VFAST Transactions on Mathematics
http://vfast.org/journals/index.php/VTM@ 2018, ISSN(e): 2309-0022, ISSN(p): 2411-6343
Volume 6, Number 1, January-December, 2018 *pp: 15-22*

MITIGATION OF THE EFFECT OF STANDARD NETWORKS ATTACKS IN SSL ENCRYPTED TRAFFIC BY ENCRYPTED TRAFFIC ANALYSIS

Muhammad Hamad[1], Dr. Hanif Durad[1] and Muhammad Yousaf[2]
[1.]Department of Computer and Information Sciences
Pakistan Institute of Engineering and Applied Sciences
Islamabad, Pakistan
[2]CESAT, Islamabad, Pakistan
[1]mhamad758@gmail.com; [2]hanif@pieas.edu.pk and y.baig@outlook.com

Abstract --With increased use of encryption, cyber threat landscape is changing. For general public this transition shifts to more private and safer internet experiences, but at the same time it is a serious concern for security personnel now. For them it hinders control over the traffic moving on their network and poses difficulty in traffic analysis and management. Security personals are interested in knowing how the network is being accessed, whether or not that traffic contains some malware and is safe enough and compliant with your organization's policies. This project is not about decrypting the encrypted content of packet's payload as it will highly degrade network performance plus some advance encryption algorithms like AES are assumed to be perfect. So aim of this project is to analyze encrypted traffic and find out some interesting patterns without need of bulk decryption. The analysis will be based on flow based features and metadata. Encrypted Traffic Analytics maintains the integrity of the encrypted flow and doesn't affect privacy of users.

Key Words:

1. Introduction. To ensure secure user experience over the internet, encryption plays vital role. Encryption mechanism provides confidentiality and integrity to the network traffic with increased level of trust among communicating parties. Network visibility is now an important concern for security personals due to increased use of encryption. To know 'what's going on?' inside your network is the first step of securing it. In the beginning of securing your network from adversaries, IP and port based rules were used in firewalls. But with increased used of proxy servers, VPN's and with many other options IP address can be spoofed and most of the applications use non-standard ports as an obfuscation measure. So to rely only on firewalls as protection measure is not a good option to try. The next step that can be taken in security of cyber premises is use of IDS and IPS. These detection and prevention systems contains large number of threat signatures, anomaly based techniques and reputation based techniques to identify malicious patterns and threats in the traffic. But as soon as traffic get encrypted it poses serious challenge to these detection systems. This helps malware traffic to easily evade from detection even if detection systems are online. So from all the discussion above it is clear that we need some mechanism to analyze encrypted traffic.

In this paper, we have presented an approach to analyze encrypted traffic to ensure that traffic flowing over the network is not malicious without affecting user's privacy. For classifying the encrypted SSL traffic correctly supervised machine learning models are employed over dataset containing both normal and malware traffic inside it. Machine learning algorithms that are used in this work are XGBoost, Random Forest, SVM, K-nearest neighbours and Decision trees. From the experiments it has been observed that XGBoost has the highest prediction accuracy. The organization of this paper is as follows. Section II presents the overview of related work in this field Section III. describes different techniques of classifying encrypted traffic. In Section IV data collection will be explained, Section V will be based on feature extraction and proposed methodology of analysis. Section VI presents the evaluation of test results of machine learning algorithms. Finally, we conclude in Section VII of the paper.

2. Related Work. To the best of our knowledge, there is very little work done in the field of encrypted traffic analytics. At the time of writing this document there exists Cisco's Encrypted Traffic Analytics (ETA) framework, a software platform that monitors network packet metadata to detect malicious traffic, even if it's encrypted. Encrypted Traffic Analytics is a product deployed on customer's premises that monitors their network and collects information about traffic flows. It uses a series of sensors placed throughout the network to screen all traffic traversing through it. ETA[1] uses a combination of local analysis engines combined with a cloud-based platform that analyzes anonymized metadata about network traffic to search for and block malicious traffic, even if it's encrypted. ETA collects metadata about traffic flows using a modified version of NetFlow[2][3][4] and searches for characteristics that indicate the traffic could be malicious. It inspects the initial data packet, which is translated in the clear, even in encrypted traffic. It also records the size, shape and sequence of packets, how long they take to traverse the network, and it monitors for other suspicious characteristics such as a self-signed certificate[5], or whether it has command-and-control identifiers on it. All of this data can be collected on traffic, even if it is encrypted. "ETA uses network visibility and multi-layer machine learning to look for observable differences between benign and malware traffic. ETA's monitoring system is named Stealth Watch and the cloud-based data store is named as Talos[6]. [7]

Some contributions have been made by academia and networks researchers in field of ETA. Although contents of cryptographically secure session is not possible but profiling of secure apps can give interesting results.[8] First capture the traffic using Wireshark[9]–[11] or any packet capturing software, then identify the packets related to apps you want to profile because lot of packets are generated by mobile devices. One can generally trigger repeated events of messaging, voice and video calls or file sharing to analyze the traffic captures. As an example, they took viber, and observed Viber client established the connection to its server on destination UDP port of 7985. The next step is to determine server ranges. Using the firewall, subnets of server ranges can be blocked to force the target app to try connections on all available options embedded in its design. During the study of their traffic dumps, IPs of both the calling and caller parties can be identified. Use firewall to determine all possible ports being used by an app. By inspecting packets, one can generally find out some interesting stuff: (i) Inspection of byte patterns is carried out by repeating the forced events in a firewalled controlled scenario. (ii) Frequency of bytes exchanged for different events. (iii) Acknowledgments and responses between server and client. (iv) Payload sizes for different services and their uniqueness with respect to different events. This type of profiling needs human involvement and they are not actually detecting malware from traffic instead only profiling the things like communication ports and servers to whom application communicates.

In [12] they detected malware on client computers present in HTTPS traffic and trained classifier based on LSTM[13].

Another work related to encrypted traffic analysis with respect to machine learning was explained in [14]. In this paper five machine learning algorithms are employed (AdaBoost[15], Support Vector Machine[16], Naïve Bayesian[17], RIPPER and C4.5[18]) to train classifier based on flow based data instead of just IP, port and payload. Secured Shell (SSH) traffic and skype is taken as example after that results of these algorithms were compared.

In [19], they develop supervised machine learning models that take advantage of TLS[20], [21] handshake metadata, DNS contextual flows linked to the encrypted flow and the HTTP headers of HTTP contextual flows from the same source IP address within a 5 minute window.

DNS response provides the address used by an encrypted flow, and the TTL associated with the name. Having the domain name for an IP address provides a lot of meaningful information on its own. Among TLS flows, this information can sometimes be gathered from the server name indication extension or the subject of the server certificate. In these cases, the contextual DNS flow has the potential to provide information which would otherwise be unavailable.

In Prior work on encrypted traffic analysis discussed above their prime focus was to find the communication patterns of certain applications. In contrast to this we are focused on detection of malwares in network traffic. The detection mechanism is based on machine learning and independent of application used.

3. METHODOLOGY

There are some ways that can help in analyzing encrypted traffic

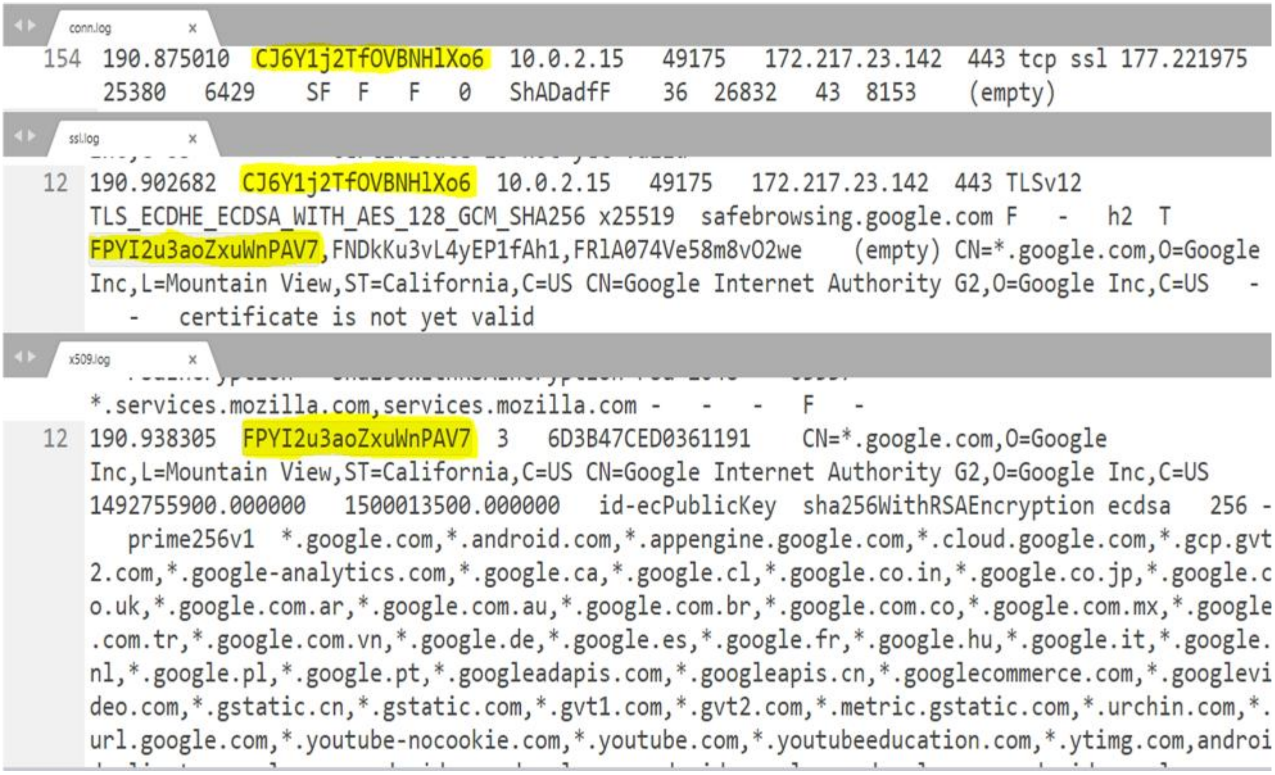
1) 1-Decrypt the encrypted traffic and pass it to analysis engine bulk decryption, analysis and re encryption is not always practical or feasible, for performance and resource reasons.
2) 2-Instead of collecting all traffic from network collect only session data. With session data we can observe normal communication pattern for example how many bytes of data is communicated from some specified host. So if there is any ab-normality in bandwidth usage of some host the alerts can be generated even if the traffic is encrypted.

The methodology we used is based on intra-flow metadata. Intra-flow metadata, or information about events that occur inside of a flow, can be collected, stored and analyzed within a flow monitoring framework. This data is especially valuable when traffic is encrypted, because deep packet inspection is no longer viable. This analysis of intra-flow metadata, called Encrypted Traffic Analytics, is derived by using new types of features that are independent of protocol details, such as the lengths and arrival times of messages within a flow. These data elements have the attractive property of applying equally well to both encrypted and unencrypted flows.

3.1. DATA COLLECTION. The dataset we used is produced by Stratosphere Malware Capture Facility Project done by Maria Jose Erquiaga and contains enough samples of malware and normal traffic [22]. She made the dataset by using malwares that uses HTTPS. The dataset contains 355 malware captures and 30 Normal captures. The dataset contains pcap files as well as log files generated by BRO IDS. The log files that are used in our work are conn.log, ssl.log and x509.log.While working with raw pcap files it is very cumbersome task to interrelate the connection information with the SSL encryption information and X509 certificates. BRO links the information present in different log files using unique id. In Figure. 1, entries of conn.log, ssl.log and x509 log is shown, each entry in log file has unique id e.g. one conn.log entry has unique id CJ6Y1j2TfOVBNHlXo6 and it also has ssl.log entry with SSL information. Similarly, unique id of X509 certificate is present as field entry in SSL log entry.

Table 1 Distribution of Normal and Malware Samples in Dataset

Connection Records	***Training data***	***Testing data***	***Total data***
Normal	37103	9284	46387
Malware	6657	1656	8313
Total	43760	10940	54700



```
190.875010  CJ6Y1j2TfOVBNHlXo6  10.0.2.15   49175   172.217.23.142  443 tcp ssl 177.221975
25380   6429    SF  F   F   0   ShADadfF    36  26832   43  8153    (empty)
```

```
190.902682  CJ6Y1j2TfOVBNHlXo6  10.0.2.15   49175   172.217.23.142  443 TLSv12
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 x25519  safebrowsing.google.com F   -   h2  T
FPYI2u3aoZxuWnPAV7,FNDkKu3vL4yEP1fAh1,FRlA074Ve58m8vO2we    (empty) CN=*.google.com,O=Google
Inc,L=Mountain View,ST=California,C=US CN=Google Internet Authority G2,O=Google Inc,C=US   -
   -   certificate is not yet valid
```

```
*.services.mozilla.com,services.mozilla.com -   -   -   F   -
190.938305  FPYI2u3aoZxuWnPAV7  3   6D3B47CED0361191    CN=*.google.com,O=Google
Inc,L=Mountain View,ST=California,C=US CN=Google Internet Authority G2,O=Google Inc,C=US
1492755900.000000   1500013500.000000   id-ecPublicKey  sha256WithRSAEncryption ecdsa   256 -
   prime256v1  *.google.com,*.android.com,*.appengine.google.com,*.cloud.google.com,*.gcp.gvt
2.com,*.google-analytics.com,*.google.ca,*.google.cl,*.google.co.in,*.google.co.jp,*.google.c
o.uk,*.google.com.ar,*.google.com.au,*.google.com.br,*.google.com.co,*.google.com.mx,*.google
.com.tr,*.google.com.vn,*.google.de,*.google.es,*.google.fr,*.google.hu,*.google.it,*.google.
nl,*.google.pl,*.google.pt,*.googleadapis.com,*.googleapis.cn,*.googlecommerce.com,*.googlevi
deo.com,*.gstatic.cn,*.gstatic.com,*.gvt1.com,*.gvt2.com,*.metric.gstatic.com,*.urchin.com,*.
url.google.com,*.youtube-nocookie.com,*.youtube.com,*.youtubeeducation.com,*.ytimg.com,androi
```

Figure 1. Interconnection of BRO logs

3.2. Features Extraction

A composite key is created to uniquely identify a flow. The key consists of source IP, destination IP, destination port, and protocol.

A- Triplet of Records. The triplet of record is actually composed of one connection record, one associated SSL record, and x509 record if the certificate exists for that connection instance. The unique id assigned by BRO IDS to log entries are same in ssl.log and conn.log entries and associated certificate id's can be found out by checking client_cert_chain_fuids field value present in ssl.log. This certificate id will have associated certificate entry in x509.log file. Thus collectively they make triplet of record. Instead of using conn.log file in the process of feature extraction all the connection entries are labeled as normal/malware or background by using list of normal and malware ips. As in process of creating this dataset normal and malware ips was known so labelling is easy. Now let's see how we made a triplet of record.

1) *First of all, pick a unique id from SSL entry in ssl.log and use this id to find connection record in conn_label.log, this connection entry will have normal/malware or background as a label. If label not is normal or malware, then read next SSL log entry because this entry will not help in training our machine learning model.*

2) *If the certificate path is not empty in SSL record, then take the first unique key and find the corresponding record in X509.log file. This will complete one triplet of record*

3) *Now read lines one by one in conn_label.log and check for each composite key {source IP, destination IP, destination port, and protocol} already triplet of record is created if not, then add these remaining records to connection triplets list.*

The connection log file give information like which ip's were involved, the protocol they used, number of bytes they communicated, and state of connection etc. As we know that in SSL encryption first packet is sent in clear so it is always of great interest, so SSL log file provide encryption info of the connection e.g. what is the version of TLS/SSL, which cipher suite is used, and SNI information etc. The X509 log file helps in proving credibility of connection. The certificate features are like when the certificates are valid from and when the certificate will expire, what was the signing algorithm of certificate, what is the length of public key etc. The feature we used in our work are described below

1) *Number of SSL flows and non-SSL flows*
2) *Average of connection duration*
3) *Standard deviation of connection duration*
4) *Number of bytes sent by originator*
5) *Number of bytes sent by the responder*
6) *Ratio of bytes sent by responder to bytes sent by originator*
7) *Ratio of total states in which connection was made to total possible values of states.*
8) *Total packets received*
9) *Total packets sent*
10) *Find standard deviation of periodicity of connections*
11) *Ratio of SSL flows to non-SSL flows*
12) *Ratio of TLS flows to total number of flows*
13) *Ratio of which SSL logs has certificate associated*
14) *How many SSL flows contains SNI?*
15) *When bro generates SSL.log file it also adds validation status field that shows either certificate is self-signed or not, so calculate the ratio of self-signed certificates to total SSL logs*
16) *Check if SNI is in certificate's SNA or not?*
17) *The average of public key*
18) *Average of validity length of certificates (in days)*
19) *Standard deviation of average of validity length of certificates*
20) *Percentage of validity completed of certificate*
21) *Number of domains in certificates*

3.3. Experimental Results and Analysis

Among total samples, 80% of the data is used as training data and 20% of the data is used as testing data. We have used 5 different machine learning classifiers and generated the confusion matrix using Scikit-learn[23] package in python. Based on the on confusion matrix different measures are calculated and shown in table 2.

Table 2 Comparisons of Different Machine Learning Algorithms

Measure	**Classifier**				
	XGBoost	***Random Forest***	***Decision Tree***	***SVM***	***KNN***
Accuracy	0.9854	0.9804	0.9730	0.9358	0.9545
Precision	0.9764	0.9681	0.9172	0.8559	0.8646
Specificity	0.9960	0.9947	0.9854	0.9792	0.9768
Sensitivity	0.9263	0.9003	0.9033	0.6926	0.8297
F1 Score	0.9507	0.9330	0.9102	0.7656	0.8468

Along with confusion matrix and different measures of accuracy ROC curve for all these classifiers is plotted and shown in figure. 2, figure. 3, figure. 4, figure. 5, figure. 6. The graphical representation of measures calculated in table 2 is plotted in figure 7 in the form of bar chart.

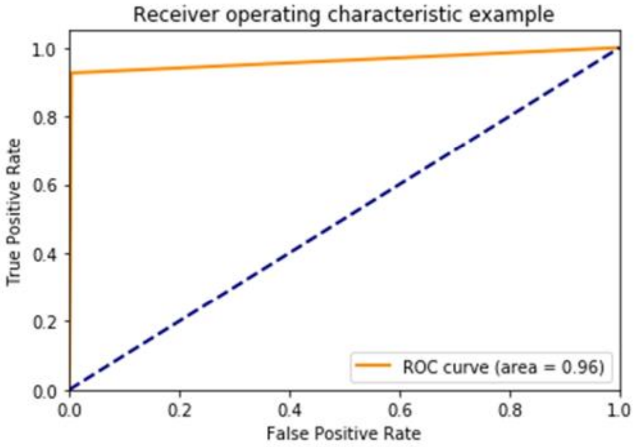


Figure 2. ROC curve for XGBoost

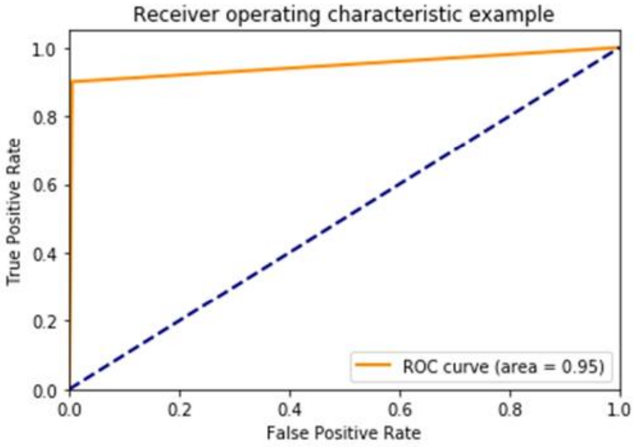


Figure 3. ROC curve for Random Forest

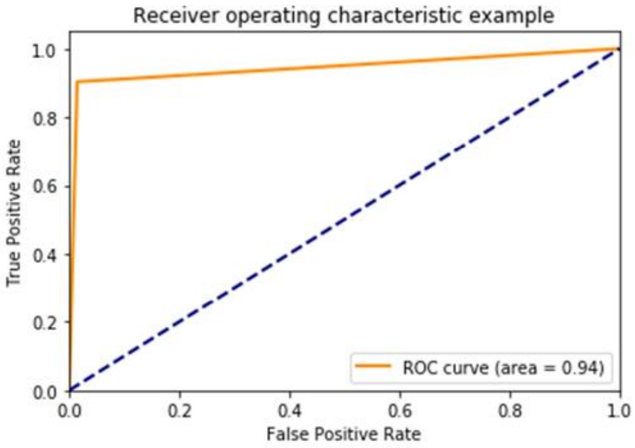


Figure 4. ROC curve for Decision tree

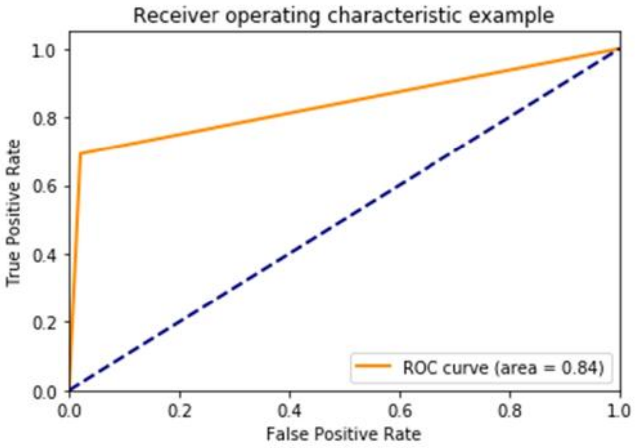


Figure 5. ROC curve for Support Vector Machine

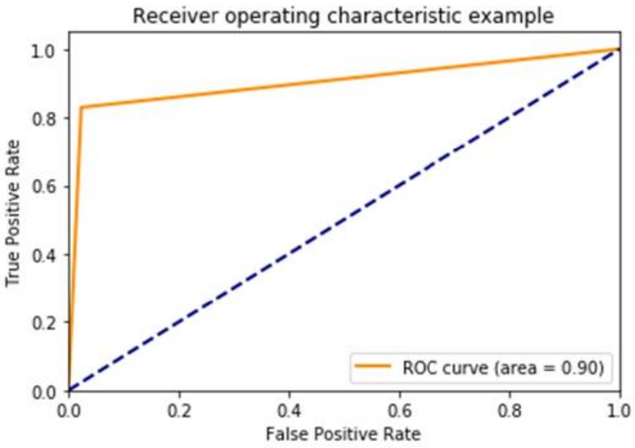


Figure 6. ROC curve for K-Nearest Neighbors

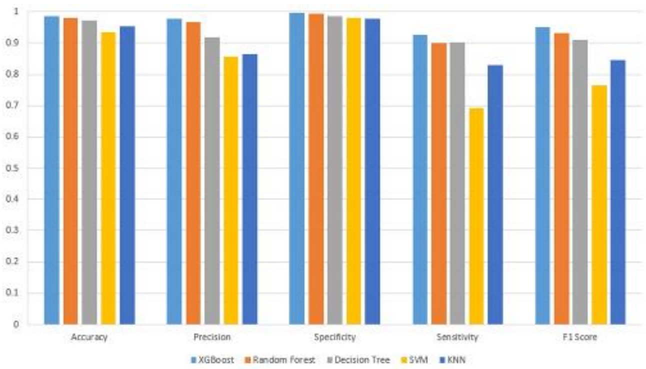


Figure 7. Comparative Analysis of different machine learning classifiers

4. CONCLUSION. In this paper we analyzed the encrypted traffic using intra-flow metadata. The training and testing data is prepared from log files generated by BRO IDS as it correlates connection information, encryption information, and X.509 certificate information in separate log files. We used 5 different machine learning algorithms i.e. XGBoost, Random Forest, SVM, Decision trees and K-Nearest Neighbours. From experiments it can be seen that XGBoost has highest classification accuracy of 98.5% among other classifiers. The method described in this paper can be used by security personnel to protect their organization from malwares that can be present in encrypted traffic flowing over the network

REFERENCES

[1] Jason Liu, “A Guide for Encrypted Traffic Analytics,” 2017. [Online]. Available: https://blogs.cisco.com/enterprise/a-guide-for-encrypted-traffic-analytics. [Accessed: 25-Sep-2018].

[2] Cisco, I. O. S. (2008). NetFlow.

[3] Claise, B., Sadasivan, G., Valluri, V., & Djernaes, M. (2004). Cisco systems netflow services export version 9..

[4] Sommer, R., & Feldmann, A. (2002, November). NetFlow: Information loss or win?. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment* (pp. 173-174)..

[5] “Hackborn, D. K., Bort, D. P., Onorato, J. M., Bornstein, D. R., McFadden, A. T., Swetland, B. J., & Cannings, R. G. (2013). *U.S. Patent No. 8,589,691*. Washington, DC: U.S. Patent and Trademark Office.

[6] Biasini, N., Esler, J., Herbert, N., Mercer, W., Olney, M., Taylor, M., & Williams, C. (2015). Threat spotlight: Cisco talos thwarts access to massive international exploit kit generating $60 m annually from ransomware alone. *Cisco Talos. Retrieved from http://www. talosintel. com/angler-exposed*.

[7] Szigeti, T., Zacks, D., Falkner, M., & Arena, S. (2018). *Cisco Digital Network Architecture: Intent-based Networking for the Enterprise*. Cisco Press..

[8] Sudozai, M. A. K., & Saleem, S. (2018, January). Profiling of secure chat and calling apps from encrypted traffic. In *2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)* (pp. 502-508). IEEE..

[9] Sanders, C. (2017). *Practical packet analysis: Using Wireshark to solve real-world network problems*. No Starch Press..

[10] Chapell, L. (2010). Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide, Protocol Analysis Institute. *EE. UU. Editorial Chapell University*.

[11] Orebaugh, A., Ramirez, G., & Beale, J. (2006). *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier.

[12] Prasse, P., Machlica, L., Pevný, T., Havelka, J., & Scheffer, T. (2017, May). Malware detection by analysing network traffic with neural networks. In *2017 IEEE Security and Privacy Workshops (SPW)* (pp. 205-210). IEEE.

[13] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search

space odyssey. *IEEE transactions on neural networks and learning systems*, *28*(10), 2222-2232.

[14] Alshammari, R., & Zincir-Heywood, A. N. (2009, July). Machine learning based encrypted traffic classification: Identifying ssh and skype. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1-8). IEEE.

[15] Hu, W., Hu, W., & Maybank, S. (2008). Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *38*(2), 577-583.

[16] Schölkopf, B., Smola, A. J., & Bach, F. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

[17] Murphy, K. P. (2006). Naive bayes classifiers. *University of British Columbia*, *18*, 60..

[18] Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

[19] Anderson, B., & McGrew, D. (2016, October). Identifying encrypted malware traffic with contextual flow data. In *Proceedings of the 2016 ACM workshop on artificial intelligence and security* (pp. 35-46).

[20] Morrissey, P., Smart, N. P., & Warinschi, B. (2008, December). A modular security analysis of the TLS handshake protocol. In *International Conference on the Theory and Application of Cryptology and Information Security* (pp. 55-73). Springer, Berlin, Heidelberg. [21] E. Rescorla, “Http over tls,” 2000.

[22] Garcia, S., Grill, M., Stiborek, J., & Zunino, A. (2014). An empirical comparison of botnet detection methods. *computers & security*, *45*, 100-123.

[23] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830..