International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Office Information Systems: The Challenges of Detecting Malware in Encrypted Network Traffics for Smooth Information Processing

Surajo Zubairu

Department of Office Technology and Management, Abdu Gusau Polytechnic, Talata Mafara, Zamfara State, Nigeria

ABSTRACT

In recent years there has been a dramatic increase in the number of malware attacks that use encrypted HTTP traffic for self-propagation and communication. Due to the volume of legitimate encrypted data, encrypted malicious traffic resembles legitimate traffic. As the malicious and legitimate traffic looks similar, it poses challenges for antivirus software and firewalls. Since antivirus software and firewalls will not typically have access to encryption keys, detection techniques are needed that do not require decrypting the traffics In this research, we will survey Intrusion Detection Systems (IDS) that can be deployed to work efficiently regardless of encryption and then attempt to apply machine learning technique to the problem of distinguishing malicious encrypted Hyper Text Transfer Protocol (HTTP) traffics from legitimate encrypted traffics without decryption. The effectiveness of machine-learning approaches crucially depends on the availability of large amounts of labelled training data. However, obtaining ground-truth class labels for Hyper Text Transfer protocol secured (HTTPS) traffic is a difficult problem when the Hyper Text Transfer protocol (HTTP) payload is encrypted, generally one cannot determine whether it originates from malware by analyzing the network traffic in isolation. We develop an approach to collecting training data based on a VPN client that is able to observe the associations between executable files and TCP/IP packets on a large number of client computers. One of the few observable features of Hyper Text Transfer protocol secured (HTTPS) traffic is the host IP address and, if a DNS entry exists for that address, the domain name. In order to extract features from the domain name, we explore neural language models (CISCO, 2017) which use neural networks to derive a low-dimensional, continuous state representations of a text. As a baseline, we also study manually-engineered domain features (A. Özgür., 2016).

INTRODUCTION

Malware also known as malicious software are computer programs designed to infiltrate a system and cause a deliberate damage to data/memory content without the consent of the user (K. Scarfone et al., 2007) This threat can be a virus, spyware, worm, trojan, adware and any other program that can alter data stored on physical or virtual drive or memory content. Malware also violates user’s privacy, harvest passwords used to commit fraud, and can steal and encrypt user’s files for ransom (J. Aycock., 2006). Malware can also be detected by analyzing network communications. TCP/IP traffic can be analyzed by network equipment without direct access to the client computer that is executing the malware. This approach allows the encapsulation of malware detection into specialized network devices and helps to protect an entire organization even if users of individual computers do not run antivirus software. Analysis of TCP/IP traffic may aim at finding specific types of malware (V. Chandola et al., 2009), (M. Grill., 2016), or at identifying malicious servers of malware on client computers (R. Sommer et al., 2010). The deployment of network-traffic analysis systems has triggered evasion strategies. An analysis of the HTTP payload of the network traffic can easily be prevented by using the encrypted HTTPS Hyper Text Transfer protocol secured (HTTPS). Currently, over 40% of the most popular websites support Hyper Text Transfer protocol secured (HTTPS) (S. Axelsson., 2000). Under the HTTPS (HTTP over TLS/SSL) protocol, the client computer establishes a TCP/IP connection to (usually) port 443 of the host. On the application layer, HTTPS uses the standard Hyper Text Transfer protocol (HTTP), but all messages are encrypted via the Transport Layer Security (TLS) protocol or its predecessor, the Secure Socket Layer (SSL) protocol. An observer can only see the client and host IP addresses and ports, and the timestamps and data volume of all packets. The Hyper Text Transfer protocol (HTTP) payload, including the header fields and URL, are encrypted. In this project, we will develop a machine-learning method that detects malware on client computers based on the observable information of Hyper Text Transfer protocol secured (HTTPS) communication. The effectiveness of machine-learning approaches crucially depends on the availability of large amounts of labelled training data. However, obtaining ground-truth class labels for Hyper Text Transfer protocol secured (HTTPS) traffic is a difficult problem when the Hyper Text Transfer protocol (HTTP) payload is encrypted, generally one cannot determine whether it originates from malware by analyzing the network traffic in isolation. We develop an approach to collecting training data based on a VPN client that is able to observe the associations between executable files and TCP/IP packets on a large number of client computers. One of the few observable features of Hyper Text Transfer protocol secured (HTTPS) traffic is the host IP address and, if a DNS entry exists for that address, the domain name. In order to extract features from the domain name, we explore neural language models (CISCO, 2017) which use neural networks to derive a low-dimensional, continuous state representations of a text. As a baseline, we also study manually-engineered domain features (A. Özgür., 2016).

PROBLEM STATEMENT:

Lately there has been an increase in the amount of communication traffic worldwide as the Internet keeps growing into rural communities. On the other-hand malware using HTTPS traffic for their communications is also on the rise (S. Yoon et al., 2009). This situation poses a challenge for the security analysts, as most of the current systems are inefficient and there is the need to discover new features and methods to detect malware. In this work, the proposed system will be highly efficient, cheaper, faster and private, respecting the original idea of HTTPS.

RESEARCH QUESTIONS:

(i) Which parameters of SSL/TLS handshake can be used for client identification?
(ii) Can we pair selected SSL/TLS handshake parameters and HTTP header fields?
(iii) How much information, i.e., number of known SSL/TLS parameters and pairings to HTTP headers, do we need to analyze a significant portion of network traffic?
(iv) Can we utilize the SSL/TLS fingerprinting in network security monitoring and intrusion detection?

While it is important to collect and study malware, this is only a means to an end. In fact, it is crucial that the insight obtained through malware analysis is translated into detection and mitigation capabilities to eliminate malicious code running on infected machines

JUSTIFICATION FOR THE STUDY:

As we have seen lately (J. Aycock., 2006) the need of cyber security is on the rise as we encounter online security treats daily. The need to handle these treats has become absolutely important as people loose valuable data daily without their consent. It is imperative to conduct a study in this area.

OBJECTIVES OF THE STUDY

The Aim of this research to improve the detection of malware that uses network traffic encryption to avoid detection. To achieve this, we utilize information that is currently not utilized by the IDS inter-network data correlations. Although the IDS is employed on hundreds of different networks, the existing anomaly processing pipeline makes no attempt to share information between individual net-works and all the information that is used for anomaly detection in local network.

1. We propose a novel way of improving the efficacy of anomaly detection by using threat intelligence gathered from a large number of enterprise networks that are all separately monitored by the IDS
2. We create a global intelligence database that is shared among all the participating IDS systems and use it to improve the anomaly detection performance on both encrypted and non-encrypted network traffics.
3. This system will be more efficient than most existing systems.

The system is also cheaper and faster to deploy compared to most existing systems.

LITERETURE REVIEW

Traditional malicious network communication techniques rely either on port-based classification or deep packet inspection and signature matching techniques. Port-based methods rely on inspection of Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port numbers (Yoon et al., 2009) and assumption that applications always use well-known port numbers that are registered by the Internet Assigned Numbers Authority (IANA). (Dreger et al., 2006) showed that malicious applications use non-standard ports to evade network intrusion detection systems (NIDS) and restricting firewalls. Even prominent applications such as Skype use dynamic port numbers to escape restrictive firewalls (Baset at al., 2006). (Madhukar et al., 2006) showed that port-based classification mis-classifies network flow traffic 30-70% of the time.(Etienne., 2009) used deep packet inspection to detect malicious traffic by inspecting payload contents and using traditional pattern matching or signature-based techniques. Etienne used Snort (Snort., 2016), an intrusion detection application, to detect malicious traffic using signature or string matching on the packet contents. Snort also hosts a popular Intrusion Protection System (IPS) rule set maintained by the community (Snort, 2016). But only around 1 % of the rule set are TLS specific which shows that traditional pattern matching techniques are not used often for TLS based malware.

METHODOLOGY

Machine learning is a field of computer science that helps us automate the process of learning from the data. It helps computers to learn and act without any human intervention (H. Kriegel,et al., 2011). The process of learning from the data can be broadly divided into two parts:

• Supervised Learning: In this the computer is given input data along with the desired outputs and the aim is to learn a function that maps the input data to the desired values.

• Unsupervised Learning: In this the computer is given only input data without any labels or desired outputs and asked to either predict new data or classify the given input data into separate classes. In this research we will label the input data and thus know if a particular connection is malicious or not. Since we already know the labels of the input data, we use supervised learning algorithms such as Support Vector Machine (SVM), Random Forest, XGBoost and Deep learning based algorithms to train our machine learning models on the input data.

We propose a global intelligence sharing model, henceforth called the global reputation model (K. Scarfone et al., 2007), which uses information gathered in individual networks to globally associate anomaly values to individual hostnames. The overall reputation model structure is visualized. There are three parts to this reputation model;

1. Aggregation of anomaly scores over hostnames, producing a set of local reputation models. This part can be implemented simply as an average of per-flow anomaly values.

2. Combination of local reputation models to form a global reputation model. This part is much more involved than the others and presents difficult challenges such as normalization of individual local reputation tables. The remainder of this thesis mainly focuses on the design and implementation of this part of the overall intelligence sharing algorithm.

3. Improvement of anomaly detection in individual networks using the global reputation model. This part can be implemented by replacing individual flow anomaly scores with hostname reputation values.

Combination of local reputation models to form a global reputation model. This part is much more involved than the others and presents difficult challenges such as normalization of individual local reputation tables. The remainder of this thesis mainly focuses on the design and implementation of this part of the overall intelligence sharing algorithm.

First, showing how the reputation model can be viewed as an outlier ensemble model. Then, presenting our implementation choices of the three parts of the ensemble model that were identified normalization, combination, and the way of dealing with missing values.

GLOBAL REPUTATION MODEL AS AN OUTLIER ENSEMBLE:

In order to use the outlier ensemble abstraction, we must first transform our input from the form of network flows with anomaly scores into a local reputation model. We do this by treating each flow's anomaly score as an estimate of the corresponding hostname anomaly score. We then estimate the host name anomaly scores by averaging the anomaly scores of all flows that communicate with said hostname in a single network. This simple procedure arrives at the local reputation model.

The global reputation model can be viewed as a data-centered outlier ensemble with horizontal sampling. Let H be the set of all existent hostnames and $H_n \subseteq$ H the set of all hostnames that were observed in the network n ∈ N. Let $f_n : H_n \rightarrow R$ be the anomaly score of a hostname measured in the network $_n$. Now, H_n is the horizontal sampling of H, and the sets of anomaly scores observed in individual networks form an ensemble E_h:

$$E_h = \{\{f_n(h) \mid h \in H_n\} \mid n \in N\}.$$

STATISTICAL ANALYSIS:

- Data obtained from the studies will be expressed as Mean ± Standard Error of Mean (SEM)
- Tables, figures and graph will also be used to summarize data

RESULTS/FINDINGS

A software will be developed to improve the detection efficacy of the existing Cognitive Threat Analytics (CTA) anomaly-based network IDS on malware that uses network traffic encryption to 87%.

CONCLUSION

In conclusion the study has shown that there is an increase in the amount of communication traffic worldwide as the Internet keeps growing into rural communities. On the other-hand malware using HTTPS traffic for their communications is also on the rise (S. Yoon et al., 2009). This situation poses a challenge for the security analysts, as most of the current systems are inefficient and there is the need to discover new features and methods to detect malware. This study therefore, proposed system that will be highly efficient, cheaper, faster and private and also respecting the original idea of HTTPS.

REFERENCES

1) Madhukar and C. L. Williamson, ''A longitudinal study of P2P traffic classification,'' in 14th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, ser. MASCOTS 2006. IEEE Computer Society, 2006, pp. 179--188.

2) W. Moore and K. Papagiannaki, ‘‘Toward the accurate identification of network applications,’’ in Proceedings of 6th International Workshop on Passive and Active Network Measurement, ser. PAM 2005, C. Dovrolis, Ed. Springer, 2005, pp. 41--54.

3) Atilla Özgür and Hamit Erdem. A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ PrePrints, 4:e1954v1, 2016.

4) Business and New Computing Services, 12th Asia-Pacific Network Operations and Management Symposium, ser. APNOMS 2009, C. S. Hong, T. Tonouchi, Y. Ma, and C. Chao, Eds. Springer, 2009, pp. 21--30.

5) Hans-Peter Kriegel, Peer Kroger, Erich Schubert, and Arthur Zimek. Interpreting and unifying outlier scores. In Proceedings of the 2011 SIAM International Conference on Data Mining, pages 13–24. SIAM, 2011.

6) H. Dreger and A. Feldmann, ‘‘Dynamic application-layer protocol analysis for network intrusion detection,’’ in Proceedings of the 15th USENIX Security Symposium, ser. USENIX Security ’15, A. D. Keromytis, Ed. USENIX Association, 2006.

7) José Ramón Pasillas-Díaz and Sylvie Ratté. An unsupervised approach for combining scores of outlier detection techniques, based on similarity measures. Electronic Notes in Theoretical Computer Science, 329:61 – 77, 2016.

8) J. Aycock, Computer Viruses and Malware, ser. Advances in Information Security. Springer, 2006, vol. 22.

9) K Scarfone and P Mell. NIST SP 800-94: Guide to Intrusion Detection and Prevention Systems (IDPS), February 2007.

10) Leo Breiman. Bagging predictors. Machine learning, 24(2):123–140, 1996.

11) Leo Breiman. Random forests. Machine learning, 45(1):5–32, 2001.

12) L. Etienne, ‘‘Malicious traffic detection in local networks with snort,’’ https://infoscience.epfl.ch/record/141022/files/pdm.pdf, 2009.

13) MartinGrill. Combining Network Anomaly Detectors. PhDthesis, CzechTechnical University in Prague, 2016.

14) S. Sen, O. Spatscheck, and D. Wang, ‘‘Accurate, scalable in-network identification of p2p traffic using application signatures,’’ in Proceedings of the 13th international conference on World Wide Web, ser. WWW 2004, S. I. Feldman, M. Uretsky, M. Najork, and C. E. Wills, Eds. ACM, 2004, pp. 512--521.

15) S. Yoon, J. Park, J. Park, Y. Oh, and M. Kim, ‘‘Internet application traffic classification using fixed ip-port,’’ in Proceedings of Management Enabling the Future Internet for Changing Business and New Computing Services, 12th Asia-Pacific Network Operations and Management Symposium, ser. APNOMS 2009, C. S. Hong, T. Tonouchi, Y. Ma, and C. Chao, Eds. Springer, 2009, pp. 21--30.

16) S. Baset and H. Schulzrinne, ‘‘An analysis of the Skype peer-to-peer internet telephony protocol,’’ in 25th IEEE International Conference on Computer Communications, ser. INFOCOM 2006. IEEE, 2006.

17) Snort, ‘‘Snort --- Network intrusion detection and prevention system,’’ https: //www.snort.org, 2014.

18) ‘‘Snort: Community rules,’’ https://www.snort.org/downloads/community/ community-rules.tar.gz, 2016.

19) Stefan Axelsson. The base-rate fallacy and the difficulty of intrusion detection. ACM Transactions on Information and System Security (TISSEC), 3(3):186– 205, 2000.

20) Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3):15, 2009.