Malicious MPLS Policy Engine Reconnaissance

A. Almutairi[1] *and S. Wolthusen*[1,2]

[1]**Information Security Group**
Royal Holloway, University of London, UK

and

[2]**Norwegian Information Security Laboratory**
Gjøvik University College, Norway

stephen.wolthusen@rhul.ac.uk

16 September 2013

Contents

1. Introduction
2. Simulation Design
3. Analysis and Results
4. Probability
5. Conclusions and Future Work

Introduction

Multi-protocol Label Switching (MPLS) is not only widely deployed in backbone networks for fast decision routing, but also for guaranteed *quality of service (QoS)* that is crucial for critical infrastructures (e.g financial services, electric power grid)

- Unlike dedicated legacy point-to-point links legacy, MPLS is a **policy-based** system that provides different services for different traffic flows that share network resources.
- Service providers may not wish to reveal policies for business and security reasons

Adversaries are known to conduct network reconnaissance before launching practical and deniable attacks

MPLS Policy Engine

MPLS policies may vary depending on the required services and operation. However, they can be split into two main categories:

1. **Traffic Policies:** Policies governing the operations of MPLS nodes on traffic as per packet by packet e.g the label operations (push, swap, or pop)
2. **Label Policies:** Policies related to management of labels inside the MPLS domain

Label allocation Independent label allocation or ordered control mode

Label distribution Unsolicited mode or downstream on demand mode

Label retention Liberal retention/ conservative retention mode

Related Work: BGP Policy Reverse Engineering

A *reverse engineering* approach can be used to reveal policies. Research in this area has been largely limited to the exterior Border Gateway Protocol (BGP):

- Machiraju and Katz (2006) as well as Liang *et al*. (2011) showed the ability to reveal local preferences in BGP routing policies by examining the BGP updates
- Work by Wang and Gao (2003) to infer the route preference that influence route selection in import policies as well as the export policies that is used for controlling the inbound traffic
- Gao (2000) described a heuristic algorithm that extracts business relations of Autonomous Systems (ASs) from BGP routing tables

Related Work: BGP Policy Reverse Engineering

- Tool developed by Siganos and Faloutsos (2004) to infer business relationships of ASs by parsing and restoring the information found in Internet Routing Registries (IRRs) in easy relational database
- Work by Ming *et al.* (2008) to reveal the taken actions by certain ASs in response to false announcements in false Multiple Origin AS (MOAS) Event using BGP updates

Malicious Reconnaissance: Research Hypotheses

The main signalling protocol in MPLS is the Label Distribution Protocol (LDP). Analysis of the control messages in MPLS domain can help in identifying the implemented policies

Hypothesis 1: Policies Traces

Is it possible to efficiently identify the used policies for labels inside MPLS domain by examining the traces that they leave (e.g signalling messages)

Hypothesis 2: Probability

Is it possible to efficiently determine the probability of the implemented policies based on limited or zero knowledge about policy traces

Research Problems

Validation particularly for hypothesis 1 was partly performed using simulation:

- For reproducibility, we designed a **simulation** network for MPLS and analysed the traces each policy leaves in response to an identified probing process

  We subsequently we analysed the ability of an adversary with limited ability to reveal information about the policies used — adversary model based on Almutairi and Wolthusen (2013a)
- Building on this, we employed a **Bayesian Network** for addressing the probability issues for hypothesis 2

Simulation Design: State Space Reduction 1

As there are many plausible policies service providers or network operators can use, we restrict our simulation to basic policies that govern the management of labels inside MPLS domain. These are:

Label distribution Unsolicited vs. downstream on demand mode

Label allocation Independent label allocation vs. ordered control mode

Label retention Liberal vs. conservative retention mode

Note that owing to the the limitation of our simulation tool the label retention policy was addressed only in the probability analysis

Simulation Design: State Space Reduction 2

For validation purposes, we combine policies with identical external (observable) trace results

1. When implementing independent + downstream on demand mode, MPLS node will answer a requested label binding immediately without waiting for label binding from next hop
2. When implementing independent + unsolicited mode, MPLS node will advertise label bindings to its peers whenever it is prepared to label switch the recognised traffic flow
3. When implementing order control mode, MPLS node must only issue a label mapping after receiving a label mapping from the egress node

Royal Holloway University of London

HØGSKOLEN I GJØVIK

Information Security Group

Simulation Design: Final State Space

For simulation purposes and to limit paper extent, we hence consider only three policy states which we denote by S — a full MPLS policy model would normally consist of dozens of states:

- The set of policy states are: **Independent unsolicited**, **independent downstream on demand**, and **ordered control mode**

Reduced state space model

Each state s is an element of set S as $s \in S : IU, ID, OC$ where independent unsolicited, independent downstream on demand, and ordered control mode are represented by IU, ID, and OC, respectively

Adversary Model

A more **restricted** adversary is needed in MPLS networks where protocol adversary models (e.g. **Dolev-Yao**), in general, assume the worst case scenario.

- Adversary Goals and Motivations: The adversary aims to reveal **policy engine state** of the MPLS nodes
- Adversary Knowledge: We assume the adversary has complete knowledge of the **topology** of the attacked network, also, the used **labels**
- Adversary has access to at most one of the **core links** with read, write, and intercept operations

Probe Elements

To explore a policy state engine and its state, some (minimal) active probing may be required:

LDP Withdraw Message

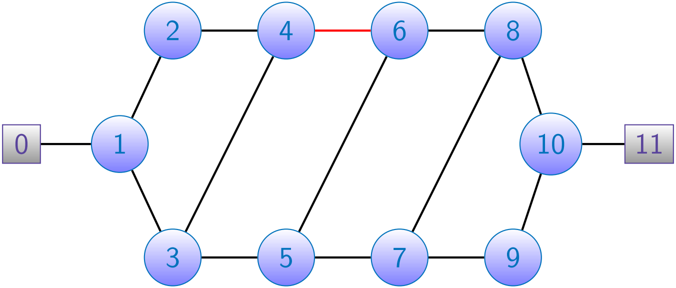
We decided to use LDP messages, particularly, the withdraw message for our probing process.

It should be noted that LDP messages have a common structure that uses type length value (TLV) encoding scheme which would typically include the path vector TLV and hop count

Simulation Design: MPLS Network

NS-2 was used for our simulation which also simulates MPLS

- The simulated network is composed of two customer edges ($node_{0,11}$) and MPLS domain ($node_{1,...,10}$) where $node_{1,10}$ represent MPLS ingress and egress edges, respectively



Simulation Scenario

Our scenario was to send a withdraw message for a label belonging to a traffic passing through the concerned MPLS nodes ($\text{node}_1, \ldots, \text{node}_6$), which would propagate towards the ingress edge due to the cooperative nature of LDP

Hence, we can learn the policy of the upstream LSRs based on observing responses (via label mapping procedure)

Observed Traces

Each one of the concerned states responded differently to the withdraw messages as following:

- **Independent Unsolicited (IU):** Label mapping for the withdrawn label will be sent independently from the affected LSRs
- **Independent Downstream on Demand (ID):** Label request for the withdrawn label will be sent from the affected LSRs
- **Ordered Control (OC):** Label request for the withdrawn label will be sent from the ingress LSR towards the egress LSR

However, we sought to determine the ability of our adversary to reveal MPLS policy states based on observed traces

Policy Recovery Algorithm

```
Require: $LDP$ message $LDP_m$ on the compromised link
Ensure: Policy states $S$ of upstream LSRs
    $S = 1, ..., n$ where $n$ is the number of upstream LSRs
    if $LDP_m = REQ_l$ then
        if $Req_l$ is initiated by ingress LSR then
            for all $i \in S$ do
                $i = OC$
            end for
        else parse the TVL;
            for $j \in TVL$ do
                $S[j] = ID$
            end for
        end if
    end if
    if $LDP_m = MAP_l$ then
    parse the TVL entries;
        for $j \in TVL$ do $S[j] = IU$
        end for
    end if
    return $S$
```

Key Results

Because there is no direct connection among the peer MPLS nodes, our adversary could use the **TLV structure** to reveal which node(s) the label mapping or request has propagated through:

- In the simulation, the adversary discovered only the two nearest nodes state (node_4, node_5) when ID policy is used
- Only the node directly attached to the compromised link with IU policy was revealed reliably by our adversary
- Nodes with OC policy have been fully revealed by the adversary

Adversary Model Modification

So far our scenario has imposed hard **boundaries** on adversary where adding more power to the adversary, particularly, the number of links that adversary has access to, would enable the adversary to reveal more policy states in MPLS nodes

Relaxation of Adversary Limitations

More links have to be compromised (in the worst case $\frac{n}{2}$ links where n denotes the number of MPLS nodes of concern) to reveal almost all policy states in MPLS network

Probability: Bayesian Network 1

Knowledge gained from our scenario and other specification information about different operation policies in MPLS networks can be represented in **Bayesian Networks (BN)** to give approximate estimation about MPLS nodes policies based on limited information

- We construct BN to represent policy states (IU, ID, OC) as well as the retention policy and **traces** found on the simulation using Bayes' rule:

$$p(h|e) = \frac{p(e|h).p(h)}{p(e)}$$

Probability: Bayesian Network 2

Our **BN** has three nodes where the root node S has three values (IU, ID or OC) and the leaf nodes under the root node represent the other policies which is the retention mode (R) that would be associated with the MPLS state and the traces observed on MPLS simulation, particularly, label allocation (L):

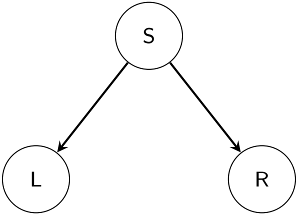


Figure : Bayesian network model

Probability: Bayesian Network 3

The prior probability of each root node is $\overline{0.3}$ as per the following equation:

$$p(S) = p(S = IU) + p(S = ID) + p(S = OC) = 1$$

- Each leaf node is associated with a conditional probability table (CPT) where:
  1. The leaf node R has two values, "Conservative" and "Liberal"
  2. The leaf node L has two values, "Label Assignment" and "Request"

Conditional Probability Tables (CPTs)

Table : CPT for node R

State	S=IU	S=ID	S=OC
Conservative	0.5	0.9	0.5
Liberal	0.5	0.1	0.5

Table : CPT for node L

State	S=IU	S=ID	S=OC
Label Assignment	1	0	0
Request	0	1	1

Royal Holloway University of London

HØGSKOLEN I GJØVIK

Information Security Group

MPLS Policies State Probability Table

State	Retention Mode	Label Assignment	Probability
IU	Conservative	Label Allocation	$\overline{0.3} \times 0.5 \times 1 = 0.1\overline{6}$
IU	Conservative	Request	$\overline{0.3} \times 0.5 \times 0 = 0$
IU	Liberal	Label Allocation	$\overline{0.3} \times 0.5 \times 1 = 0.1\overline{6}$
IU	Liberal	Request	$\overline{0.3} \times 0.5 \times 0 = 0$
ID	Conservative	Label Allocation	$\overline{0.3} \times 0.9 \times 0 = 0$
ID	Conservative	Request	$\overline{0.3} \times 0.9 \times 1 = 0.3$
ID	Liberal	Label Allocation	$\overline{0.3} \times 0.1 \times 0 = 0$
ID	Liberal	Request	$\overline{0.3} \times 0.1 \times 1 = \overline{0.03}$
OC	Conservative	Label Allocation	$\overline{0.3} \times 0.5 \times 0 = 0$
OC	Conservative	Request	$\overline{0.3} \times 0.5 \times 1 = 0.1\overline{6}$
OC	Liberal	Label Allocation	$\overline{0.3} \times 0.5 \times 0 = 0$
OC	Liberal	Request	$\overline{0.3} \times 0.5 \times 1 = 0.1\overline{6}$

Conclusions and Future Work

We have analysed restricted adversaries' ability to reveal policy engine states in MPLS nodes which is relevant to attacks on the quality of service of network flows as may be required for real-time protocols found in many CI such as financial services networks control systems including electric power networks

- This can be revealed with limited probing, relying on **Bayesian networks** to model incomplete information obtained over noisy channels

Ongoing work seeks to extend the policy model and states which may be capturable. We also are developing novel attacks aiming to degrade and disrupt MPLS flows both overtly and in deniable form

Royal Holloway University of London

HØGSKOLEN I GJØVIK

Information Security Group