

A Scalable Method for Router Attack Detection and Location in Link State Routing

Anirban Chakrabarti and G. Manimaran *
{anirban,gmani}@iastate.edu

Abstract

The routing table poisoning attack is one of the most devastating and least researched topic among Internet attacks, which needs immediate research attention. In this paper, we develop a scalable method for detecting router attacks and locating the malicious routers (within a small bounded set of nodes) in link state routing protocols. We carry out analytical and simulation studies to evaluate the proposed secure link state protocol (SLIP) for two performance metrics, viz. attack detection probability and fault detection time, under different network and attack scenarios. Our studies show that the SLIP offers a very high attack detection capability with a little degradation in fault detection time compared to the link state protocol.

1 Introduction

The Internet has been witnessing enormous growth over the last several years. Due to the enormity of the Internet, it is vulnerable to a variety of attacks which can be classified into: (i) DNS “hacking” attacks: (ii) Routing table “poisoning” attacks: (iii) Packet “mistreating” attacks: (iv) Denial of Service (DoS) attacks:

We focus our research attention to the routing table “poisoning” threat. The majority of work on routing protocols for the Internet has proceeded in two main directions: distance vector protocols (e.g. RIP [2]) and link state protocols (e.g. OSPF [3]). In case of link state routing protocols, a router either *proactively* sends malicious updates or remains *inactive* when the link state of the malicious router has changed. The problem falls under the category of path vector related attack problems. Since the attack is on the Internet infrastructure, therefore the solution needs to be scalable.

In this paper, we develop a scalable method (called SLIP) for detecting router attacks in link state protocols. The primary capability of SLIP is to detect malicious LSA sources, which is not possible in [3, 4]. In addition, the protocol has the capability to locate the malicious source within a small bounded set of nodes.

*Dept. of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011.

2 Secure Link State Protocol (SLIP)

In this section, before describing the SLIP protocol, we mention the assumptions and fault model used for the protocol.

Fault Model and Assumptions

Our research focuses on securing link state protocol, based on the following assumptions: (i) We assume an undirected network. (ii) Both the nodes supporting an edge can identify the change in the link status. (iii) The malicious nodes are subject to Byzantine [6] as well as fail-stop type of faults. (iv) Whenever a node identifies that the link status of one of the links incident on it has changed, it sends a *Link State Advertisement (LSA)* to all the nodes in the network.

Description of SLIP

The protocol consists of two steps:(a) *Consistency Check* and (b) *Synchronization*.

Consistency Check: This procedure is performed to validate the routing updates at each receiving router. The validation involves standard validation checks such as authentication and sequence number [3] comparison. In addition to the standard checks, a *delayed routing update check* is carried out which results The *delayed checks* involve the following steps:

1. Change of state of each link is identified by both the nodes *supporting* the link.
2. When a routing update (LSA) arrives, the receiving node checks whether there is any change in the link state.
3. If a node x reports that the cost of link has changed, the receiving router checks whether the other node supporting the link has also notified the change. If the other node has notified and if the change matches, then the receiving router accepts the change. If they do not match, then the update is considered to be a malicious update and the nodes supporting the link are put into the Malicious List (M). Otherwise, it starts a timer.
4. If the timer expires, then the update is considered to be malicious and the nodes supporting the link are put into the Malicious List (M).

Synchronize: The second step in the secured link state

protocol lies in the synchronization of the topology among the neighbors. The synchronization is carried out using the principle of voting used in N-Modular Redundancy (NMR) systems [5]. Following are the steps involved in the synchronization process:

1. Each node receives topology information from each of its neighbors.
2. The entry which is agreed by the maximum number of nodes (neighbors and the receiving node) is selected as the final link state information.
3. In case of tie, the entry present in the receiving node is kept as it is.
4. The nodes present in Suspicion Matrix are not considered for synchronization.

The proposed checking algorithm has the following properties: (i) The checking algorithm runs in $O(n)$, where n is the number of nodes in the network. (ii) The ratio of number of malicious nodes identified by the algorithm, to the actual number is bounded by:

$$\frac{|M_S|}{|M|} \leq 1 + \frac{\sum_{i=1}^{|M|} \alpha_i}{|M|} \quad (1)$$

where α_i is the degree of the i^{th} malicious node, M is the list of malicious nodes, and M_S is the set of malicious nodes returned by the checking algorithm.

3 Simulation Studies

In our simulation study, we capture the performance of SLIP against the traditional link state protocol. Simulation studies were carried out using NS-2 [7] using the following performance metrics:

Fault Detection Time (δ): δ is defined as the average time taken by any node to detect a fault.

LSA Confidence (λ): λ is defined as the probability that any node i works correctly in presence of malicious attackers. If m malicious nodes are uniformly distributed in a n node network, then $\lambda_{SLIP} = (1 - \frac{m \times (m-1)}{(n-1) \times (n-2)})$ and $\lambda_{LS} = (1 - \frac{m}{(n-1)})$.

Attack Detection Probability (α): α is defined as the probability that a malicious update is detected. It can be shown that if there are m malicious nodes in a network having n nodes with degree r , $\alpha_{SLIP} = \frac{\binom{n-m}{r}}{\binom{n-1}{r}}$.

Out-of-Sync Parameter (ψ): ψ measures the probability that an entry in the routing table of a node is not synchronized with the rest of the network. This parameter is generally expressed per thousand entries.

Overview of the Results

The simulation results are summarized as follows. For details of the simulation, please refer to [8].

1. The fault detection time induced in SLIP is around 5–7% higher than that of traditional link state protocols, when the network is moderately dense.
2. Confidence exuded by SLIP is consistently higher

than that of normal link state protocol. The simulated results follow the analytical results closely.

3. In case of SLIP, α decreases with increasing number of malicious nodes in the network. In presence of 4 attackers, α is around 80%. The simulated results closely match the analytical results for α .
4. When the network is sparse (node degree ≤ 3.5), ψ_{SLIP} value is lower than that of link state. When the node degree is greater than 3.5, ψ_{SLIP} becomes at most 5% higher than that of link state.

4 Conclusions

In this paper, we proposed an elegant solution to the routing table poisoning threat in link state protocol, under certain fault model. Our scalable secure link state protocol (SLIP), is based on the principle of suspicion such that a node does not believe a link state update until and unless it receives confirmation from the other node supporting the link. SLIP also includes a synchronization procedure to provide synchronization for the updates. We carried out analytical and simulation studies to evaluate the proposed SLIP for two performance metrics, viz. attack detection probability and fault detection time, under different network and attack scenarios. Our studies showed that the SLIP offers a very high attack detection capability with a little degradation in fault detection time compared to the link state protocol. As part of our future work, we identify the following areas: (i) To extend the SLIP protocol to asymmetric networks. (ii) To develop efficient attack recovery techniques.

References

- [1] A. Chakrabarti and G. Manimaran, "Internet Infrastructure Security: A Taxonomy," *IEEE Network*, vol.16, no.6, Nov./Dec. 2002.
- [2] C. Hendrik, "Routing Information Protocol," *RFC 1058*, June 1988.
- [3] J. Moy, "OSPF Version 2," *RFC 1583*, March 1994.
- [4] S. L. Murphy and M. R. Badger, "Digital Signature Protections of OSPF Routing Protocols," in *Proc. SNDSS*, 1996.
- [5] Arun K. Somani, Vinod K. Agarwal and David Avis, "A Generalized Theory for System Level Diagnosis," *IEEE Trans. Computers*, 38(5), pp. 538-546, 1987.
- [6] L. Lamport, R. Shostak and M. Pease, "The Byzantine General's Problem," *ACM Trans. Prog. Languages and System*, vol. 4, no. 3, pp. 382-401, Apr. 1982.
- [7] UCB/LBNL/VINT Network Simulator - ns (version 2), Available at www.isi.edu/nsnam/ns.
- [8] A. Chakrabarti and G. Manimaran, "A Scalable Method for Router Attack Detection and Location in Link State Routing," *DCNL Tech. Report*, Oct 2002.