International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 3

# Defensing Confidentiality During Complete Packet Inspection On A Middlebox

**K. Geetharani[1], K. Kowsalya[2], A. M. SenthilKumar[3], M. S. Vijaykumar[4], M. Saravanakumar[5]**

[1, 2, 3, 4, 5] Department of CSE, Tejaa Shakthi Institute of Technology for Women, Coimbatore, Tamil Nadu, India
[3]HOD, [4,5]Assistant Professor

## ABSTRACT

In Internet to encrypt traffic, HTTPS provides secure and private data communication between clients and servers. Network operators often deploy middleboxes to perform deep packet inspection (DPI) to detect attacks using techniques ranging from simple keyword matching to more advanced machine learning and data mining analysis. But this approach cannot protect users' private information from a service provider who deploys middleboxes. SPABox, a middlebox-based system that supports both keyword-based and data analysis-based DPI functions over encrypted traffic. SPABox preserves privacy by using a novel protocol with a limited connection setup overhead. In this paper to further improve the performance, we are working on the network performance requirements.

***Keywords***: *DPI, middlebox, privacy preserving.*

## I. INTRODUCTION

HTTPS is a popular Internet protocol which uses Transport Layer Security (TLS) or its predecessor, Secure Sockets Layer (SSL) to encrypt communication between clients and servers to ensure data integrity and privacy. Many middleboxes that provide deep packet inspection (DPI) functionalities are deployed by network operators to detect attacks by searching for specific keywords or signatures in non-encrypted traffic [9]. As malwares use various concealment techniques such as obfuscation, and polymorphic or metamorphic strategies to try to evade detection both industry and academia have considered adding more advanced machine learning and data mining analysis in DPI[10],[12],[13]. With the growing adoption of HTTPS, existing approaches are unable to perform keyword or signature matching, let alone advanced machine learning analysis for malware detection of the encrypted traffic. In this paper, we present SPABox, the first middlebox based system that supports both signature and data analysis based DPI functionalities over encrypted traffic with improved network performance requirements. SPABox supports three types of operations: keyword matching, regular expression evaluation and malware detection via machine learning .The main ideas of our design behind these three operations are 1) instead of matching keywords directly, we tokenize the keywords so that we can build a Trie-like structure to accelerate searching and matching at a DPI middlebox, 2) regular expressions are processed at the receiver side, and garbled circuits and oblivious transfer are used to protect the privacy of both traffic and rule sets, and 3) we utilize the homomorphic properties on addition and scalar multiplication, respectively. To validate our design, we analyze the security guarantees of our design, implement SPABox on a standard server and evaluate its performance with several real data sets and traffic. We consider performance metrics such as throughput, bandwidth and memory overhead on the sender, middlebox and receiver sides, and show that SPABox is practical for both long-lived and short-lived connections.

## II. System Architecture

Fig. 1 shows the system architecture, and the highlighted boxes indicate components added by SPABox. As in the previous work [4] there are four parties: sender (S), receiver (R), middlebox (MB) and rule generator (RG).

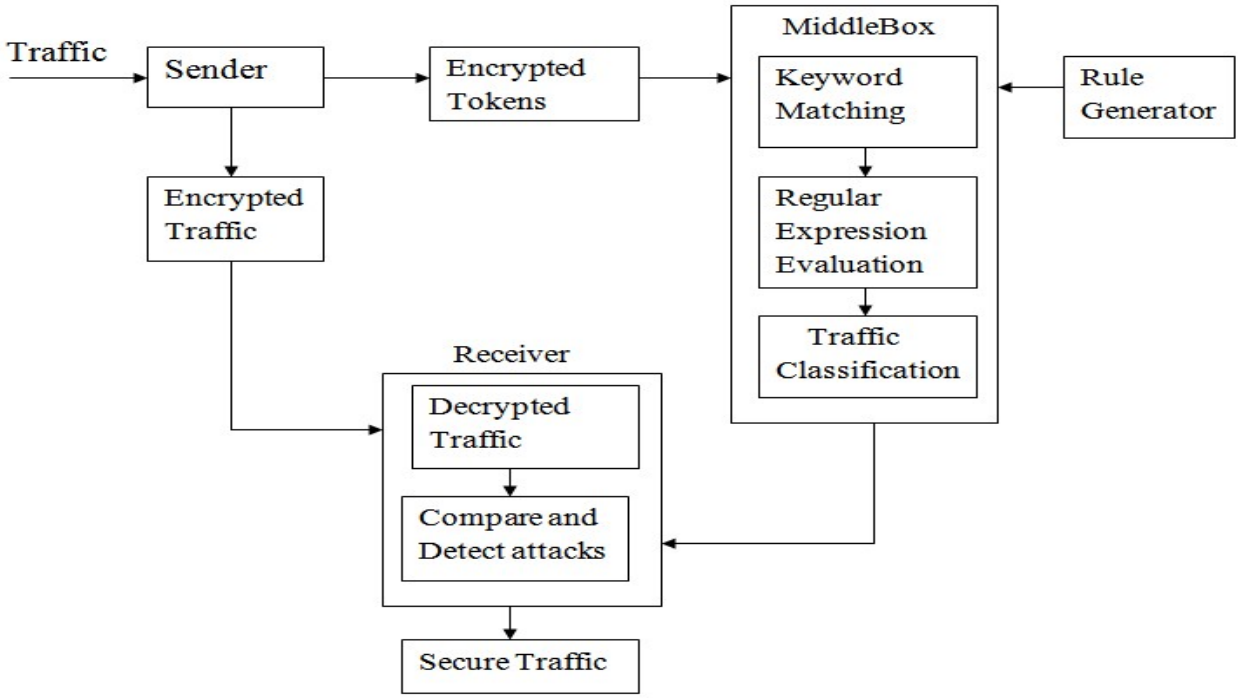


**Fig 1:** SPABox system architecture

## III. Proposed System

### A. Sending traffic

At Sender, two logical connections are setup, to be referred to as an SSL connection and an SPA connection, respectively. On the SSL connection, S encrypts the traffic with unmodified SSL. On the SPA connection, S makes a copy of the traffic, tokenizes it and then encrypts the tokens using the Triple DES algorithm.

Triple DES uses a "key bundle" that comprises three DES keys, K1, K2 and K3, each of 56 bits (excluding parity bits). The encryption algorithm is:

ciphertext = EK3(DK2(EK1(plaintext)))

Decryption is the reverse:

plaintext = DK1(EK2(DK3(ciphertext)))

### B. Keyword Matching

In this subsection, we present our protocol design for performing keyword matching at S, R and the MB [2]. Note that only few minor changes need to be made at the MB, S and R. An attack rule may contain multiple keywords as well as position information of these keywords. Only if all the keywords in one attack rule are matched with the correct offset, we consider this attack rule is matched. The MB compares the encrypted traffic with the rule set provided by the RG, and observes a matching between the traffic and one of the attack rules in the rule set. If there is a match between the traffic and the rule set and there is no regular expression in the corresponding attack rule, the MB can choose to drop the packet and inform administrator or issue a warning just as what a regular MB would do over unencrypted traffic, If a regular expression needs to be further evaluated, it will be forwarded to R for further processing.

### C. Regular Expression Evaluation

The reason for evaluating regular expressions is twofold. First, some rule sets require evaluation of regular expressions besides keyword matching [1]. By enabling such operations, all of the attack rules of many public and industrial rule sets can be addressed [8].If one rule contains keywords that are shorter that 5 bytes, Keyword matching would not work. To enable matching on keywords that are shorter than 5 bytes long, the MB can build one regular expression. A regular expression can be converted to a deterministic finite automata (DFA) with an accepting state F indicating that the input string contains malicious factors if this DFA finishes in state F[6].If the

corresponding DFA can achieve an accepting state, R would know that S is malicious (a hit of an attack rule) and can drop the packet.

### D. Traffic classification

S needs to tokenize the traffic using a sliding window, which is also called n-gram. N-grams used to represent features for malware detection using ML methods [5], [7].In order to perform ML at the MB, the object's features need to be extracted first. Then S encrypts these n-grams, and S sends the encrypted features over the SPA connection. So that the MB can perform classification [11], [3].In order to get the classification result, all R needs to do is to decrypt the classification result sent by the MB.

### E. Comparing traffic between SPA and SSL connection

By comparing the ciphertexts generated from the traffic over the SSL connection against the encrypted tokens received from the SPA connection, R can determine whether S in this session follows the SPABox protocol correctly, including both keyword matching and malware detection using ML. If there is any discrepancy, R may think S is an attacker and immediately drop the connection. Otherwise, R may process information forwarded by the MB containing the classification results and regular expressions to decide whether the traffic from S over the SSL connection is malicious or not.

### Drawbacks of Existing System

The tokenizing and encrypting process of an existing system may cause some delay at R side. Even though the randomized schemes provide stronger security guarantees, it results in a low throughput at the MB.

### Advantages of Proposed System

In our proposed system we use Triple DES algorithm. It reduces complexity. It improves the execution speed. Speed of encryption and decryption is very fast. It reduces the execution time.

### Conclusion

In this paper, SPABox supports both keyword based and data analysis based DPI functionalities over encrypted traffic, while guaranteeing the privacy of the user data at the middlebox. The most notable feature of SPABox is that protocol setup does not require any interaction or data transmission between a middlebox and clients.

## REFERENCES


1. D. Ficara *et al.*, "An improved DFA for fast regular expression matching,"*ACM SIGCOMM Comput.Commun.Rev.*,vol.38,no.5,pp.2940,2008
2. K. Namjoshi and G. Narlikar, "Robust and fast pattern matching for intrusion detection," in *Proc.IEEE INFOCOM*, Mar. 2010,pp. 740–748.
3. K. Rieck, T. Holz, C. Willems, and P. Düssel, and P. Laskov, "Learning and classification of malware behavior," in *Detection Intrusions Malware, Vulnerability Assessment*. Berlin, Germany: Springer, 2008, pp. 108–125.
4. J. Sherry, C. Lan, R. A. Popa, and S. Ratnasamy, "BlindBox: Deep packet inspection over encrypted traffic," in *Proc. ACM SIGCOMM*,2015, pp. 213–226.
5. J. Z. Kolter and M. A. Maloof, "Learning to detect and classify malicious executables in the wild," in *Proc. ACM SIGKDD*, 2004, pp. 1–6.
6. M. A. Salehi *et al.*, "RESeED: Regular expression search over encrypted data in the cloud," in *Proc. IEEE CLOUD*, Jul. 2014, pp. 673–680.
7. S. M. Tabish, M. Z. Shafiq, and M. Farooq, "Malware detection using statistical analysis of byte-level file content," in *Proc. ACM SIGKDD Workshop CyberSecur. Intell. Informat.*, 2009, pp. 23–31.
8. J. Sherry, C. Lan, R. A. Popa, and S. Ratnasamy, "BlindBox: Deep packet inspection over encrypted traffic," in *Proc. ACM SIGCOMM*,2015, pp. 213–226.
9. A. Bremler-Barr, Y. Harchol, D. Hay, and Y. Koral, "Deep packet inspection as a service," in *Proc. ACM CoNEXT*, 2014, pp. 271–282.
10. *Symantec Adds Deep Learning to Anti-Malware Tools to Detect Zero-Days*. Accessed: Jan.5,2016.[Online].Available:http://www.eweek.com/security/symantec-adds-deep-learning-to-antimalware-tools-to-detect-zero-days.
11. R.Bost,R.A.Popa,S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data,"*CryptoePrintArch.*,vol.2014,p.331,2014.
12. *Overcoming Targeted Attacks: A New Approach*.Accessed:Nov.31,2015.[Online]. Available:https://securingtomorrow.mcafee.com/mcafee-labs/overcoming-targeted-attacks-new-approach/
13. Y. Ye, D. Wang, T. Li, and D. Ye, "IMDS: Intelligent malware detection system," in *Proc. ACM SIGKDD*, 2007, pp. 1043–1047.