain Entropy is a kind of Characteristic that relates to the words. The training algorithm for the auto encoder is shown in Algorithm 1. First, DNS traffic is processed into time-series.

b) Unique Query Ratio: A domain whose sub domains are used as messages is not likely to repeat them. Therefore, when comparing domains used for exfiltration to normal primary domains we expect to see a much higher unique query ratio for the latter. Unique Query Ratio is a kind of characteristic that relates to the query to DNS server.



Algorithm1
Extracting time- Series features with LSTM based Auto encoder

Input: Time series generated from traffic $c^{(1)},\ldots,c^{(N)}$;

```
Output: time-series features $F^{(1)},\ldots,F^{(M)}$
Initialize parameters $\phi$, $\theta$
Filter benign channel with alexa as $x^{(1)},\ldots,x^{(k)}$
Feed all data $c^{(1)},\ldots,c^{(N)}$ into encoder $f\theta$
repeat
    $E = \sum_{i=1}^{N} \| x(i) - g\theta(f\phi(x(i))) \|$
    /*Calculate sum of reconstruction*/
    Update parameters $\phi$, $\theta$
    /*Using gradients of E*/
until convergence of parameters
```

c) *Unique Query Volume:* In a normal condition, DNS traffic is rather sparse as responses are largely cached within the stub resolver. However, for the case of data exchange over the DNS, the domain-specific traffic is expected to avoid cache by non-repeating messages, or short time-to-live in order for the data to make it to the attackers server. Avoiding cache, as well as lengthy data exchange, might result in a higher volume of requests compared to the normal one.

d) *Query Length Average :* As complementary to the volume feature and given a query size limitation, there is a trade-off between the volume of queries and their length. We, therefore inspect both for anomalies.

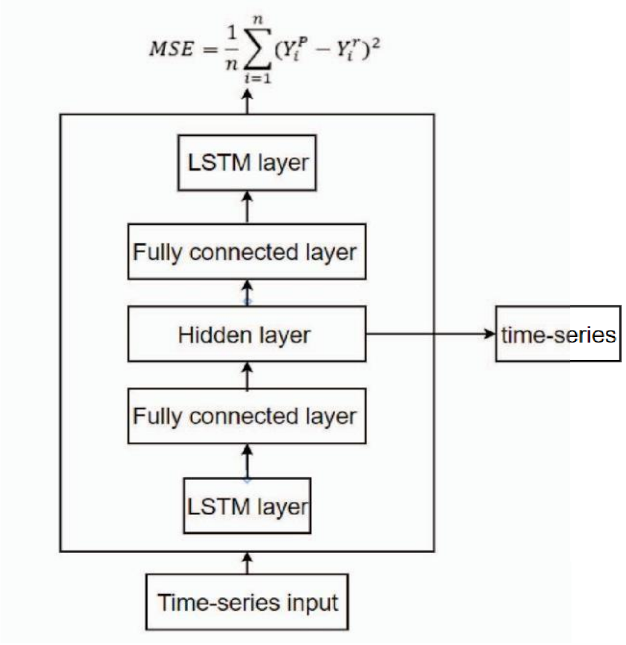


Fig.3. Time-series Extraction with LSTM based Auto encoder

a) *Label Count:* To bypass simple security counter measures, malware may use three or more labels in the sub domain. When data is transmitted, the information is usually encoded in third or even fourth sub domains, which can avoid simple detection on the first label.

b) Percentage of Readable Words: Because this kind of sub domain encoding data leakage usually uses encoding methods such as base32 or base64, the readability is much different than that of normal domain names. Therefore, we match the proportion of readable words in each domain namebasedonthecommonlyusedworddatabaseasahighlydistinguishablefeature.

FeatureName	Types	Description
Entropy	Lexical	High,disordered
UniqueQueryRatio	Resolve	Morethanhalf
UniqueQueryVolume	Resolve	Highvolume
QueryAverageLength	Resolve	Longstring
LabelCount	Lexical	High,morethan1
PercentageofReadableWords	Lexical	Fewreadablewords

TABLE I DOMAIN FEATURES ELECTION AND DESCRIPTION

C. Anomaly Detection Based on Auto Encoder Model

An auto encoder is composed of two parts, an encoder and a decoder. A neural network with a single hidden layer has an encoder and decoder as in Formula (1) and Formula (2), respectively. W and b are the weight and bias of the neural network and σ is then online or transformation function.

$$h=\sigma(W_{xh}x+b_{xh}) \quad (1)$$
$$z=\sigma(W_{hx}x+b_{hx}) \quad (2)$$
$$\|x-z\| \quad (3)$$

The encoder in Formula (1) maps an input vector x to a hidden representation h by a nonlinear finite mapping. The decoder in Formula (2) maps the hidden representation h back to the original input space through the same transformation as the encoder. The difference between the original input vector x and the reconstruction z is called the reconstruction error a sin Formula(3). The auto encoder model learn show to minimize reconstruction errors and can infer the input according to there construction errors. In other words, it attempts to reconstruct the traffics features, and computes the reconstruction error in terms of mean squared errors (MSE). Anomaly detection uses MSE as an anomaly metric. Generally, a threshold value of MSE is determined by training auto encoder model.

In this paper, the output of LSTM-AE is the MSE anomaly score. The larger the score, the greater the anomaly. We evaluate LSTM-AE detection capabilities based on its MSE score.

IV. EVALUATION

In this section, we will introduce the data set used for the experiments and the anomaly detection model LSTM-AE constructed using time-series, Features, domain features , and finally compare LSTM-AE with other machine learning algorithms.

Networklayer	Hyperparameters	ParametersValue
LSTMLayer	OutputUnits	15,30,60,90,120
FullConnectLayer	HiddenLayerUnits	10,20,30,40,50
-	DropoutRatio	0.1,0.2,0.3,0.4,0.5
-	LearningRate	1e-4,1e-5,1e-6,1e-7
-	Optimizer	Adam,SGD,Adadelta,RMSprop,Nadelta,Nadam,Radam

TABLE II HYPER PARAMETERS OF LSTM-AE

A. Dataset

Our traffic data is collected from Shanghai Jiao Tong University DNS server data, including the response information of the client of one month. We evaluate LSTM-AE in real world environment using the real time traffic data. The real-time DNS traffic contains 1,889,224,056 DNS queries, which is a sum of an average of 60,942,711 queries per day. We also need the data set including benign samples and malicious samples for training. Benign samples are obtained from filtered selection of real-time DNS traffic, and malicious samples are obtained by building DNS traffic generated by malware.

1) Benign samples. We use Alexa top10,000 domain list to filter normal DNS behavior. Among all queries, about30%are selected as an absolute benign query. These data will be used as white list in later model training. This part of the data will be referred to as ART-Dataset (Alexa in Real Time).
2) Malicious samples. We choose some common DNS tunnel software and some malicious software samples of DNS data theft to Simulate the attacker to establish communication and use the packet capture tool to record the traffic.
3) In DNS that were once applied for data leakage. The software is included in Table III. Among those software , we chose two DNS data leakage malware, Framework POS [12] and Wekby [13], and two DNS tunnel software, Ozyman DNS[14] and Heyoka [15].Using those four software applications, we generated malicious traffic for model training. This part of the malicious sample will be referred to as WHOF-Dataset (Wekby+Heyoka+OzymanDNS+FrameworkPOS).

After processing, ART-Dataset has 20,179,032 records, containing Alexa traffic of one day. WHOF-Dataset has 1,123records, containing traffic generated by four different DNS communication software. It is about 0.01% of benign traffic, which is a appropriate rate of real world.

B. Metric

The data of the experiment consists of simulated malicious traffic and legal traffic. Accuracy is defined as the number of actual malicious domain names / the total number of detected domain names in the detection results, and the Recall rate is the number of correct domain names / the total number of detected domain names. Since the total number of malicious samples cannot be known in the real environment, the real-time Precision is defined as the number of verified malicious domain names / the total number of detected malicious domains.

Name	**Type**	**Year**
OzymanDNS	Tunnel	2004
Iodine	Tunnel	2006
TCP-over-DNS	Tunnel	2008
Dns2tcp	Tunnel	2009
Morto	Leakage	2011
YourFreedom	Leakage	2015
FrameworkPOS	Leakage	2016
Wekby	Leakage	2016
BernhardPOS	Leakage	2015
DNSMessenger	Leakage	2017
APT41-Speculoos	Leakage	2020

TABLE III DNS DATA LEAKAGE MALWARE AND DNS TUNNEL SOFTWARE

C. Parameters and Network Structure of LSTM-AE

The performance of the deep learning model depends on the predetermined super parameters, which are obtained by the optimization process. Hyper parameters are the external configuration of the model, and their values cannot be estimated from dataset. Unlike super parameters, model parameters are learned by optimizing functions to minimize the objective (or loss) function, but in the process of model training, hyper parameters are not needed. There are many hyper parameters in the deep neural network model. The model proposed in this paper optimizes five super parameters, as shown in Table II.

We first train an LSTM-AE model to automatically extract features on domain name resolution request time-series data. We choose the time window length wl=15min, and that will make 60/15*24=96 dimensions of input. The second layer has a third neuron of the input layer, which is 32 neurons. And finally, the encoded layer has 16 neurons. As for the activation function, we chose tan h between the first layer and the second layer. Relu function is applied between second layer and third layer. Loss function is a mean squared error. All records are clustered by their IP and private suffix, and are indexed in the order they were queried in traffic. The model was trained for50 epochs, with batch size set 32. Learning rate is 1e-4 and optimizer is a dam. After training loss does not decline, we apply the final model to whole dataset and illustrate the MAE and MSE of 5 classes of traffic.

Next, we combine the time-series features with the traditional domain name morphology and parsing features to build an anomaly detection model again. The input of the model includes time-series features, domain features and the output of the model is used as the standard of anomaly detection. The input of the second part is 38 dimensional features, and we finally adopt a two-layer auto encoder model.Thefirstlayercontains19 neurons, and the hidden layerconsistsof10 neurons. In the selection of activation function, tan h is selected between the first layer and the second layer, and relu is applied between the second layer and the third layer. MSE is chosen as the loss function.

D. Evaluating the Performance of LSTM-AE

We first validate the performance of the LSTM-AE modelusing only time-series features as input. Here, we combine ART (begine samples) and WHOF(malicious samples) as experimental dataset.

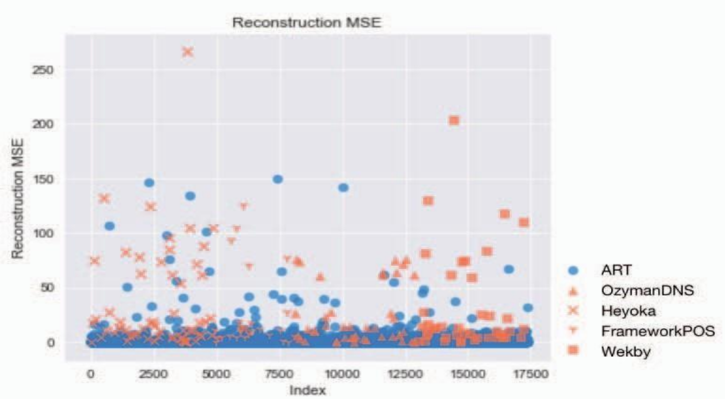
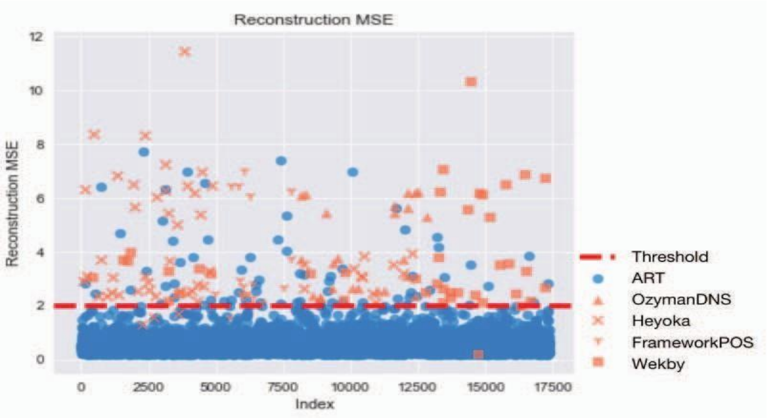


Fig. 4. With time-series, LSTM-AE reconstruction error of 5 Classes of DNS traffic, which shows 80% of malicious domains (in orange) have a reconstruction error greater than 5,while only 1% of normal traffic has a reconstruction error of more than 5. Use MSE as an indicator to distinguish between legitimate and malicious domains and get an AUC indicator of 0.73.

Experimental results are shown in Fig. 4. 80% of malicious domains (in orange) have a reconstruction error greater than 5, while only 1%of normal traffic has a reconstruction error of more than 5.UseMSEasanindicatortodistinguishbetween legitimate and malicious domains and get an AUC indicator of 0.73. By using time-series feature generated by the auto encoder model with loss function, we can separate most of the data leakage from benign traffic. However, only using time-series features is not enough. So we combines time-series features with the traditional domain features. The same is to use benign legal domain dataset for training, and after



knowing that the mean square error no longer decreases, the model is applied to the entire dataset. Calculate the anomaly score of each IP's request cluster for the domain. All the scores are shown in Fig.5. Finally, we choose MSE =2 as the threshold to distinguish benign and malicious domain names, which can distinguish more than99% of legitimate domain names from covert communication channel domain names.

Fig. 5. With time-series features and domain features, LSTM-AE reconstruction error of 5 Classes of DNS traffic with threshold, which can distinguish more than 99% of legitimate domain names from covert communication channel domain names.

E. Comparison between LSTM-AE and other ML methods

In this section, we use the decision tree algorithm in reference [49], the isolation forest algorithm in reference [50], and the original auto encoder, the variation auto encoder and the LSTM-AE proposed in this paper. The results are shown in Table IV.
Through the comparison of different models, we can find that the LSTM-AE algorithm has the best performance with94.13% accuracy, 99.38% recall ratio, and 91.43% real-time precision.

LSTM-AE detected three malicious domain names in the real-world environment. After manual verification, these domain names belong to the covert channel using DNS for covert data transmission, and multiple sub domain requests are used to encode and transmit information. However, the detection results of these domains by Virus total are all below 2. It shows that the LSTM-AE model has a good detection effect on the new covert communication channel.

In general, these results show that our method is significantly better than other models in DNS tunnel detection in a complex real-world environment. With the semi supervised learning mode, our model can automatically extract the time-series features of domain name resolution request sequence. We don't need to learn the characteristics of malicious samples and legitimate domain name samples to complete the classification task. Our model can obtain feature vectors from a large number of normal DNS traffic and extract the similarity between Feature vectors.

Model	SupervisedLearning	TrainingData	Features	Accuracy	Recall	real-timePrecision
DecisionTree[19]	True	ART+WHOF	Lexical+Resolve	0.8921	0.9923	0.7129
IsolationForest[20]	False	ART+WHOF	Lexical+Resolve	0.9144	0.9838	0.8336
AutoEncoder	False	ART	Lexical+Resolve	0.9325	0.9902	0.8942
Variational-AE	False	ART	Lexical+Resolve	0.9223	0.9854	0.8523
LSTM-AE	False	ART	Lexical+Resolve+TimeSeries	0.9413	0.9938	0.9143

TABLE IV DETECTION PERFORMANCE IN TRAINING DATA SET AND REAL TIME TRAFFIC

F. Analysis of False Detection

We analyzed the misclassification cases in domain name detection and found some misuse of DNS protocols for data exchange. These misclassified domain names are mainly some software services.

kr0.io is one of the domains whose DNS queries are answered by the name server NS1. ipass.com website. This name server belongs to the iPass group, a US-based mobile connectivity company that provides Wi-Fi-as-a-Service solutions. It is a legitimate and normal domain name, but there is suspicion of malicious operation, as it is not a secure company, but their service seems to make frequent DNS queries to their servers in the following format "c3auaaemnpkwuqaizgirxj4ma4rf7qervhn7ejpc rc "fhrdzbfos-rsqblywkova.kr0.io". Therefore, these queries detected by our proposed method have an average length of more than 60 characters for exceptions due to high entropy. The name servers of group infra.comarefromlog-ica.com. The Logica.com website belongs to a British company and was acquired by CGI. The full request for this domain is "uldap._tcp.052bfd48-d82f-48e7b789-cf90b86a25.groupinfra.com" and this request type is SRV. Our detection model is based on anomalous average request lengths, high entropy values and SRV records (which are usually used by tunneling software such as Iodine, DNS2tcp, etc.) and determines it to be a malicious communication channel. However, its actual controller is legitimate, so this behavior cannot be judged as malicious.

V. CONCLUSION

Using DNS for data transmission has a long history, but in recent years, attackers have adopted a more covert way of disclosure. This way of using a small number of DNS requests or responses for communication can bypass the traditional defense approaches, and it is very hard to detect because of the low volume of traffic. In this paper, we propose an anomaly detection algorithm based on domain query and time-series features. The method can detect hidden data leakage accurately. After the actual traffic deployment of Shanghai Jiao Tong University campus network, the data leakage events which are not known before are detected, and the model effect is verified.

ACKNOWLEDGEMENTS

This work is supported by National Key Research and Development Program of China [No. 2020YFB1807500] and National Natural Science Foundation of China [No.61831007].

REFERENCES

[1] Alexa top sites.http://www.alexa.com/topsites.

[2] Zander S, Armitage G, Branch P, A survey of covert channels and counter measures in computer network protocols, IEEE Communications Surveys & Tutorials, vol.9, no.3, pp.4457, 2007.

[3] Homem I, Papapetrou P, Dosis S, Entropy-based prediction of network protocols in the forensic analysis of dns tunnels. IEEE conference proceedings, 2016.

[4] Tatang D , Quinkert F , Dolecki N , et al. A Study of Newly Observed Host names and DNS Tunneling in the Wild[J].2019.

[5] Patsakis C, CasinoF, KatosV. Encrypted and covert DNS queries for botnets: Challenges and countermeasures[J]. Computers & Security,2020,88:101614.

[6] P. Rascagneres, New Framework POS variant exfiltrates data via DNSrequests,2016,[online] Available: https://bit.ly/2VpfVd5.

[7] Anagnostopoulos M,JohnAndréSeem. Another Step in the Ladder of DNS-Based Covert Channels: Hiding Ill-Disposed Information in DNSKEYRRs[J]. Information(Switzerland),2019,10(9):284.

[8] Wang Z, Dong H, ChiY,etal. DGA and DNS Covert Channel Detection System based on Machine Learning[C]// Proceedings of the 3rd International Conference on Computer Science and Application Engineering.2019:1-5.

[9] Sivaguru R , Peck J , Olumofin F , et al. Inline Detection of DGA Domains Using Side Information [J]. IEEE Access, 2020, 8:141910-141922.

[10] Mc Carthy S M, Sinha A, Tambe M, et al, Data exltration detection and prevention: Virtually distributed pomdps for practically safer networks, in International Conference on Decision and Game Theory for Security.Springer,2016,pp.3961.

[11] Schölkopf B, Williamson R C, Smola A J, et al, Support vector method for novelty detection, in Advances in neural information processing systems, 2000, pp.582588.

[12] Cambiaso E, Aiello M, Mongelli M, Feature transformation and mutual information for dns tunneling analysis, in Ubiquitous and Future Net-works (ICUFN), 2016 Eighth International Conference on. IEEE, 2016,pp.957959.

[13] Engelstad P, Feng B, van Do T, Detection of dns tunneling in mobile networks using machine learning, in International Conference on Information Science and Applications. Springer,2017, pp.221230.

[14] Mc Carthy S M, Sinha A, Tambe M, et al, Data exltration detection and prevention: Virtually distributed pomdps for practically safer networks, in International Conference on Decision and Game Theory for Security.Springer,2016,pp.3961.

[15] Cambiaso E, Aiello M, Mongelli M, Papaleo G. Feature transformation and mutual information for DNS tunneling analysis. In: Proceedings of the eighth international conference on ubiquitous and future networks(ICUFN).IEEE;2016.p.9579.

[16] Bilge L,Kirda E, Kruegel C,et al. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis[C]//Ndss.2011:1-17.

[17] Bilge L, Sen S, Balzarotti D, et al. Exposure: A passive dns analysis service to detect and report malicious domains[J].ACM.

[18] Transactions on Information and System Security (TISSEC), 2014,16(4): 1-28. Messabi K A, Aldwairi M, Yousif A A, et al. Malware detection using dns records and domain name features[C]// Proceedingsofthe2ndInternationalConferenceonFutureNetworksandDistributedSystems. 2018:1-7.

[19] Isolation forest. In: Proceedings of the eighth IEEE international conference on data mining,ICDM08.IEEE;2008.p.41322.

[20] Homem I, Papapetrou P,Dosis S. Entropy-based prediction of network protocols in the forensic analysis of dns tunnels. IEEE conference proceedings, 2016.