Sniper: Social-link Defense for Network Coordinate Systems

Xiaohan Zhao†, Eng Keong Lua‡, Yang Chen†,Xiaoxiao Song†, Beixing Deng†, Xing Li†
†Tsinghua National Laboratory for Information Science & Technology,
Electronic Engineering of Tsinghua University, Beijing 100084, China
E-mail: homeisland03@gmail.com; chenyang04@mails.tsinghua.edu.cn
‡Carnegie Mellon University, Pittsburgh, PA15213, U.S.A.
E-mail: englua@cmu.edu

Abstract—**Recent work on securing the Network Coordinate (NC) service attempts to defend attacks based on nodes' past behavior and activities. They detect malicious nodes only after attacks occurred. In order to secure NC system before malicious nodes mount attacks, we propose a social-link defense system for NC service---Sniper. We are able to provide NC system with trusted neighbors (landmarks) using social-link filtering technique. Our preliminary results demonstrate that Sniper can protect NC system successfully before they are attacked by malicious nodes.**

Keywords-Security; Social Link ; Network Coordinate;

I. Introduction

Network Coordinate (NC) Systems, which can estimate network delay with low overhead accurately, have been proposed to support application overlays. Previous studies prove that NC Systems are vulnerable to attacks. Thus, several security policies, such as RVivaldi[1], Kalman Filter[2] and Veracity[3], have been designed to solve security problem. However, they defend attacks based on nodes' previous behavior and cannot detect potential malicious nodes before attacks launch. Considering relationship between nodes in networks is the same as social link in social network, we propose a social-link based proactive security system named *Sniper*. The key idea of Sniper is to filter out suspicious nodes before they launch potential attacks by utilizing existing social link. To the best of our knowledge, this is the first social-link based defense system for NC service. Furthermore, we study the performance of Sniper to protect NC systems from attacks. The results prove that Sniper can defend NC successfully.

The remaining part of this paper is organized as follows. Section 2 gives a detailed description of Sniper. Section 3 shows the performance of Sniper. The paper concludes with a brief summary and a vision on the future work in section 4.

II. Sniper Design

A. Sniper Overview

Sniper is deployed in NC system provided by NC service provider. A typical NC system such as Vivaldi[4] consists of several NC servers that are setup and managed by the NC service provider and other nodes. A DHT-based overlay network can be easily implemented to manage this set of NC servers which act also as the bootstrap nodes of the overlay operation. DHT-based overlay network can provide effective lookup service during NC server churn or failure, thus it has self-organization and fault tolerance characteristics. These NC servers (overlay bootstrap nodes) can be considered to be trusted set of overlay nodes as managed and controlled by the NC service provider.

These NC servers are responsible to provide reliable and trusted lookup responses in terms of the usage frequency of the nodes by other nodes --- *History Profile*. When a node joins the NC service (we use Vivaldi as a running example), Sniper will be installed in the node and run. Sniper is able to provide the facility to select and validate neighbor (or landmark) nodes that are used for the computation of node coordinates. We describe the security operation in the following sub-sections.

B. Sniper Security Mechanism

There are three steps in Sniper: Before computation, node X starts to learn social link information from its direct social link named *direct friends*. Note that Sniper does not require blacklist used in DST (Dempster-Shafer Theory) of SLINCS[5], a naive social link based evaluation idea for NC. Then, after sythetical rank, it selects the more trustable neighbors based on social link evaluation results. Finally, validation process is embedded into NC computation to feed back these neighbors' performance. To adapt dynamic change in network, the process of social link learning and synthetical rank will be repeated every T computation rounds and in this paper we set T=30.

1. **Social Link Learning**: Since the information of direct friends a node has is too limited to make a judgment in later steps. Node X will learn social relationship from its direct friends, including both their direct friends and their indirect social link learnt from others. We define *T(X, B),* a number between 0 and 1, as the direct social trust score X has on B. And we define $T_B(X,C)$ as social trust score node X has on node C based on the social information provided by node B. Then node X rates the social trust score of every node learnt from its direct friends following equation (1) and if there are *m* friends introducing node C, an average trust score can be computed with equation (2) to give a fair view to every friend.

2. **Synthetical Rank**: With social trust scores computed in step 1, node X ranks its neighbor candidates and thus every candidate gets its rank number named *social rank*. Then, node X asks the corresponding bootstrap nodes for the candidates' history profile to rank them called *history rank*. With the social rank R_s, history rank R_t and the degree a node trusts its social

978-1-4244-3968-3/09/$25.00 ©2009

link *d* which is a number between 0 and 1, *mixed trust* for every candidate can be computed through equation (3). Finally, by ranking mixed trust *R*, node X selects the first *K* nodes as its neighbors or landmarks.

$$T_B(X,C) = T(X,B)\bullet\overline{T(B,C)} \qquad (1)$$

$$\overline{T(X,C)} = \frac{\sum_{n=1}^{m} T_n(X,C)}{m} \qquad (2)$$

$$R = d \times R_s + (1-d) \times R_t \qquad (3)$$

3. **Neighbor Validation**: In case there are remaining malicious nodes in neighbors, node X will chose *M* nodes with highest social trust scores to verify its neighbors before they are used in NC computation. Firstly, the *M* nodes measure the distances to a neighbor through simple ping and compute relative errors between the measured distances and the predicted distances. Then, if the relative error is less than α, the node will vote that this neighbor can be trusted; otherwise, we can classify the neighbor as malicious. We monitor the parameter of the number of nodes that agree a neighbor is honest. If this parameter is more than N, node X will utilize its information to compute coordinates at the same time notify the relevant bootstrap node to update this neighbor's history profile by adding one. In contrast, when a node is voted as malicious by more than N nodes, node X will report it to the bootstrap node. If the node is proved to be malicious by the verification from bootstrap nodes, its history profile will be halved and other similar reports will be ignored in next 10 rounds.

III. PERFORMANCE EVALUATION

We use a data set of measured RTT between 226 nodes of PlanetLab in our simulation. In order to accord with the authentic social networks between 226 nodes, we utilize a group including 226 nodes from YouTube social data set containing 30087 groups. Every node here is mapped into 5-dimentional space using 10 neighbors selected from 30 candidates. The parameters used in Vivald NC computation algorithm are $c_e = 0.25$ and $c_c = 0.25$. In Sniper, we set up *L*=20 bootstrap nodes. The parameters in step 3 are M=5, N=3, α=0.3. In addition, the direct social trust score *T(X,B)* is 0.95 and *d* in equation (3) is set to 0.8.

Malicious nodes attack others in this paper with inflated Round Trip Time (RTT) and their computed coordinates. In specific, the inflated RTT is 2.0 times more than the original one. Every malicious node starts its attack at the 6000th computation round and every node runs NC 30000 rounds. We repeat the whole computation process 20 times with 10%, 20%, 30% and 40% malicious nodes respectively. We use Relative Error and Relative Rank Loss (RRL) [6] as performance evaluation metrics.

Since the results are rather similar when there are different malicious nodes in the system, here we only plot the results of 10% and 30% malicious nodes. Fig. 1 shows CDF of relative error and RRL of victims of attack with 10% and 30% malicious nodes. From this figure, we find that when malicious nodes are present, Sniper outperforms Vivaldi based on both of the two metrics. Especially in the metric of RRL, the performance of Sniper with 30% malicious nodes is much better than that of Vivaldi attacked by 10% malicious nodes.

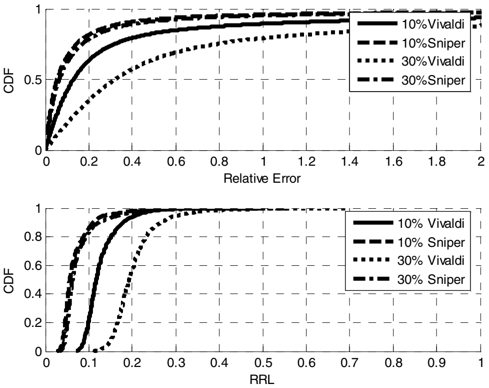

Fig.1 CDF of Relative Error and RRL in Sniper & Vivaldi

IV. CONCLUSION

Unlike proposed security mechanisms that use nodes' previous record to detect malicious nodes after attacks took place, this paper presents Sniper, a security system importing social link to detect suspicious nodes before they mount attacks. To the best of our knowledge, this is the first defense system for NC service based on social link. According to our simulation result, we have demonstrated that Sniper can protect Vivaldi against attacks successfully. In the future, we will implement Sniper into real network to evaluate its performance and to study the influence of social link on securing NC system under various attacks.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China (No.60850003, No.60473087, No.60703052) and the National Basic Research Program of China (No.2007CB310806).Thanks to Alan Mislove and colleagues at MPI for providing social link data on YouTube users.

REFERENCES

[1] D. Saucez, B. Donnet and O. Bonaventure. A Reputation-Based Approach for Securing Vivaldi Embedding System. Lecture Notes in *Computer Scienc*e, vol 4606, 2007.

[2] M.A. Kaafar, L. Mathy, C. Barakat, K. Salamatian, T. Turletti and W. Dabbous. Securing Internet Coordinate Embedding Systems. In proceedings of *ACM SIGCOMM'07*.

[3] M. Sherr, B. T. Loo and M. Blaze. Veracity: A Fully Decentralized Service for Securing Network Coordinate Systems. In proceedings of *IPTPS'08*.

[4] F. Dabek, R. Cox, F. Kaashoek and R. Morris, Vivaldi: A decentralized network coordinate system. In Proceedings of the *ACM SIGCOMM'04.*

[5] E. K. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft. On the Accuracy of Embeddings for Internet Coordinate Systems. In *proceedings of IMC'05*.

[6] X. X. Song, X. H. Zhao, E. K. Lua, Z. B. Zhang, B. X. Deng and X. Li. SLINCS: A Social Link based Evaluation System for Network Coordinate Systems. In proceedings of *IEEE CCNC'09*.