Impact Factor (SJIF): 4.542

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444
Volume 4, Issue 11, November-2017

Data Recovery And Intrusion Detection System Using 2SE(2-Seed Expansion)

Jaypal Momale
Dept. Computer Engineering, ICEM
Indira College of Engineering and Management
Pune, India

Amarjit Choudhary
Dept. Computer Engineering, ICEM
Indira College of Engineering and Management
Pune, India

Pratik Bakshi
Dept. Computer Engineering, ICEM
Indira College of Engineering and Management Pune, India

Aishwarya chavan
Dept. Computer Engineering, ICEM
Indira College of Engineering and Management
Pune, India

Abstract— **Although many different detection mechanisms have been proposed, exiting detection methods generally tend to successfully detect attacks only after the attacks have finished and caused damage to the system. As recent attacks employ polymorphism technology and complicated attack techniques, it has become even more difficult for these approaches to detect attacks in a timely manner. In this paper, we propose an efficient network attack detection algorithm called seed expanding (SE) that detects attacks before they damage the system. SE employs the Two-Seed-Expanding network traffic clustering scheme, which clusters attack traffic into different attack phases. First we pre-process the networks traffic, including constructing the network flow, changing continuous-valued attributes into nominal attributes by adopting the discretization method, and further turning into binary features. Then based on these features, SE computes a weight for each flow and iteratively selects seeds to expand until all flows are divided into clusters. To investigate the effectiveness of the proposed approach, we undertook extensive experimental analyses. The results of the experiment show that the pre-procession greatly improves clustering performance, and the Two-Seed-Expanding Algorithm is better than K-Means and other kinds of Seed-Expanding in attack-flow clustering. These cluster results can be further used in attack detection.**

Keywords— **malicious network traffic, attack detection, attack phase, network flow clustering.**

I. INTRODUCTION

To detect attacks earlier, it is important to identify what phase the attack is in as early as possible. To detect the attack phase, the first mission is to use some methods to distinguish different attack phases. Clustering analysis is a good technology for classifying network traffic into different attack phases, although it cannot mark the current attack phase. Results from clustering can further be used to analyze the features of the attack phase. The typical centroid-based clustering algorithm is K-Means. Distribution-based clustering utilizes the distribution information of a dataset to find the cluster. Density-based clustering finds the high density areas that are separated by low density areas. However, because we cannot control the similarity level of data points in the clustering process, these cluster algorithms are not appropriate for detecting malicious network flows. In other words, the network administrator cannot control the similarity level in the process of the clustering attack phase according to network policy.

II. LITERATURE SURVEY

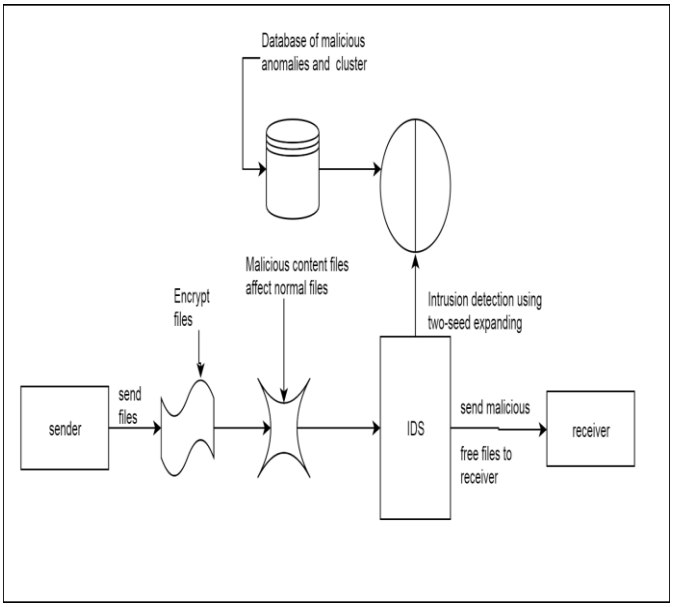
1. Accurate network anomaly classification with generalized entropy metrics 2011
2. Collaborative anomaly-based detection of large-scale internet attacks 2012
3. Internet traffic clustering with side information 2014
4. Combining Supervised and Unsupervised Learning for Zero-Day Malware Detection 2013
5 .Internet Traffic Classification Using Constrained Clustering, IEEE transactions on parallel and distributed systems,2013

III. PROBLEMS IN THE EXISTING SYSTEM

We design an efficient clustering algorithm which divides one attack into multiple phases, where every attack phase is considered as one type of attack behavior, including discovering a vulnerable host, getting root access privileges or gaining access to a vulnerable host, and gathering sensitive information from the Internet. Specially, although one attack may have one or multiple steps, we consider it as a multi-step attack with default steps. The clustering results will be used to further detect malicious network traffic. We further expand SE to other kinds of Seed Expanding, and discuss how many chosen seeds are appropriate in Seed Expanding.

The Clustering analysis is a good technology for classifying network traffic into different attack phases, although it cannot mark the current attack phase. Results from clustering can further be used to analyze the features of the attack phase.

IV. ARCHITECTURE



V. ALGORITHM

1) AES Algorithm
AES 256:

In the implementation of this AES-256 algorithm has a plaintext of 128 bits and key of 256 bits size. The number of rounds of operations in AES- 256 is 14. The key generation process of AES 256 is different from other AES algorithms.

The AES-256 algorithm is composed of three main parts: Cipher, Inverse Cipher and Key Expansion. Cipher converts data to an unintelligible form called cipher text while Inverse Cipher converts data back into its original form called plaintext. Key Expansion generates a Key Schedule that is used in Cipher and Inverse Cipher procedure. Cipher and Inverse Cipher are

composed of specific number of rounds For both its Cipher and Inverse Cipher, the AES algorithm uses a round function that is composed of four different byte-oriented transformations:
1) Byte substitution using a substitution table (S-box)
2) Shifting rows of the State array by different offsets
3) Mixing the data within each column of the State array
4) Adding a Round Key to the State

The Cipher transformations can be inverted and then implemented in reverse order to produce a straightforward Inverse Cipher for the AES algorithm. The individual transformations used in the Inverse Cipher are listed below.
1) Inverse Shift Rows
2) Inverse Sub Bytes
3) Inverse Mix Columns
4) Add Round Key

The AES inverse cipher core consists of a key expansion module, a key reversal buffer, an initial permutation module, a round permutation module and a final permutation module. The key reversal buffer first store keys for all rounds and the presents them in reverse order to the rounds. The round permutation module will loop maternally to perform 14 iterations (for 256 bit keys).

IMPLEMENTATION OF AES 256

In the Encryption process we have a plaintext of 128 bits and key of 256 bits size. The number of rounds in AES 256 is 14. The first round include all the five operation like Pre-round operation, sub byte, shift rows, mix columns and Add round key operations. From 2nd round to 13th round have four operations sub byte, shift rows, mix columns and Add round key operations. And the last 14th round consists of three operations sub byte, shift rows and Add round key operations.

Description of the AES-256 Encryption algorithm:
1. Key Expansion round keys are derived from the cipher key using Rijndael's key schedule.
2. Initial Round.
a) Add Round Key each byte of the state is combined with the round key using bitwise xor.
b) Sub Byte a non-linear substitution step where each byte is replaced with another according to a lookup table.
c) Shift Rows a transposition step where each row of the state is shifted cyclically a certain number of steps.
d) Mix Columns is a mixing operation which operates on the columns of the state, combining the four bytes in each column.
e) Add Round Key.
3. Rounds (2nd to 13th)
a) Sub Bytes
b) Shift Rows
c) Mix Columns
d) Add Round Key
4. Final Round (no Mix Columns)
a) Sub Bytes
b) Shift Rows
c) Add Round Key

In AES 256 the process of generating the key is each round key is a 256-bit array generated as follows:

Input key of 256 bit is divided into eight parts of 32 bits as columns in a matrix4*8. Last column is taken and given as input to S box. The achieved output of S box is given shift rows operation. The output MSB side 8 bits are xored with the round constant (i.e. round constant value is different for different rounds). The achieved output is xored with 0th column of input key.
The 0th column of new key is xored with 1st column of input key gives 1st column of new key.
1st column of new key is xored with 2st column of input key gives 2nd column of new key.
2nd column of new key is xored with 3rd column of input key gives 3rd column of new key.

The above obtained 3rd column of new key is given to S box. The output of S box is xored with 4th column of input key which gives 4th column of new key.
4th column of new key is xored with 5th column of input key which gives 5^{th} column of new key.
5th column of new key is xored with 6th column of input key which gives 6^{th} column of new key.
6th column of new key is xored with 7th column of input key which gives 7^{th} column of new key.

In this way we generate new keys of 256 bits in AES 256 algorithm by attaching the eight obtained columns of new key. In the first round this key of 256 bits is divided into two parts each of 128 bits size and these keys of 128 bits are used one in the pre-round operation (i.e. xor operation between plaintext and key) and other is used in Add Round key operation. At the end of round function there will be 128bit output and 256 bit key output obtained from key generation process. In the second round as we don't have pre-round operation so the round output of 1st round is applied as input to sub byte and the remaining operations are same as round 1.

This process of round operation is repeated up to 13th round operation. And the last round is similar to previous round the only change is it does not have mix columns operation. AES Decryption process is the inverse process of encryption AES. The output of the encryption process i.e. cipher text is given as input to decryption. The same input key of 256 bits is used as another input. The input key is given to the key generation module to generate new keys as we do in the encryption process. The output keys generated are given as inputs to inverse mix columns which give keys for the fourteen rounds of decryption

2) two seed expansion

A. Similarity Computing

For asymmetric binary attributes, the two states are not equally important. Given two asymmetric binary attributes, the agreement of two 1s (a positive match) is then considered more significant than that of two 0s (a negative match).

The process of the algorithm is as follows. The inputs of algorithm include D and threshold r. The D is the set of data points {d1, d2, dn}. Threshold r is a pre-value. We can control the similarity level in the clustering process by adjusting the number of thresholds. First, Weight Computing is finished. Next, the two steps of Seed Selection and Seed Expanding are run continuously when the size of Q is larger than 2. Especially, two seeds are selected, then two cluster for the two seeds are expanded until the seed and the data points of a candidate set is less than the threshold. When Q is empty, the above two steps are stopped. When Q has only one data point, the data point is considered as a single cluster. When Q contains two data points p and q, the similarity sim(p, q)between two data points are calculated. If sim(p, q) >= r, p and q constitute one cluster. Otherwise, the two points respectively constitute two new clusters. Finally, the step Noise
Removal is run and all clusters are returned.

REFERENCES

[1] B. Tellenbach, M. Burkhart, D. Schatzmann, D. Gugelmann, D. Sornette. Accurate network anomaly classification with generalized entropy metrics, Computer Networks 2011; 55(15): 3485-3502.

[2] T. Gamer. Collaborative anomaly-based detection of large-scale internet attacks, Computer Networks 2012, 56(1): 169?85.

[3] J. Mazel, P. Casas, Y. Labit and P. Owezarski. Sub-space clustering, inter-clustering results association and anomaly correlation for unsupervised network anomaly detection. In Proceedings of the 7th International Conference on Network and Service Management, Paris, France, 2011: 1-8.

[4] P. Casas, J. Mazel, and P. Owezarski. Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge. Computer Communications, April 2012, 35(7): 772-783.

[5] Y. Wang, Y. Xiang and Jun Zhang et.al., Internet Traffic Classification Using Constrained Clustering, IEEE transactions on parallel and distributed systems, 25(11):2932-2943, 2014.

[6] A.W. Moore, D. Zuev, M.L. Crogan. Discriminators for use in flowbased classification. Technical Report, RR-05-13 [R]. UK: Queen Mary University of London, 2005.

[7] Y. Wang, Y. Xiang, J. Zhang, W.L. Zhou, B.L. Xie. Internet Traffic clustering with side information. Journal of Computer and System Science, 2014, 80: 1021-1036.

[8] T. Gamer. Collaborative anomaly-based detection of large- scale internet attacks, Computer Networks 2012, 56(1): 169?85.

[9] J. Mazel, P. Casas, Y. Labit and P. Owezarski. Sub-space clustering,inter-clustering results association and anomaly correlation for unsupervised network anomaly detection. In Proceedings of the 7th International Conference on Network and Service Management, Paris, France, 2011:1-8.

[10] P. Casas, J. Mazel, and P. Owezarski. Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge. Computer Communications, April 2012, 35(7): 772-783.

[11] Y. Wang, Y. Xiang and Jun Zhang et.al., Internet Traffic Classification Using Constrained Clustering, IEEE transactions on parallel and distributed systems, 25(11):2932-2943, 2014.

[12] A.W. Moore, D. Zuev, M.L. Crogan. Discriminators for use in flowbased classification. Technical Report, RR-05-13 [R]. UK: Queen Mary University of London, 2005.

[13] Y. Wang, Y. Xiang, J. Zhang, W.L. Zhou, B.L. Xie. Internet Traffic clustering with side information. Journal of Computer and System Science, 2014, 80: 1021-1036.

[14] L.L Yang, J. Wang, P. Zhong. Combining Supervised and Unsu-pervised Learning for automatic attack signature generation system. Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing. Dalian: Springer Verlag, 2014: 607-618.

[15] U.M. Fayyad and K. B. Irani, Multi-interval discretization of continuousvalued attributes for clasification learning, In Proceedings of the Internationaloint