Detecting Ransomware in Encrypted Web Traffic

Jaimin Modi[1], Issa Traore[1(✉)], Asem Ghaleb[1], Karim Ganame[2], and Sherif Ahmed[3]

[1] ECE Department, University of Victoria, Victoria, BC, Canada
jaiminmodi@uvic.ca, itraore@ece.uvic.ca, aalmekhlafy@gmail.com
[2] StreamScan, 2300 Rue Sherbrooke E, Montreal, QC, Canada
ganame@streamscan.io
[3] Computer Science Department, University of Windsor, Victoria, Canada
Sherif.SaadAhmed@uwindsor.ca

Abstract. To date, only a small amount of research has focused on detecting ransomware at the network level, and none of the published proposals have addressed the challenges raised by the fact that an increasing number of ransomware are using encrypted channels for communication with the command and control (C&C) server, mainly, over the HTTPS protocol. Despite the limited amount of ransomware-specific data available in network traffic, network-level detection represents a valuable extension of system-level detection as this could provide early indication of ransomware activities and allow disrupting such activities before serious damage can take place. To address the aforementioned gap, we propose, in the current paper, a new approach for detecting ransomware in encrypted network traffic that leverages network connections, certificate information and machine learning. We leverage an existing feature model developed for general malware and develop a robust network flow behavior analysis model using machine learning that separates effectively ransomware traffic from normal traffic. We study three different classifiers: random forest, SVM and logistic regression. Experimental evaluation on a diversified dataset yields a detection rate of 99.9% and a false positive rate of 0% for random forest, the best performing of the three classifiers.

Keywords: Ransomware detection · Encrypted web traffic · Machine learning · Network traffic

1 Introduction

Maintaining adequate communication channel between an infected device and the command and control server (C&C) is an essential characteristic of modern malware. Increasingly, such communications are taking place using web protocols, such as HTTPS. According to a report by Cyren, a leading enterprise cybersecurity service provider, 37% of malware now utilize HTTPS [4]. The same report also stated that every major ransomware family since 2016 has used HTTPS communication [4]. Although several approaches have been published in the literature for ransomware detection, the current

A. Benzekri et al. (Eds.): FPS 2019, LNCS 12056, pp. 345–353, 2020.
https://doi.org/10.1007/978-3-030-45371-8_22

approaches overwhelmingly focus on analyzing system data available on the endpoint with limited or no consideration for the C&C communications. Network level detection is made more complicated because, as aforementioned, malware communications are increasingly carried over HTTPS encrypted channels. Only a limited amount of research have been published on detecting malware in encrypted traffic [5, 7, 9]. To the best of our knowledge, all the existing works have focused on general malware, while none of them has addressed the challenges pertaining to ransomware specifically.

We propose, in the current paper, a new model for detecting ransomware from encrypted network traffic. We use a feature model introduced in a previous work on general malware detection from encrypted traffic by Strasak [9] and develop a robust network flow behavior analysis model using machine learning. The approach consists of generating initially log files from network captures using the BRO intrusion detection system (IDS). We construct the flows by aggregating network packets from the log files and calculate the feature values. We utilize a wide variety of ransomware families and normal traffic data for constructing a diversified dataset to evaluate our approach. Experimental evaluation results are presented for balanced and imbalanced datasets. The remainder of the paper is structured as follows. Section 2 summarizes and discusses related works. Section 3 presents our datasets. Section 4 describes our feature model. Section 5 presents the experimental evaluation of our proposed approach. Section 6 makes some concluding remarks.

2 Related Work

Ransomware detection has been studied extensively in recent years. The existing works can be divided into three major categories: static, dynamic and hybrid. Dynamic approaches can be divided into system level and network level approaches. The majority of the dynamic approaches work at the system level. Examples of such proposals include ShieldFS by Continella et al. [3] and ELDERAN by Sgandurra et al. [8]. A much smaller number of proposals have been published on network-based dynamic analysis, including works by Cabaj et al. [2] and Almashhadani et al. [1]. Most of the current work on network-based dynamic analysis were based only on a small number of ransomware families. For instance, the experiment in [2] was based only on 2 ransomware families, while in [1] only a single ransomware family (i.e. locky) was considered.

3 Evaluation Datasets

To the best of our knowledge, there is no publicly available ransomware detection dataset. It has been decided at the ISOT lab to fill this gap by collecting a new dataset to be shared with the research community. The collected dataset includes both network and file activities generated from ransomware as well as benign applications. The binaries collected in the ISOT dataset consist of 666 different samples from 20 different ransomware families. Table 1 provides a detailed breakdown of the ransomware samples from each family.

Table 1. Ransomware dataset breakdown

Family	Samp	%	Family	Samp	%
TeslaCrypt	348	52.3	Mole	4	0.597
Cerber	122	18.236	Satan	2	0.299
Locky	129	19.28	CTBLocker	2	0.299
CryptoShield	4	0.597	Win32.Blocker	18	2.69
Unlock26	3	0.448	Spora	5	0.747
WannaCry	1	0.149	Jaff	3	0.448
CryptoMix	2	0.299	Zeta	2	0.149
Sage	5	0.747	Striked	1	0.149
Petya	2	0.299	GlobeImposter	4	0.598
Crysis	8	1.196	Xorist	2	0.299
Flawed	1	0.149			
Total	**666**				

The normal (i.e., benign) dataset was collected in 2 different ways - manual data collection and network files from the Stratosphere project [10]. The manual data collection was done while keeping Wireshark on capture mode and browsing through the websites on Alexa top 100 list. The duration for each capture was 30 min and 30 total samples were collected. More details on the dataset can be found in [6].

4 Feature Model

In this section, we present our feature model, and explain the creation of our flow dataset. More details on feature definitions and extraction can found in [6].

4.1 Creating Flows

After collecting the log files through Bro IDS, the next step is to preprocess the data and calculate the feature values. Before calculating the feature values, it is important to build the flows from the log files. The three major log files that are used for building flows in the current work are – ssl.log, conn.log and x509.log. Since the traffic is encrypted, we start by building our flow from the ssl.log file. An overview of the connection information provided by the ssl.log, conn.log and x509.log files is provided in Fig. 1.

The important point while building the flows is to connect all the log files. As it can be seen from Fig. 1, ssl.log and conn.log files can be connected through the Unique ID of connection (UID) column. Similarly, the ssl.log and x509.log files can be connected using the certificate identifier (cert_id) column from ssl.log file. The 'cert_chain_uid' column in the ssl.log file is a vector of strings such that each string value is present in the 'id' column of x509.log file. If the 'cert_id' column is empty, then the corresponding

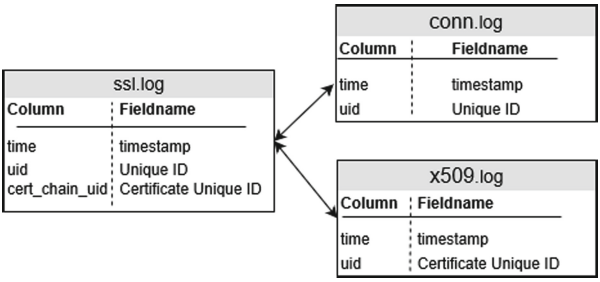

Fig. 1. Bro log files connection information

packet does not contain any certificate related information. Ultimately for a given flow, three different types of values will be added to it as the key: ssl values, connection values and certificate values. The basic algorithm for building a flow can be found in [6].

4.2 Feature Definitions

Our feature model which is based on the work of Strasák [9] consists of 3 groups of features carved out from each of the log files. In total, the model consists of 28 features as follows:

Conn.log features

- Number of flows
- Average duration of flows
- Standard deviation of flows duration
- Percent of standard deviation of flows
- Originator payload size
- Responder payload size
- Ratio of responder to originator payload size
- Percentage of established connections
- Inbound packets
- Outbound packets
- Periodicity average
- Periodicity standard deviation

Ssl.log features

- SSL flow ratio
- SSL-TLS ratio
- SNI-SSL ratio
- SNI as IP
- Certificate path average
- Self-signed certificate ratio

X509.log features

- Public key average
- Average certificate validity
- Standard deviation of certificate validity
- Certificate validity during capture
- Average certificate age
- Number of certificates
- Average number of domains in SAN
- Certificate-ssl logs ratio
- SNI in SAN DNS
- CN in SAN DNS

5 Classification and Evaluation Experiments

In the current paper, we use and compare three different classification algorithms, including logistic regression (LR), random forest (RF) and support vector machine (SVM). We use the current dataset as it is and a balanced dataset analysis using the smote oversampling technique. Table 2 shows the accuracy results for all classifiers with and without smote for the '*train_test_split*' set (80% training and 20% testing). Similarly, Table 3 shows the accuracy for the 10-fold cross validation. We present and discuss the performance results for each of the separate classifiers in the following.

Table 2. Accuracy score for the classifiers for 'train_test_split' set

Classifier	Accuracy (%)	
	Without SMOTE	With SMOTE
Logistic regression	94	94
Random forest	100	100
Support vector machine	95	94

Table 3. Accuracy score for the classifiers for 10-fold cross validation set

Classifier	(Average) Accuracy (%)	
	Without SMOTE	With SMOTE
Logistic regression	91	96
Random forest	98	100
Support vector machine	91	96

Table 4. Classification report for the logistic regression classifier

Classification report				
Metrics	SMOTE		Without SMOTE	
	Normal	Ransomware	Normal	Ransomware
Precision (%)	100	72	97	79
Recall (%)	93	100	96	83
f1 - score (%)	96	84	97	81
Support	2640	465	2640	465

Logistics Regression: Table 2 shows that smote has no effect on accuracy for '*train_test_split'* method. Table 4 shows the classification report for the logistic regression classifier.

Since applying smote on the dataset gives better output, we perform hyper-parameter tuning only for that dataset. The following model parameters were utilized for tuning:

a. *C* (Inverse of regularization strength) is varied from 0.0001 to 1000.
b. *Penalty* – We explore two possibilities in our work called L1 and L2 regularizations as follows: L1 regularization (also known as Lasso regression.) and L2 regularization (also called Ridge regression).

After doing a grid search through the parameters, the logistic regression gives best accuracy at 96% for the following parameter values: *C = 1* and *Penalty = L1*. This means that accuracy does not change, however, by looking at the confusion matrix shown in Fig. 2 it can be observed that there is a decrease in false positive as well as false negative rates.

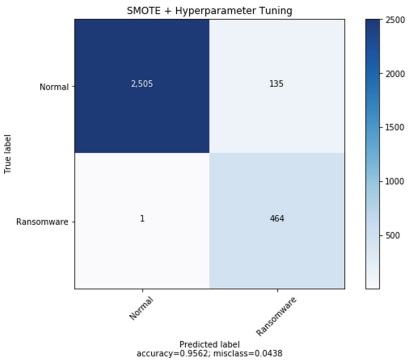

Fig. 2. Confusion matrix for the logistic regression classifier after applying hyper-parameter tuning

Random Forest: Random forest gives the best accuracy amongst all the three classifiers.

Table 5 show the classification results for the random forest classifier. The random forest classifier achieves an accuracy of 100% and a detection rate of almost 100% with an FPR of 0%. We obtain a FPR of 0% since we have no false positives. The only reason for applying hyper-parameter tuning is to check if this can reduce the false negatives to 0. That way we can achieve perfect detection rate. The parameters considered for tuning were as follows:

Table 5. Classification report for the random forest classifier

Classification report				
Metrics	SMOTE		Without SMOTE	
	Normal	Ransomware	Normal	Ransomware
Precision (%)	100	100	100	100
Recall (%)	1.00	1.00	1.00	99
f1 - score (%)	100	100	100	100
Support	2640	465	2640	465

a. *n_estimators* – the number of trees to be used in the classifier, varied from 1 to 200.
b. *max_features* – the number of features to be considered while splitting the treed, varied from 1 to 28.
c. *max_depth* – the depth of each tree in the classifier, varied from 1 to 32.

After conducting a grid search, the random forest classifier gives an accuracy of 100%. The number of false positives and false negatives also remain the same. Hence, there is no change in the detection rate.

SVM: Table 2 shows that SVM does not perform best in terms of cross validation accuracy. Again, smote does well in reducing the number of false negatives. Table 6 show the classification results for the SVM classifier.

Table 6. Classification report for the SVM classifier

Classification report				
Metrics	SMOTE		Without SMOTE	
	Normal	Ransomware	Normal	Ransomware
Precision (%)	97	81	1.00	73
Recall (%)	96	85	93	1.00
f1 - score (%)	97	83	97	84
Support	2640	465	2640	465

The parameters considered for hyper-parameter tuning is C – the area of hyperplane created by misclassified training samples around the hyperplane. We vary the value of C from 1 to 100. The SVM classifier achieves an accuracy of 96% with value of '$C = 15$'. Also, hyper-parameter tuning SVM improves the detection rate slightly.

6 Conclusion

In the current paper, we proposed a new system for detecting ransomware in encrypted web traffic. Our approach consists of extracting meaningful information from network connections and certificates, and utilizing machine learning for detecting ransomware packets in network data. We explored a feature space in three broad domains of network characteristics: connection based, encryption based, and certificate based. We implemented and studied 3 different classifiers for our model, namely, logistic regression, SVM and random forest. Our experimental evaluation yields for random forest, the best performing classifier, a detection rate of 99.9% and a false positive rate of 0%. The obtained results are very encouraging and an indicator of the feasibility of effective ransomware detection from encrypted network traffic.

Although detecting ransomware in network traffic data is our main focus, identifying the family to which the ransomware belong would be beneficial. Our future work will explore how to extend our model to detect specific ransomware families, in addition to detecting individual ransomware samples.

References

1. Almashhadani, A.O., Kaiiali, M., Sezer, S., O'Kane, P.: A multi-classifier network-based crypto ransomware detection system: a case study of locky ransomware. IEEE Access **7**, 47053–47067 (2019)
2. Cabaj, K., Gregorczyk, M., Mazurczyk, W.: Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics. Comput. Electr. Eng. **66**, 353–368 (2018)
3. Continella, A., et al.: ShieldFS: a self-healing, ransomware-aware filesystem. In: Proceedings of the 32nd Annual Conference on Computer Security Applications, pp. 336–347. ACM (2016)
4. Cyren - SSL Traffic Growth - Malware is Moving Heavily to HTTPS. https://www.cyren.com/blog/articles/over-one-third-of-malware-uses-https
5. Prasse, P., Machlica, L., Pevný, T., Havelka, J., Scheffer, T.: Malware detection by analysing encrypted network traffic with neural networks. In: Ceci, M., Hollmén, J., Todorovski, L., Vens, C., Džeroski, S. (eds.) ECML PKDD 2017. LNCS (LNAI), vol. 10535, pp. 73–88. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71246-8_5
6. Modi, J., Traore, I., Ghaleb, A., Ganame, K., Ahmed, S.S.: Detecting ransomware in encrypted web traffic. Technical report ECE-2019-10-1, University of Victoria, ECE Department, 5 October 2019. http://www.uvic.ca/engineering/ece/isot/publications/index.php
7. Niu, W., Zhuo, Z., Zhang, X., Du, X., Yang, G., Guizani, M.: A heuristic statistical testing based approach for encrypted network traffic identification. IEEE Trans. Veh. Technol. **68**(4), 3843–3853 (2019)

8. Sgandurra, D., Muñoz-González, L., Mohsen, R., Lupu, E.C.: Automated dynamic analysis of ransomware: benefits, limitations and use for detection, arXiv preprint arXiv:1609.03020 (2016)
9. Strasák, F.: Detection of HTTPS malware traffic. Bachelor's thesis, Czech Technical University in Prague, May 2017
10. Normal Captures—Stratosphere IPS. https://www.stratosphereips.org/datasets-normal. Accessed 19 July 2019