Peer to Peer Botnet Detection for Cyber-Security: A Data Mining Approach

Mohammad M. Masud
Dept. of Computer Science
University of Texas at Dallas
mehedy@utdallas.edu

Jing Gao
Dept. of Computer Science
University of Illinois at Urbana-Champaign
jinggao3@uiuc.edu

Latifur Khan
Dept. of Computer Science
University of Texas at Dallas
lkhan@utdallas.edu

Jiawei Han
Dept. of Computer Science
University of Illinois at Urbana-Champaign
hanj@cs.uiuc.edu

Bhavani Thuraisingham
Dept. of Computer Science
University of Texas at Dallas
bxt043000@utdallas.edu

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms

Keywords

botnet, classification, data stream, ensemble

1. EXTENDED ABSTRACT

Botnet is a network of compromised hosts or *bots*, under the control of a human attacker known as the *botmaster* [7, 8] . Botnets are used to perform malicious actions, such as launching DDoS attacks, sending spam or phishing emails and so on. Thus, botnets have emerged as a threat to internet community. Peer to Peer (P2P) is a relatively new architecture of botnets [4]. These botnets are distributed, and small. So, they are difficult to locate and destroy. Most of the recent works in P2P botnet are in the analysis phase [4, 5, 6]. On the contrary, our work is aimed at *detecting* P2P botnets using network traffic mining.

CSIIRW '08, May 12-14, Oak Ridge, Tennessee, USA

Network traffic can be considered as an infinite data stream. So, our data mining approach is specialized for mining stream data. There are two major problems related to stream data classification. First, it is impractical to store and use all the historical data for training, since it would require infinite storage and running time. Second, there may be concept-drift in the data. For example, in the context of botnets, the botmaster usually updates the bot software frequently, which may change the characteristics of botnet traffic, resulting in a concept drift in the data. If there is a concept-drift in the data, we need to refine our hypothesis to accommodate the new concept. Thus, most of the old data must be discarded from the training set. There are two mainstream techniques available for stream data classification: single classifier approach [1], and ensemble classifier approach [10, 9]. Among these, the ensemble classifier is often more robust in handling concept drifts. We also propose an ensemble classification approach for that solves both the problems related to stream data classification.

A common approach in classifying stream data is to divide the stream data into equal sized chunks [2, 10, 9, 3]. We also follow this approach. However, instead of storing historical data, we store the trained classifiers. We always store an ensemble A of best K classifiers $\{A_1,...,A_K\}$. The ensemble A is actually a two-level ensemble. That is, each classifier A_i in the

ensemble A is actually a collection (ensemble) of v classifiers. Thus, we build a hierarchy of ensembles, where A is at the top level of the hierarchy, and each of its children A_i is at the middle level. The lowest level (or the leaves) contains the actual classifiers.

Each middle-level ensemble A_i is trained with r consecutive data chunks. As soon as a new data chunk appears, we train a new middle-level ensemble A_n. Let D=$\{D_n, D_{n-1}, ..., D_{n-r+1}\}$, i.e, the most recent r data chunks including D_n. We randomly divide D into v equal parts = $\{d_1, ..., d_v\}$, such that roughly, all the parts have the same number of positive and negative examples. We then build A_n with v classifiers = $\{A_{n(1)}, A_{n(2)}, ... , A_{n(v)}\}$, where each classifier $A_{n(j)}$ is trained with the dataset D - $\{d_j\}$. We compute the expected error of the ensemble A_n by testing each classifier $A_{n(j)}$ on d_j and averaging their error. Finally, we update the top-level ensemble A by replacing a middle-level ensemble $A_i (1 \leq i \leq K)$ with the new ensemble A_n, if A_n has lower error rate than A_i. By introducing this multi-chunk multi-level ensemble, we reduce the expected error by a factor of rv over the single-chunk, single-level ensemble method (e.g. [10]). We prove the effectiveness of our approach both theoretically and empirically.

We have several contributions. First, we propose a novel multi-chunk, multi-level ensemble technique for stream data classification, which is a generalization over the existing single-chunk single-level ensemble techniques. Second, we prove the effectiveness of our technique theoretically. Finally, we apply our technique on for detecting P2P botnet traffic, and achieve better detection accuracies than other stream data classification techniques. No botnet detection techniques so far applied the stream classification approach. We believe that the proposed ensemble technique provides a powerful tool for network security and it will encourage the future use of stream data classification in botnet detection.

2. REFERENCES

[1] P. Domingos and G. Hulten. Mining high-speed data streams. In *Proc. SIGKDD*, pages 71–80, 2000.

[2] W. Fan. Systematic data selection to mine concept-drifting data streams. In *Proc. KDD*, pages 128–137, 2004.

[3] J. Gao, W. Fan, and J. Han. On appropriate assumptions to mine data streams. In *Proc. ICDM*, 2007.

[4] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon. Peer-to-peer botnets: Overview and case study. In *Usenix/Hotbots '07 Workshop*, 2007.

[5] L. T. I. Group. Sinit p2p trojan analysis. lurhq. *http://www.lurhq.com/sinit.html*, 2004.

[6] R. Lemos. Bot software looks to improve peerage. *http://www.securityfocus.com/news/11390*, 2006.

[7] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proc. of the 6th ACM SIGCOMM on Internet Measurement Conference (IMC)*, 2006.

[8] B. Saha and A. Gairola. Botnet: An overview. *CERT-In White Paper CIWP-2005-05*, 2005.

[9] M. Scholz and R. Klinkenberg. An ensemble classifier for drifting concepts. In *Proc. ICML/PKDD Workshop in Knowledge Discovery in Data Streams.*, 2005.

[10] H. Wang, W. Fan, P. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *Proc. KDD*, 2003.