Proof Of Concept for Opportunistic Security in MPLS Networks

Salil Ajgaonkar B.E

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Future Network Systems)

Supervisor: Stephen Farrell

August 2018

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Salil Ajgaonkar

August 27, 2018

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Salil Ajgaonkar

August 27, 2018

Acknowledgments

...ACKNOWLEDGMENTS...

Salil Ajgaonkar

University of Dublin, Trinity College
August 2018

Proof Of Concept for Opportunistic Security in MPLS Networks

Salil Ajgaonkar, Master of Science in Computer Science

University of Dublin, Trinity College, 2018

Supervisor: Stephen Farrell

Multiprotocol Label Switching (MPLS) is a high performance packet switching technology which supports packet switching for multiple protocols and access technologies. It is a switching technology that redirects data based on short label paths rather than an IP table lookup. Each Label Switched Router in the MPLS network has this table that tells the router how to handle and redirect specific types of data. This helps the router to handle data in a consistent fashion while maintaining flexibility and congestion control at the same time.

MPLS provides networking corporations and government high speed and reliable data transfer over geographically dispersed sites as well as maintaining flexibility and high bandwidth data exchange.It is very easy to scale and can be efficiently tuned to meet service level agreements. it is primarily used to provide Virtual Private Network (VPN) capabilities for corporations who wish to transfer private data over a public network.

Such heavy reliance on MPLS by big companies and the general public to manage their data means that any forms of successful attacks often have the potential to disrupt vital everyday operations. Attacks ranging from data theft to denial of service if successful may lead to severe damages to the corporation as well as the owner of the data.

Earlier work conducted by the Internet Engineering Task Force (IETF) have analyzed the various vulnerabilities and security threats that plague the MPLS technology and have drafted a potential resolution to some of the security issues. The work proposed the implementation of Opportunistic Security in the MPLS network using payload encryption to encrypt the underlying application data and protocols using secure key exchange between end to end or hop by hop Label Switching Routers (LSR's) on an MPLS Label Switched Path (LSP).

This paper aims to provide a proof of the concept to this proposed solution by implementing it on a practical controlled environment and measuring its effects on the overall functionality and feature of the presently running MPLS technology. The aim of this paper is to provide an idea regarding the feasibility of this solution in practical commercial application in the world.

Summary

MPLS is a network routing technology which uses circuit switching and packet switched network technologies to support data routing for multiple internal protocols. It is a protocol independent and highly scalable technology whose flexibility enables efficient utilization of network resources resulting in high quality of service at low cost. MPLS relies on the use of fixed bytes of data that represent as labels for routing decisions. These labels are pushed on top of a data packet at the ingress switch of an MPLS cloud. All the succeeding Label Switching Routers (LSR's) in the MPLS cloud refer these labels to appropriately route the data to the correct destination. The path to the destination is pre-configured in the LSR's Label lookup table rather than an IP routing table. This avoids any complex lookups or reading long network addresses and implementation of longest match algorithms. Once the MPLS packet reaches the egress switch of the MPLS cloud, the MPLS label is popped off and the data packet is then transmitted as a regular IP packet along the succeeding switches. The LSRs between the ingress and the egress switch of the MPLS cloud also have the capability of pushing and popping off the MPLS labels on the data packet to efficiently engineer the network traffic.

Networking corporations and service providers can leverage this flexibility to scale their network services. Using the advantages of MPLS they are able to run high bandwidth applications over seamless IP based networks that connect multiple, remote site [2]. The reliance on MPLS VPNs by corporations and government agencies means

that attacks ranging from intercepting sensitive data to disrupting data, voice and multimedia services can significantly impact vital operations [2].

Data Security in MPLS previously mainly relied on Data security (e.g., confidentiality) in MPLS has previously relied on just two features: 1) Physical isolation of MPLS networks has been used to ensure that interception of MPLS traffic was not possible. 2) Higher-layer protocol security such as IPsec has been used whenever a particular flow has determined that security was desirable [1]. However these features have a number of vulnerabilities. The network is still vulnerable to network taps between links, misconfiguration of routers, data replication, as well as users might not enable end to end security of their applications [1].

To mitigate these vulnerabilities to some extent, the IETF drafted an Internet-draft titled "Opportunistic Security in MPLS Networks draft-ietf-mpls-opportunistic-encrypt-03" which proposed a novel idea of implementing Opportunistic Security (OS) in the currently existing MPLS implementation. The Internet draft suggested the implementation of opportunistic encryption of data packet payload which is held by the MPLS packet. As the LSR's only rely on the information contained in the MPLS labels for routing information the contents of the MPLS packet are of no significance for the flow of traffic. Thus end-to-end or hop-by-hop encryption of the data payload of the MPLS packet by the LSRs can contribute greatly to maintain the confidentiality, integrity and authenticity of the data that flows through the MPLS network.

This paper aims to implement this proposed experimental idea in a controlled environment by simulating the flow of traffic in an MPLS network in a virtual network and comparing the difference in performance between MPLS with the OS and MPLS without the OS. This will help validate or invalidate the proposed solution experimentally and further spread more light on its impact and feasibility on real world implementation.

This experiment will be carried out using an experimental network setup consisting

of mininet as a virtualized network platform. On this virtual platform we will setup OpenVswitch (OVS) virtual switches which will simulate the flow of network traffic through the virtual switches. The MPLS flows needed to configure the OVS switches will be configured onto the switches using Software Defined Networking (SDN) Technology. The SDN is enabled using the OpenFlow communication protocol. OpenFlow configures these OVS switches to simulate the behavior of an MPLS switch. The Opportunistic Security feature in the MPLS will be implemented by coding this new feature into the existing OVS code base which would then be configured onto the OVS switches in the mininet network to simulate the behavior of the system with the OS. We would then evaluate the performance of the system by measuring certain network performance metrics and comparing them with the ones measured without the OS. This comparative analysis should give us a good idea of how the OS in MPLS fairs against MPLS without OS, if there are any room for improvements and its potential impact on practical implementation.

Contents

List of Tables

List of Figures

Chapter 1

State Of The Art

Let us first understand what is MPLS and itsunderlying functionality.

1.1 Multi Protocol Label Switching (MPLS)

In order to better understand the MPLS functionality let us first understand how traditional IP routing works.

1.1.1 IP routing

IP routing is a destination based routing protocol. This means that the switches inside a network determine where to forward the IP packet based on the destination information stored in its IP header. The IP packet is forwarded from one switch to another in a hop-by-hop basis till it reaches its destination. Interior gateway protocols (IGPs) such as routing information protocol (RIP) and open shortest path first (OSPF), or exterior gateway protocol (EGPs) such as border gateway protocol (BGP) help route the IP packets from switch to switch[25]. Using these protocols the IP packets are routed through the network without any pre-determined path. All the IP packets with the same destination may flow through different paths to reach their detonation. That is the reason why they call IP routing protocol as a connectionless protocol. Routing protocols, such as Open Shortest Path First (OSPF), enable each router to learn the topology of the network. The routers build forwarding tables using the information provided by routing protocols [25]. A switch references this forwarding table when it

analyzes an arriving packet to decide which is the next switch in the network that can bring the IP packet lose to its destination.

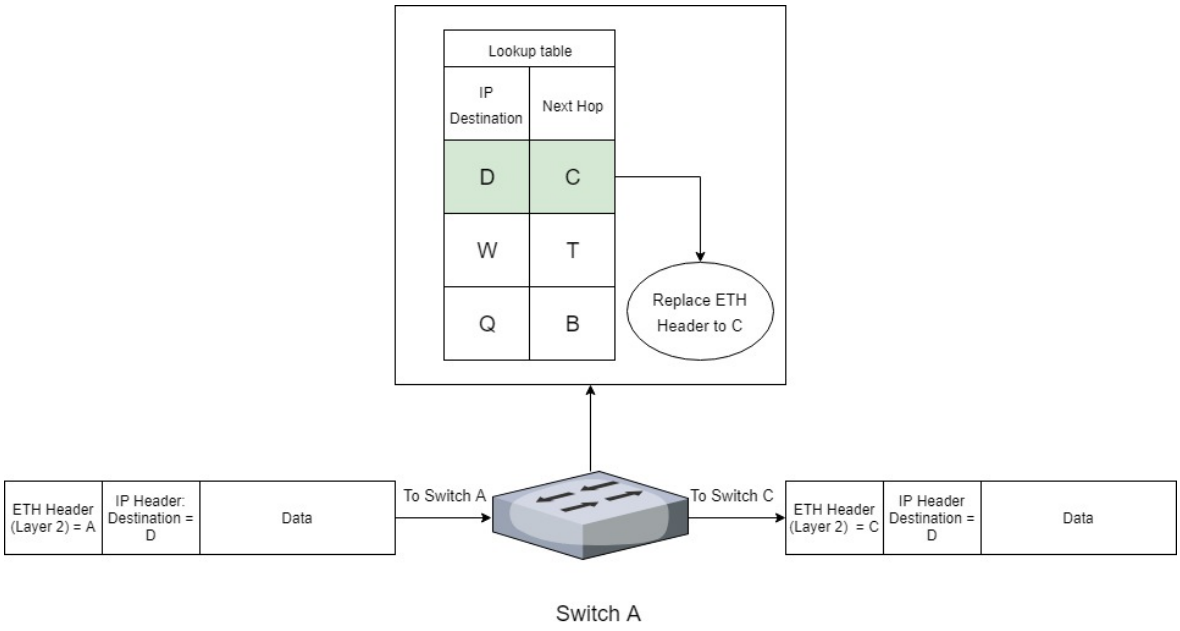

Figure 1.1: IP Routing

The information regrading the IP packets destination is located in the Layer 3 header of the IP packet. The switch has to use this information to compare it with the lookup tables. The switch then replaces the Layer 2 header of the IP packet so it can be routed to the next switch as determined by the lookup. [Refer Figure 1] an IP packet with destination D arrives at switch A. Switch A analyses the IP header and finds the destination to be D. Switch A then refers to its lookup table to find the next hop for all packets destined to D. The next switch based on the IP lookup is C whose information is then stored into the ETH header (Layer 2) of the packet by switch A. Switch A then routes the packet to switch C which would eventually lead to its destination D. All the switches follow this same process as the IP packet traverses through the network to reach its destination. The routing algorithms need not necessarily follow the shortest path logic to deliver the IP packet to its destination. Certain protocols also have the functionality to make decisions by considering the bandwidth occupied in between links to determine the best and fastest path to deliver packet.

1.1.2 MPLS routing

MPLS Routing is a routing technology that leverages the benefits of both packet switching technologies as well as IP routing technologies. IT is a data plane protocol most widely used in core network systems for high speed data transfer. MPLS is an IETF standard approach to integrate the best attributes of traditional layer 2 and layer 3 technologies. It defines a set of protocols and procedures that enable the fast switching capabilities of ATM and frame relay to be utilized by IP networks [25].

The main concept of MPLS routing is that instead of referring the IP header destination information for routing, the switches have the functionality to push or pop certain additional routing information onto the data packets in the form of labels. The switches refer these labels to appropriately route the packets to the desired destination. The Label information mapping is based on existing IP routing technologies.

Switches having the functionality to push and pop labels together form an MPLS network. Routers in this network are specially called Label Switch Routers (LSRs). Routers at the edge of the MPLS network are responsible for communicating with the external traditional IP routers. These Routers are named as Label Edge Routers (LERs). The LERs are responsible for attaching the very first labels on top of the data packet that newly enters into the MPLs network. The label information is based on Forward Equivalency Class (FEC). Packets are then forwarded through the MPLS network, based on their associated FECs, through swapping the labels by routers or switches in the core of the network called label switch routers (LSRs), to their destination [25].

The mapping of Label values to their respective path is stored in the Label Information Base (LIB) of every LSR in the MPLS network. Each LSR builds it LIB when it is first initialized when the network is established. The LIB is similar to a IP lookup table with the difference that instead of reading long network addresses the LSR refer these short label values which give a granular control over a packets path in the MPLS network, which is aptly named a Label Switch Path (LSP). LIB is typically established in addition to the routing table and Forwarding Information Base (FIB) that traditional routers maintain [25].

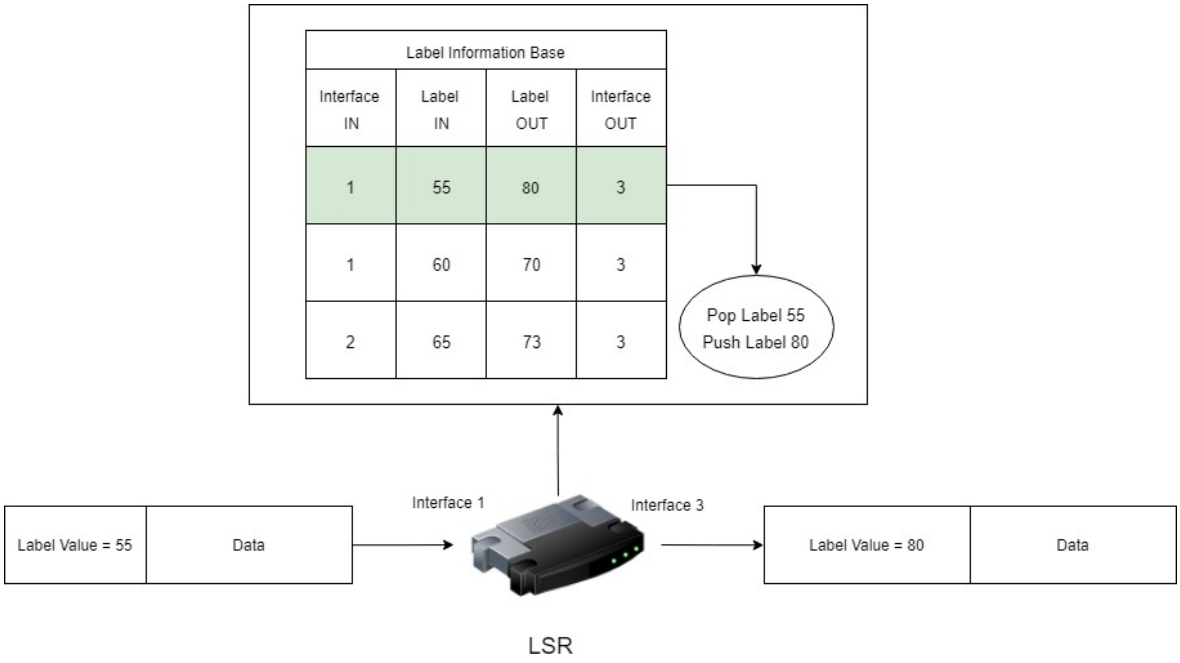

Figure 1.2: MPLS Routing

[Refer Figure 2] a packet with a label value of 55 enters an LSR on its interface 1. The LSR then reds the Label value and refers its LIB too find out which LSP to forward this data packet. The LSR can also pop the Label value an insert a new value on the data packet if the LIB of the next LSR expects the new value. the next LSR will follow the same procedure which eventually routes the packet to the LER on the other end. This LER pops the last MPLS label forwarding the packet as the original data packet onto the next traditional router which lies outside the MPLS network.

1.1.3 Components of MPLS

For MPLS to correctly route the data in its network it relies on certain special components to bring about the special functionality. Some of which are as follows.

Forward Equivalency Class (FEC)

FEC is a class of packets which are routed along the same LSP in the MPLS network. A FEC can include all packets whose destination address matches a particular IP

network prefix, or packets that belong to a particular application between a source and destination computer [25]. When a packet reaches an LSR, the LSR classifies the packets into an FEC based on its destination address, Quality of Service, the interface on which it arrived among other criteria. FECs are usually built through information learned through an IGP, such as OSPF or RIP [6].

Label Configuration

MPLS introduces it own header information termed as an MPLS Label or just Label. The MPLS label is a 32 bit (4 Bytes) fixed length, contiguous identifier used to denote the FEC of the packet. Just like the IP header in the IP routing mechanism, the MPLS header has all the information required to forward the data packet in the MPLS network. [Refer diagram 3] for the structure of the MPLS Label. The MPLS label sits between the ETH header and the IP header. This is the reason why they say MPLS is a layer 2.5 protocol.

[Refer diagram 4] for the MPLS Label structure. The MPLS Label consists of the following parts.

Label

The Label consist of an id that identifies the FEC of a particular Data packet. The LSR refer this value to determine the FEC of a data packet that arrives at its interface. This FEC values determines the LSP that the data packet will take from the current LSR.

Traffic Class (TC)

The TC is primarily used to denote the Quality of Service implementation. This value along with the Label value can influence the LSR in selecting the LSP.

Bottom of the stack (S)

This is a simple flag that help identify whether the MPLS header is the last MPLS header in the MPLS stack. If the current label is the only one in the stack then the value is set to 0 else it is set to 1.

Time to Live (TTL)

This value denotes the number of hops the data packet has taken while traversing through the MPLS network. This value helps identify whether the packet was routed through the expected number of router hops or if some attacker has manged to re-route the packet via some malicious LSRs. The value is copied from the packet header and copied back to the IP packet header when it emerges from the Label Switched Path [25].

MPLS by itself is vulnerable to a number of security threats. This is primarily because MPLS in its primary idea aims to solve the problem of high speed data delivery over large geographical network while maintaining flexibility so it can be scaled to meet business requirements. Routing and traffic engineering using MPLS is very easy and service providers can leverage this benefit to provide varying levels of Quality of Service.

With the increasing deployment of MPLS, security of data passing through the MPLS network has become a grave concern for corporations and the service provider alike. Service providers are concerned with the security and confidentiality of their customers data while corporations using MPLS to share data between their geographically distributed sites demand authentication and integrity as well.

1.2 Security Requirements in MPLS

Some of the General Service Provider Security Requirements are as follows:

Protection of data at the Data-Plane

Encryption of data is not provided as a basic feature in all telecommunication protocols. Protocols like IPSEC have all the features of authentication, data integrity and confidentiality however it is not widely adopted by all applications. however, roughly 30% of web sites actually take on this burden of implementing IPSEC, even though the software required is ubiquitous [2].

Protection from attacks on Label Distribution protocol

In [16] the authors have stated that attacks on Label Distribution protocol (LDP) exploit 3 weaknesses: The LDP specification, Service provider implementation and underlying infrastructure. The authors have expressed that these attacks can lead to various DOS attacks or Route modification attacks most of which ma lead to violation of SLA for the Service provider. Naturally the Service provider would like protection against such attacks as a major requirement in implementing MPLS networks.

Prevent Malicious External Controllers from mis-configuring the SDN switches

SDN Switches are vulnerable to being mis-configured by external controller. An attacker can send malicious control-plane messages to the switches which can mis-configure them to send packets to a malicious switch, replicate the packets or drop the packets in general to trigger a Denial of service attack (DOS). Service providers expect some form of authentication of messages received from the control plane by the switches to make sure any control-plane messages received by the SDN are from a legitimate controller in the network.

Prevention of attacks that spoof IP addresses

Many attacks on protocols running in a core involve spoofing a source IP address of a node in the core (e.g., TCP-RST attacks).[3] If such a spoofed IP address gets accepted in the MPLS network, the MPLS switches can route sensitive packets to the spoofed address leading to data leakage. The attacker can then have the luxury to analyses and read the data at his leisure till the systems identifies the spoofed address.

Hiding Service Infrastructure

In general the service provider would like to hide his service infrastructure from the external network. An MPLS/GMPLS provider may make its infrastructure routers unreachable from outside users and unauthorized internal users. For example, separate address space may be used for the infrastructure loopbacks [3]. A service that is hidden

to the external network has less chances of being targeted by attackers.

Protection from mis-merging of LSP

Care needs to be taken that any implementation of security procedures do not alter the the MPLS Label stacking logic which then becomes vulnerable to mis-merging of LSPs. LSP mis-merging has security implications beyond that of simply being a network defect. LSP mis-merging can happen due to a number of potential sources of failure, some of which are due to MPLS label stacking [17].

Link Authentication

Service providers would prefer to authenticate a site before linking a connection. this helps validate the site based on certain security protocols like IPSEC. If the user wishes to hold the authentication credentials for access, then provider solutions require the flexibility for either direct authentication by the PE itself or interaction with a customer authentication server [3].

Security Considerations in Operations, Administration, and Maintenance messages

Operations, Administration, and Maintenance (OAM) messages are messages that are used for the internal functionality of the MPLS switches. OAM messages help in monitoring devices and implementing data transport mechanisms on a network level. They are responsible for the overall performance of the network device. On a service-oriented functionality they provide monitoring services to end users which is vital to keep a track of the performance so as to make sure the SLAs are met.The nature of OAM therefore suggests having some form of authentication, authorization, and encryption in place. This will prevent unauthorized access to MPLS-TP equipment and it will prevent third parties from learning about sensitive information about the transport network [18]

Meeting these requirements can easily fail if the system in place is vulnerable to any security threatening attacks. With the intention of securing these vulnerabilities we first need to analyze what are the different forms of attacks and how we can resolve

them. A number of research has been conducted in analyzing the various security threats that plague the MPLS networks.

1.3 Security threats in MPLS

[1] has analysis a number of security threats to MPLS in VPN. In this work the authors have discussed the principal security issues in MPLS related to Network separation, Inter-provider connectivity and MPLS packet labeling. In [2] the authors have stated that security in MPLS previously relied only on the physical isolation of the MPLS network and High-layer protocol security like IPSec. The authors have also gone further as to why these features fall short of being the ideal solutions to current network attacks.In [3] the authors have elaborated on specific types of exploits that threaten the MPLS/GMPLS network. The authors have segregated the attacks in the form of attacks on the data plane, attacks on the control plane, attacks on the operation and management plane and insider attacks. they have also recommended defensive techniques for MPLS against these forms of attacks. In [2] the author has suggested a mechanism to enhance the security in MPLS networks by using multi-path routing using a threshold secret sharing scheme. In [8] has elaborated the security issues that are inherent in the MPLS architecture. The authors state that some routers may implement security procedures relying on certain headers being in fixed place relative to a certain layer header in the IP stack. Another security issue mentioned was that the MPLS routers agree on the meaning of the labels and thus work upon a chain of trust to transfer data in the network and that if packets from untrusted sources are accepted then they may get routed illegitimately.

In this section we will try to list out the various security attacks that the MPLS network is vulnerable to. The Security attacks have been categorised into 4 types: Attacks on the Data Plane, Attacks on the Control Plane, Attacks on Operational and Management of the MPLS Network and Insider attacks.

1.3.1 Attacks on the Data Plane

Attacks that are mainly aimed at the User's or the Service Provider's data are categorized into Data-plane security attacks. These attacks aim to either manipulate the

data flowing through the MPLS network, delete the data flowing through the MPLS network, inject malicious data or just plain observe data unauthorized data all with malicious intent.

Plain IP forwarding

MPLS Switches can forward un-labelled IP packets as normal IP packets if they are configured to do so. An attacker already inside the core can exploit this information to reach other core devices compromising them. It is difficult to manage this type of attack by just traffic engineering or implementing any VPN service.

Forwarding captured packets from the core to the outside network

An attacker inside the core can capture the data sensitive packets and forward them to any destination he desires even if IP traffic is not being forwarded. This can be done by encapsulating the captured packet in the payload of a UDP packet. The source IP address can be spoofed and the detination IP address can be set to wherever the attacker may wish to forward the packet. This however is only possible if the attacker is already aware of the Labels needed to route through the LSP.

Sniffing of Data

Data sniffing can be explained as the action of capturing data packets and analyzing its contents to understand what they contain. Private corporations and Users in general often transfer confidential and sensitive data over the internet. If this data is not sufficiently encrypted before forwarding them onto the internet then an attacker can use the wide variety of packet sniffing tools to sniff these data sensitive packets and use the sensitive information in them for malicious purposes. Unauthorized packet sniffing can also be a first step in other attacks in which the recorded data is modified and re-inserted, or simply replayed later [3]. Unauthorized packet sniffing is one of the most commonly faced security issue in any network system including and especially the MPLS network considering its wide spread use for mission critical systems and reliance of both common users as well as Multi national corporations.

Modification of Data Packets

If an attacker is able to manipulate the contents of the data passing through the MPLS network then he has the potential to initiate a wide range of security attacks on the network. Though not as easy as it sounds the attacker must first be aware of the internal configurations of the MPLS netowork which more often than not is more difficult than the actual manipulation of packets. MPLS often has an "egg-shell" security model where it is very difficult to penetrate the internal network core, but once inside the attacker can cause a lot of damage to the service provider and the network service as a whole [1]. Some of the potential attacks that can be triggered by data manipulations are:

Route modification attacks

Once inside the network an attacker can manipulate the data flowing through the network and route them to any destination he desires, provided he has the information necessary to route the information correctly through the network. Route modification attacks enable an attacker to gain access to certain traffic (e.g., maneuver traffic through a compromised link); affect accounting (e.g., trigger automatic financial transactions among cooperating providers); or route traffic across domains (e.g., send one customers traffic to another customers network) [16].

1. Path Switching: A normal traffic flowing through the MPLS network ideally follows a fixed set path or to be more precise an Label Switched Path (LSP). The LSP is determined by the initial traffic engineering setup done by the service provider. The traffic engineering primarily relies on the label configuration of the packets. If an attacker were to modify this label information of a data packet, then he may be able to route the data packet that it was not intended to. Such a path is called a rogue path. By doing this the attacker can deceive the traffic engineering and reap benefits out of it. For example an attacker can forward his online gaming data packets via the live video conferencing path to get better speed advantage and disrupting the video conferencing quality of service.

2. Destination Switching: Just like Path switching, an attacker is also able to modify the destination of a packet if he modifies the label configurations appropriately.

This can fool the MPLS network into forwarding the packet to a rogue destination where the attacker can further conduct malicious operations on the packet which he wasn't able to do while inside the MPLS network.

3. Brute-Force Label Prediction: An attacker can deduce the LSP of an MPLS network if the destination address is known. He can target this address and pass data packets into the MPLS with an initial label value. He can then test out the Label with incremental values till he receives a reply from the destination address. The reply from the destination address can help him deduce the LSP for the destination address and thus reuse the label information to further pass data to the destination.

4. Brute Force Target Location: Similar to label prediction, the target can also try to identify what type of service lies at the end of an LSP, for example if the user were trying to identify if a web service lies at the end of an LSP then he can set the target tcp port to 80 and incrementally try out the IP address for a successful hit. This however is a very time consuming process considering the probabilities of getting the correct IP address compared to the total combination of IP addresses possible.

5. Forward Equivalency Class (FEC) Specificity Exploitation: When configuring the MPLS network Packets are configured in such a way that packets of similar type are routed through the same LSP. These packets are thus bound by the same MPLS label and routed through the same path as designated for labels of that value. Such classification of packets is termed as Forward Equivalency Class. This attack takes advantage of the most specific or longest match rule applied by ingress routers to incoming IP packets. An attacker needs access to a link or a connection to an interface to establish an LDP session. The attacker identifies a target FEC and advertises label bindings for more specific FECs. LSRs that receive the label mappings distribute them throughout the network, thereby building new LSPs toward the compromised link [16].

6. Label Mapping Messages Modification: An attacker can modify the Label values of a datapacket inside the MPLS network. He can thus reroute the traffic or create loops within the MPLS network. This modified message is sent on to the next

MPLS switch. When the upstream LSR receives a packet for the target FEC, it applies the incorrect label, which causes the downstream router to mistakenly recognize the packet as belonging to a different FEC. The packet is then forwarded along the desired LSP [16].

7. Address Messages Modification: Similar to label modification an attacker can redirect a data packet by spoofing the the destination IP address. This attack, also known as Fabricating Address Messages, reroutes traffic or creates loops by manipulating the least cost mechanism used to select the next hop. Traffic can be redirected using access to a compromised link adjacent to an LSR along a selected LSP. This modified message forces the LSR to adjust its local label information base and generate a Label Request message.Thus, a new LSP is constructed that forces the targeted traffic along the compromised link [16].

8. Label Edge Router (LER) label Modification: Label Edge Router (LER) are routers situated at the end of an MPLS network. These routers are the final routers in the MPLS network that pop the final MPLS Label off the data packet and forward the data packet its original form before entering the MPLS network. This attack requires access to a link along the path between VPN sites. Redirection of packets to a different site in the same VPN requires the attacker to know the routes and labels corresponding to that site. The attack is executed by modifying the topmost label of a transit packet (before the penultimate pop) to another label. If the new label is valid at the next hop, the packet is forwarded to a different LER [1]. This new LER may or may not be connected to the destination. It could either drop the packet as a whole which could lead to denial of service if the attacker modified to many of the packets or the LER may forward the IP packet as a normal IP packet, depending on its configuration.

9. VPN label Modification: VPN label modification is similar to the LER Label Modification. In this type of attack two VPNs are involved. One is the legitimate source VPN and the other is the destination VPN which is incorrect. The attacker captures and modifies the datapacket label of the source VPN and redirects it to the incorrect VPN. When this modified data packet reaches the LER, the LER redirects it to the incorrect VPN. If the LER doenot have a VRPN routin gand

Forwarding (VRF) Table then the LER may forward the packet as a regular IP packet. Given knowledge of routes and VPN labels in the network core, the combination of VPN label modification with LER label modification (previous attack) enables an attacker to redirect and/or drop any traffic passing through the compromised link [1].

10. VRF table Modification: A more serious issue is if the attacker is able to modify the VRF tables themselves. This would grant the user the ability to control the traffic from the LER. The attacker can change outgoing LER labels and outgoing VPN labels to impact QoS. More over, the attacker can control the routes taken by ingress VPN traffic and divert it to the wrong VPNs [1].

Data Insertion attacks

Data insertion attacks are type of attacks which involve insertion of malicious traffic into the network with the intention of making a switch accept the external data as valid data and forward the same to end devices. Insertion attack come in different forms with different intentions.

1. Insertion of malicious data traffic to Spoof and Replay: Spoofing refers to sending a user packets or inserting packets into a data stream that do not belong, with the objective of having them accepted by the recipient as legitimate [3]. Once a spoofed data is accepted an attacker constantly replay the accepted packets to incite the same response from the recipient even though the request is not authentic.

2. Insertion of malicious data using VPN Labels The attack is executed by fabricating packets labeled for the target VPN and corresponding egress LER, causing the egress LER to send the fabricated packets into the target VPN [1]. This type of attack is mostly implemented in MPLS networks providing VPN services. For this attack to be successful the attacker first needs to find a vulnerable LER which accepts external labeled packets. He should also be aware of the valid labels of the target VPN as well as the label information needed to route the packets to the target VPN.

3. Insertion of malicious data using LER Labels This attack is similar to the previous type of attack the only difference being that in this attack an attacker can ignore the insertion of the VPN label and just insert the malicious traffic with only the MPLS labels. When a packet with no VPN label reaches the egress LER, it is forwarded to a locally attached VPN site, or back through the network core to a remote VPN site, or to a core LSR. This 'bouncing' maneuver helps hide the source of an attack because packets appear to originate from the egress LER. An attacker still needs to have access to an LER that accepts external labelled data to perform this type of attack.

Denial-of-Service (DOS) Attacks

Denial of service is a type of attack where an attacker aims to make a target service unable to its users. This is carried out by disrupting the services of the service machine by bombarding it with large number of requests with the intention of overloading the system such that legitimate requests are not able to request for service. Banks, e-commerce websites and services which aim to provide high availability are particularly vulnerable to the damages caused by DOS attacks. There are plenty of ways an attacker can leverage the MPLS network to initiate a DOS attack on a service in the network.

1. Modifying the Community Attribute in LERs: In this attack the attacker modifies the VRF table that corresponds to the target VPN in an MPLS network that provides VPN services. The attack disrupts inbound and outbound traffic to and from the VPN site associated with the affected VRF table [1]. However to conduct this attack the attacker must first gain access to the LER which is connected to the target VPN.

2. Notification Messages Fabrication: An attacker who has access to a link in the MPLS network can fabricate false notification messages and release them in the MPLS network link. The attacker needs to have read and write access to this link. When an LSR receives this notification message it closes its LDP session thereby disconnecting the link. This prevents the flow of traffic through that LDP session and all packets destined to flow through that link are dropped, preventing requests from reaching the service. If an attacker has access to all the links of an

LSR then using the same form of attack he can close all the links of that LSR isolating it from the entire MPLS network.

3. Blocking KeepAlive Messages: Similar to the previous attack another way an attacker can close LDP links is by selectively blocking the LDP keep-alive messages which are periodically sent by the LSRs to continue the link session. This causes the LSRs to time out and close the LDP session preventing any traffic from routing through the LDP. This attack is tricky in the sense that identifying the cause of timeout can distract the network engineer from finding out about the attack.

4. Address Withdraw Messages Fabrication In this type of attack an attacker target three LSRs in an LSP. Lets assume the three LSR's involved in this example are A,B,C which are linearly connected in an LSP. The attacker fabricates an Address Withdraw message stating that address C has been withdrawn. He sends this packet to LSR A by gaining access to link A-B. LSR A withdraws the address associated with the interface for LSR C. LSR A now believes that LSR B cannot direct the traffic to C and thus forwards the traffic to C via other LSP congesting them. This congestion may eventually lead to a DOS attack.

5. Label Withdraw Messages Fabrication: This attack targets a specific LSP and requires access to a link along the target path. If the network employs label merging, then the attack also affects all upstream portions of paths merged with the target LSP [16]. The attacker fabricates a label Withdraw message and passes it along the LSP. the proceeding LSRs on receiving this message withdraw the label from their label information base. The LSP is thus destroyed and all traffic engineered to flow through this LSP now needs to be redirected along different LSP leading to congestion and eventual DOS.

6. Label Memory Exhaustion: This attack exploits LSRs with label retention mode. an attacker can flood such an LSR with Label mapping messages based on random FEC and label information. An LSR with Label retention mode will be forced to store these label information in its LIB which will eventually exhaust. The LSR is then forced to drop the older Label mappings to accommodate the new malicious Label mappings from the attacker which affects the legitimate mappings.

7. LSP Deletion. LSP deletion attack is an attack that involves the injection of malicious 'PathTear' messages from the Resource Reservation protocol that remove label bindings and resource reservations of targeted LSPs. A particularly insidious attack involves crafting a 'PathTear' message for a specific node in a targeted LSP; this message must be re-sent at least once per refresh period [19]. The receiving LSR deallocates all resources and configurations of the LSP and stops routing packets through that LSP leading to packets drops and denial of service. This attack is able to target an LSP individually. This helps in targeting attacks to an individual target rather than disrupting the whole system to initiate a DOS attack. This helps prevent any warnings from raising in the network and the attack may go undetected [19].

1.3.2 Cross domain attacks

An MPLS Service provider has to allow a customer edge router to send data into the MPLS network which can be further routed in the MPLS network. An attacker can exploit this cross-domain interaction to initiate attacks on the MPLS network. [19] describes two types of attacks that can happen and has termed them as cross-domain attacks.

1. Promiscuous Path Acceptance. Integrated Services (IntServ) [21] is a Quality of Service Architecture that incorporates a variety of signaling, admission, traffic management and scheduling protocols. A service provider implementing IntServ QoS in a traditional switched network typically uses Resource Reservation Protocol (RSVP) messaging. In such cases, a PE node in an adjacent MPLS network would accept packets with the Router Alert Option set as specified by the RSVP protocol [22]. Once these RSVP messages enter the MPLS network they are guaranteed to be forwarded. This attack is particularly dangerous as if an attacker is able to pass RSVP messages into the MPLS network then an attacker is basically able to perform any traffic engineering changes.

2. Pre-Labeled Traffic Acceptance. A Provide Edge node needs to be configured to accept traffic from a certain Customer Edge node that is authenticated. The PE node must use pre- platform scoping rules, otherwise the labels sent from the CE

node interface would not have bindings. Accepting pre-labeled traffic exposes an MPLS network to any number of signaling attacks from a compromised CE node [19].

1.4 Exisiting security tools

Given that the MPLS technology is implemented widely in the core networks, several Security tools already exists to make MPLS more secure. Each tool having its own set of features and complexities to make data transmission over MPLS network more secure. MPLS mostly relies on external security protocols or frameworks for secure data transmission. A thorough analysis needs to be made regarding which tool to be used based on the operational requirements and feasibility of its implementation in the current scenario. Each of these tools provides a different mechanism to provide different security functions to the MPLS network. There is no one single tool that solves all the security issues plaguing the MPLS network, however proper consideration and analysis of the security requirements of the system and using the appropriate tool to mitigate them is more than enough to make MPLS secure for business requirements. Based on this idea it is best to have a suitably large number of security tools in the arsenal to fight off the possible security threats. Security tools range from a wide variety of mechanism to make MPLS secure. some of which are described in the following.

1.4.1 Application Data Encryption

Encrypting application data by the application itself is the best form of security that any application can guarantee. It puts the responsibility of the security of the application data on the user and application developer itself. The security of the application data becomes independent of the security of the underlying data transport system. In this scenario the application developer has the freedom to choose any security tool that best suits for his application and requirements rather than relying on the inherent security of the underlying network system. The issue with application data encryption is that not many developers do not take the necessary steps to secure their data. It is an additional cost to the development company to take the effort to implement appropriate security measures to secure their vital data as it is being passed onto the internet.

Thus implementing opportunistic security in the underlying network will provide that additional safety net to prevent any security breaches.

1.4.2 Transport Layer Security (TLS)

One of the ways to make MPLS more secure is to encrypt the underlying data that is to be passed through the LSP. This prevents any sniffing attacks from sniffing any confidential data that a company or service provider may be transmitting via the MPLS network. Packet Encryption is ideally a responsibility of the application that is sending or receiving the data. Applications can use security tools like Transport Layer Security (TLS) which is an Internet standard to implement privacy and data integrity between communicating computer applications. The issue with implementing TLS is solutions like TLS leave some metadata (such as the destination IP address) exposed as the packets transit the IP network [2]. An attacker can sniff this information and identify the source and destination IP of the packet. The attacker can thus implement any of the route modification attacks as mentioned in the previous section leading to security breaches.

1.4.3 IP Security (IPSec)

IPsec is another tool which can be applied end-to-end or hop-by-hop applications to maintain privacy and data integrity. It is mainly used to encrypt IP packets that flow through the network. IPSec can be used to encrypt the IP packets before they are passed onto the MPLS network. IPsec has historically placed a heavy "full-mesh" configuration burden on implementation although this is now ease with the introduction of the NULL Authentication Method in the Internet Key Exchange Protocol Version 2 allows for opportunistic key exchange to support IPsec [1]. MPLS is a multi protocol data forwarding technology. It's main strength lies in its ability to transfer data of different type of protocols via its MPLS network. IPSec may be a very good security measure to secure IP data in a network, however from an MPLS implemntation opoint of view it will only provide security if the data is of IP data type. Furthermore, IPsec has historically placed a heavy "full-mesh" configuration burden on implmenetations although this is now ease with the introduction of the NULL Authentication Method

in the Internet Key Exchange Protocol Version 2 [23] allows for opportunistic key exchange to support IPsec.

1.4.4 Link layer security (MACSec)

Moving down the IP stack, encryption is also possible in the Layer 2 (Link Layer). Packets can be encrypted on a hop by hop basis between two communicating routers using MACsec. MACsec encrypts Ethernet frames that transmit across ethernet network. Thus end to end security can be implemented by creating a chain of trust between all the participating routers in the network path.

1.4.5 Pseudowire Encryption

MPLS is used to transport data of multiple protocols like Ethernet, ATM, TDM along with IP. Security tools limited by protocol types often don't help in fully leveraging the strengths of MPLS networks. Security tools that work for all types of protocols go well with MPLS implementation. Such a tool is pseudowire (PW) encryption. PW security is carried out by setting up pseudowires that tunnel the native service through the MPLS core by encapsulating at the edges [13]. The benefits of PW encryption is that compromising of edges or routers becomes very difficult. Protection of control plane messages means protection from majority of attacks. PEs are usually configured to reject MPLS packets from outside the service provider network, thus ruling out insertion of PW packets [1] from the outside [13] The very fact that MACSec is implemented hop-by-hop in every switch makes it less ideal in an MPLS network scenario. Since it is hop-by-hop encryption and decryption of the data packets happens in each and every switch. This can lead to higher latency in data packet transfers if there are a large number of switches in an LSP through which the data packet is flowing through.IPsec has historically placed a heavy "full-mesh" configuration burden on implmenetations although this is now ease with the introduction of the NULL Authentication Method in the Internet Key Exchange Protocol Version 2 [RFC7619] allows for opportunistic key exchange to support IPsec [2].

As we see all these security tools are powerful sets to provided the required security in a given situation. Each tool has its own Pros and cons, some may need additional enhancements while most are self sufficient in a given scenario. Most of the litera-

ture covering MPLS security are also concerned with a certain use case in mind. [1] [11][12][15] analyzed the security issues of MPLS VPN. [3] concerned with the security framework to be implemented for the MPLS network. [8] elaborated on the security concerns in implementing label stack encoding. [19] Emphasized on the Security issues involved in RSVP signaling in MPLS. Opportunistic Security in MPLS is not aimed to replace or displace any of the existing security tools currently present. It is only an addition in the arsenal of security tools aimed to make the internet more secure.

Chapter 2

Design

In this chapter we will aim to explain the underlying functionality of the various technologies involved in implementing Opportunistic Security in the MPLS network. We will discuss on the Architectural design of the system and the reason for choosing them.

2.1 Opportunistic security (OS)

Historically, The approach to internet security has always been with an 'all-or-nothing' attitude. Security protocols aimes at either providing the service with full security or a complete hard failure. This discouraged many service providers from implementing the latest secure tools to make their network secure. Service providers have a responsibility to maintain their Service Level Agreement (SLA) and as such will often end up disabling these security tools and pass clear text through their network whenever the connection becomes too slow or customers have trouble connecting to services.

Authentication, Confidentiality and Integrity are the three pillars of network security. Confidentiality and Integrirty can be brought about using Encryption techniques with hashing mechanisms. Authentication on the other had has to be carried out on a peer to peer basis. The ability to authenticate any potential peer on the Internet requires an authentication mechanism that encompasses all such peers. No IETF standard for authentication scales as needed and has been deployed widely enough to meet this requirement [30].

Browsers, web services and network devices refer to the Public Key Infrastructure

(PKI) model to authenticate their peers or web services. Implementation of the PKI has its own additional costs and burdens and considering the large number of Certificate Authorities that provide this service on the internet, many service providers have trust issues with regards to the CAs. This may lead to disagreement between peers in the network that are trying to connect to eachother.

Thus if authentication of nodes in a network becomes optional, where the system can run normally with plain text if authentication cannot be easily attained then the network system is more likely to implement Encryption in majority of its links. This way the approach to network security changes from 'security being default and anything less than that as degraded security' to 'No protection being default and anything more than that is an improvement'

"Opportunistic Security" (OS) is defined as the use of cleartext as the baseline communication security policy, with encryption and authentication negotiated and applied to the communication when available [30]. The aim of OS security is to implement encrypted and authenticated communication between peers whenever they are capable else only encrypted communication without authenticating the users or just clear text communication.

OS is not intended as a substitute for authenticated, encrypted communication when such communication is already mandated by policy (that is, by configuration or direct request of the application) or is otherwise required to access a particular resource. In essence, OS is employed when one might otherwise settle for cleartext [30].

2.1.1 OS in MPLS

As proposed in [2] the basic requirement in MPLS OS is to provide a way to encrypt data passing between two MPLS switches by doing a key exchange between them to create a session key using which the encryption can be carried out. The key exchange between the LSRs is to be carried out using the Diffie-Hellman key exhange. Using the Keys values agreed after the Key exhange we will encrypt the flowing packets using Advanced Encryption Standard (AES) cryptographic algorithm. To enable authentication of the peers as well in the encryption it is suggested to use AES in Galois/Counter Mode (GCM).

Diffie-Hellman Key Exchange

Diffie-Hellman Key Exchange is a way of sharing secret information between 2 nodes over an insecure channel. It was developed by Martin Hellman, Whitfield Diffie and Ralph Merkle. Using this protocol the 2 nodes agree on a secret symmetric key which can be later used in encrypting the sensitive data that is to be communicated.

The Diffie-Hellman key exchange will be modelled based on the Internet Key Exchange Protocol Version 2 as specified in the RFC [31].

Advanced Encryption Standard - Galois/Counter Mode (AES-GCM)

AES is a symmetric blockcipher algorithm established by the U.S. National Institute of Standards and Technology in 2001. It was developed by Vincent Rijmen and Joan Daemen who won the NIST conducted AES selection process. AES is able to handle block encryption of sizes 128, 192 and 256 bits. It is expected to provide data security for 20-30 years and is free to use for all devices.

GCM is a mode of operation for block ciphers like AES. It extends AES's encryption functionality to incorporate Authentication and Data Integrity as well. GCM is widely adopted because of its efficiency and parallel processing. Unlike cipher block chaining whose pipeline oprations bottle neck its efficiency, GCM makes efficient use of its instruction pipeline to provide high speed operations using parallel processing.

The Data packet in the MPLS OS must be encrypted with AEAD-AES-GCM-128 encryption algorithm based on the specifications mentioned in the [33].

2.1.2 MPLS Packet Encryption

The MPLS packet is encrypted using AES-GCM encryption algorithm whose Key and Initialization Vector are determined by the initial Diffie-Hellman Key exchange carried out between the two LSRs in the MPLS network. The encryption state of the MPLS packet is identified by the addition of a special purpose MPLS label called the MPLS Encryption Label (MEL). The bottom of the stack of the MEL is set to 1 and should be followed by a 4 byte Pseudowire control Word (CW).

As specified in the RFC [32] the packet counter nonce used in the AES-GCM needs to be managed properly for successful Encryption and Decryption on the LSRs. The

initial nonce value is derived from the HMAC-based Key Derivation Function (HKDF) output (see Section 4.3.3) at key agreement time and the counter is incremented by one for each packet encrypted on the sending side and by one for each packet successfully decrypted on the receiver side [2].

The CW carries the 16 bit nonce value modulo 65536 which helps the receiving LSR to identify any dropped packets or mismatch in the IV packet counter. The REceiving LSr can then update the counter or resynchronize the counter to successfully decrypt the packets. The receiving LSR can raise alerts if more than 65536 packets are lost as the LSR will face a decryption failure, thus it is advisable to implement small window size to accept mismatched counters, beyond which the LSR will stop further decryption attempts to mitigate denial of service.

The AES-GCM will generate a cipher packet which is slightly longer than the original packet. This is because of the additional authentication tag generated by the AES-GCM algorithm to provide authentication capabilities. The Receiving LSR will verify the authenticity of the encrypted packet as specified in RFC [33] by refering this authentication tag.

[Refer diagram 5] shows the format of a normal MPLS packet with the MPLS with OS packet. In the normal MPLS packet an MPLS label with its bottom-of-stack (S) value 1 is pushed on top of the payload. Subsequent MPLS labels can be pushed over eachother with values S = 0. Before transmission on the network the final Layer 2 header is pushed on top of the top MPLS label. In the MPLS OS data packet, the structure slightly changes. The Layer 2 header and the top MPLS label stays the same. The structure is then followed by the MPLS label with value 15. The value 15 informs the LSR that following this packet is a special purpose MPLS label. Then comes the MEL followed by the CW containing the nonce modulo 65536 value. The remainder of the packet is encrypted and contains the rest data of the packet.

MPLS Encryption Label

The MPLS Encryption Label (MEL) is a normal label stack entry carrying an extended special-purpose label with a value from the experimental range 240-255 [2]. The structure of the MEL is same as that of the traditional MPLS Label. The values of the Label contents is what differentaites the MEL from the traditional MPLS labels.

Label

Considering this idea is still under experimentation no special purpose label value is yet assigned by the IANA. The value however must be selected from the experimental range of 240-255

TC

The TC field should be set to the value of the unencrypted label stack entry directly preceding the the MEL else it should be set to 0.

S

The MEL should always be the bottom of the MPLS stack and thus its value S should always be 1.

TTL

The TTL should ideally be set to 2 for end-to-end MPLS OS encryption to prevent the encrypted packets from being forwarded beyond the decrypting LSR.

CW

An LSR may tend to inspect the contents of an incoming packet to analyse its underlying protocol eg. check if it is IP and forward it via normal IP routing if so configured. The presense of the Cw along with the MEL informs the receiving LSR that the contents of the MPLS packet is not of any standard protocol and thus cannot be inspected.

[Refer figure 6] shows the internal structure of the CW as specified in [2].

Flags

The Flags field is treated as a four-bit number. It contains the key-id that identifies the algorithm and key as established through configuration or dynamic key exchange [2].

Fragmentation FRG

FRG indicates Fragmentation, MPLS OS doesnot support fragmentation of the Data packets and as such the FRG should always be set to 0.

Length

Length is set to 0 and should be ignored by the receiving LSR

Sequence Number

This field contains the nonce which is currently being used for the currently agreed encryption parameters. It is indicative of the counter used in the AEAD-AES-GCM encryption which the receiveing LSR can use to check it its counter is correct and if it can go ahead with the decryption.

2.2 Technologies Involved

Considering the experimental nature of this project a proper base architecture needs to be designed to cover all the requirements of the experiment. This architectue will form the blueprint on which the technologies involved will be implemented to mimic the behaviour of the MPLS network with and without the OS as close to the real life behaviour as possible.

The ideal controlled test bed for this project should involve a virtualized environmental system on which network routers could be set up. The building up and tearing down of the network should be quick and clean so as to prevent long loading times or incomplete changes in the network settings even before the experimentation could begin. The virtualized routers involved should have all the basic functionality required for proper implementation of the MPLS network. The Routers should also be able to process the data packets at the kernel level so as to get readings as close to real-life routers as possible without any latency added due to the implemented technology. Secondly the Router functionality should be configurable to implement the OS functionality into the switches.

Based on these requirements the following technologies were chosen to implement the MPLS OS experiment.

2.2.1 Mininet

Mininet is a technology that can create realistic virtual network on a single machine. It is able to run real kernel and application code to provide authentic network behaviour in a virtualized form. It has support for various switches all set up in a Software Defined Architecture. Mininet can yield a more efficient use of time and resources compared to other workfows. It provides a local environment for network innovation that complements shared global infrastructure [34*6],

Mininet is used for a wide variety of cases such as optimization of topology designs, tutorials for various network technologies, demonstrations, Regression Suits and most importantly Prototyping [34]. Mininet has the capacity of rapid and simplified prototyping, the applicability safety, the possibility of sharing results and tools at zero cost [36].

[Refer Diagram 6] [34] has descirbed a basic architecture of how Mininet create a virutalized network with kernel functionality. Mininet creates a virtual network by placing host processes in network namespaces and connecting them with virtual Ethernet (veth) pairs. In this example, they connect to a user space OpenFlow switch.

2.2.2 OpenVSwitch (OVS)

OVS is a multi layer Virtual Switch technology used to implement the functionalities of a hardware switch in a virtualized manner. It is designed for network automation on a large scale using programmatic extension, while still supporting standard management interfaces and protocols (e.g. NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag) [OVS Website].

Open vSwitch works with most hypervisors and container systems, including Xen, KVM, and Docker. Open vSwitch also works out of the box on the FreeBSD and NetBSD operating systems and ports to the VMware ESXi and Microsoft Hyper-V hypervisors are underway [6]. It is designed to be flexible and portable and lies inside the hypervisor to provide connectivity between the virtual switch and the physical interface. [Refer Diagram 7] for a brief explanation of Openvswitch architecture.

OpenVswitch relies on 2 major components for forwarding packets in a network. The ovs-vswitchd which is a daemon that lies in the userspace of the system and the smaller datapath kernel module. The ovs-vswitchd daemon is different in different

operating system environment however provides the same functionality, the data path kernel module on the other hand is written especially to funciton at the kernel level.

[Refer Diagram 7] explains how the components of the OpenVSwitch interact with eachother. A packet which arrives on the interface of a physical or virtual NIC is passed onto the datapath kernel module. The datapath kernel module will do either of the following things: Forward the packet based on the flow logic instructed by the ovs-vswitchd or Pass it to the ovs-vswitchd to await instructions on what actions to take on such types of packets.

Kernel Forwarding

The ovs-vswitchd daemon instructs the kernel datapath module how to handles packets of specific types or 'flows'. These instructions are called 'actions'. The actions may specify a range of funtionalities like modification of packets, sampling, packet dropping, re-routing etc. The kernel datapath simply follows the flow rules set by the ovs-vswitchd and executes the actions mapped to them. The kernel datapath can store these flow rules from the ovs-vswitchd daemon in its cache to act accordingly on similar types of data packets. If an incoming packet doesnot match any of the flows stored in the kernel modules cache, i.e a cache miss, then the module forwards it to the userspace instead.

Userspace Forwarding

When the kernel module forwards a packet to the userspace due to a cache miss the ovs-vswitchd daemon determines what actions are to be taken on the data packet. The Daemon then forwards the action to the kernel Datapath to cache it and handle future similar packets.

Open vSwitch is commonly used as an SDN switch and the main way to control forwarding is OpenFlow [6*27]. It leverages the use of Open Flow protocol to add, update, delete flows in a switch's flow table. The ovs-vswitchd daemon receives these flow table from n SDN controller. It also has a great support for MPLS from version 2.4 onwards where you have kernel support for MPLS packets upto a maximum of 3 label stack.

2.2.3 Openflow

Networks have become a critical part of our day to day lives from business to research. Due to the large number of equipment and protocol installation overhead along with the reluctance of Network owners to experiment with their production network so as to not affect ongoing services, Network researchers have a difficult time in experimenting with new protocols and network technologies. This is where OpenFlow comes in. OpenFlow provides an open protocol to program the flowtable in different switches and routers [40]. A network researcher is able to use OpenFlow protocol to control the flows in a flow table of all the switches inside the experimental newtwork. This allows for easy and quick implementation of experimental setups which are ideal for research purposes.

An OpenFlow enabels a switch to perform actions based on the experimental protocol. using OpenFlow a researcher experiments with his protocol by running it on a controller. The protocol will pick a route in the network of OpenFlow enabled switches and add the flow entry into all the switches in its path. The protocol can instruct the switches to encapsulate, drop, forward or even encrypt and decrypt incoming data, which is our requirement. He can also set the flow rules to pick the type of data flows that pass through the Open flow enabled switch on which the actions are to be performed.

2.3 Architecture of Experiment

As Described in the previous chapter we will be implementing the MPLS OS system in a virtualized environment using mininet, OpenVSwitch and Open Flow technology. In order to simulate the MPLS OS structure and its effects on LSRs we need to design a network topology where we can analyse its overall effects in all cases. [Refer Diagram 8] Descirbes the topology of the network system we will be using to demonstrate the MPLS OS system.

We will be using Mininet as our virtual network environment. The fast and clean network topology build up and tear down speed makes Mininet a good choice for setting up the virtual network environment. The Mininet version used in this experiment is version 2.2.1.

We will be deploying OpenVSwitch switches in the mininet virtual network. Mininet

has inbuilt support for setting up OpenVSwitches in its Virtual environment. The collaboration of both mininet and OpenVSwitch makes both their use together in a system ideal for experimentation. The OpenVSwitch version used in this experiment is version 2.9.2

Flow rules and actions will be passed onto the OpenVSwitches using OpenFlow protocol. For this we need to setup an OVS controller which communicates the OpenFlow messages to all the OVS switches in the network. The OpenVSwitch Controller comes in built with Mininet and needs to be passed in the parameters for mininet to setup in the network.

As described in [Refer diagram 8] we will be setting up 3 switches (S1, S2, S3) in a linear topology. Each switch has its own host (H1, H2, H3) connected to it. Switch S1 has two interfaces s1-eth1 connected to host H1 and s1-eth2 connected to Switch S2 on its interface s2-eth2. Switch S2 has 3 interfaces s2-eth1 connected to host S2, s2-eth2 connected to Switch S1 on its s1-eth1 interface and s2-eth3 inteface connected to Switch S3's s3-eth2 interface. Switch s3 has 2 interfaces, s3-eth1 connected to Host H3 and s3-eth2 connected to switch S2 on its s2-eth3 interface. Controller C0 will communicate the MPLS flow rules to all the three switches.

The Expected behaviour of the system in normal MPLS process will be as follows:

1. Host 1 will ping an ICMP packet to Host 3
2. The packet will be first forwarded to Switch S1 at its s1-eth1 interface
3. S1, based on the flow rules that we passed via the controller, will push an MPLS label on top of the packet and forward the packet along s1-eth2 interface.
4. Based on the linkage, S2 will receive the data packet. S2 will read the MPLS header value and forward it along its s2-eth3 interface.
5. Based on the linkage, S3 will receive the data packet. S3 will then pop the MPLS header and forward the normal ICMP packet to H3 via its s3-eth1 interface.
6. H3 will reply to the icmp packet following the reverse path. the only difference being that now S3 will push the MPLS label and S1 will pop it at the end

The Expected behaviour of the system in MPLS OS process will be as follows:

1. Host 1 will ping an ICMP packet to Host 3
2. The packet will be first forwarded to Switch S1 at its s1-eth1 interface
3. S1, based on the flow rules that we passed via the controller, will first Encrypt the payload using AES-GCM encryption algorithm the keys and IV of which, for the scope of this project, are currently hard coded. After encryption the CW with the nonce value modulo 65536 is pushed on top of the Data packet. Then the MEL is pushed on top. A normal MPLS with value 15 is then pushed on top of the MEL completeing the MPLS OS data packet which is then forwarded onto the s1-eth2 inteface
4. Based on the linkage, S2 will receive the OS data packet. On parsing the Data packet S2 should stop any further inspection or hash checking functions on the underlying encrypted data. S2 should just read the MPLS header value and forward it along its s2-eth3 interface.
5. Based on the linkage, S3 will receive the data packet. S3 will then pop the MPLS header with the value 15. It will then read the MEL value and pop it, expecting the next header to be the CW. S3 will the compare its nonce counter modulo 66536 value with the one present in the CW. If its a match then the counters are correct and s3 can proceed with the decryption. If not then S3 will update its counter to continue decrypting the next oncoming packets. once successfully decrpyted S3 will then forward the normal ICMP packet to H3 via its s3-eth1 interface.
6. H3 will reply to the icmp packet following the reverse path with encryption on S3 and Decryption on S1.

The Diffie-Hellman Key exchange required to initialize the Symmetric Encryption Key and the Initialization vector for the AES-GCM encryption Algorithm is not implemented in this experiment and as such these values are currently hard coded into the system and can be later replaced with further developement on that part. This experiment currently only demonstrates the encryption functionality of the OS and how the implementation of the Encryption affects the performance of the overall system.

Chapter 3

Experiment

In the previous chapter we described the design details that we will be using to implement the experiment. In this Chapter we go into the implementation details of the project.

3.1 Testing the MPLS System

We will start of by first setting up the virtual network with the OVS Switches running the traditional MPLS System. To do this we first set up the mininet virtual network with the OpenVSwitch switches and hosts as described in Diagram 8. Once the network system is in place we add the MPLS flows to the switches using OpenFlow. And finally we run a ping test to pass the ICMP packets through the MPLS switches to test the system.

3.1.1 Setting up Mininet and OpenVSwitch

The Mininet tool package is available in the Advanced Package Tool (APT) of all Linux Distributions. It can be easily installed on the system using the package name 'mn'. The Mininet package has a dependency on OVS packages and thus are installed along with the mininet package by the APT command.

The Network topology can be set up by calling the command: sudo mn –topo=linear,3 –switch=ovsk –controller=ovsc

The parameter meanings are as follows: 1) –topo=linear,3 This parameter creates a topology of 3 switches each with their own host connected in a Linear fashion. 2) –switch-ovsk This parameter determines the type of switch that will be created in the mininet network. The value 'ovsk' indicates that OVS switches that run in kernel space are to be created 3) –controller=ovsc This parameter determines the type of controller that will be created in the mininet network. this controller is responsible for forwarding the flow rules and actions to all the switches in the network. the value 'ovsc' indicates that an OVS controller is to be created to communicate with the OVS switches.

3.1.2 Addition of the MPLS Flows

Once the network infrastructure is set up we then add the MPLS Flows to the switches via the OVS controller. This is done passing the flow rules and actions to the Openflow Switch Manager (ovs-ofctl).

The OpenFlow Switch management command structure is as follows: ovs-ofctl -O OpenFlow13 add-flow s1 "table=0,in_port=1,eth_type=0x800,actions=goto_table:1"

The parameter meanings are as follows: 1) -O Indicates the OpenFlow Version to use. The value 'OpenFlow13' indicates OpenFlow Version 1.3. 2) add-flow Indicates the action to be taken on the switches. We can add flows, delete flows modify flows etc using the OpenFlow Protocol. The value 's1' indicates the switch on which the flow rules are to be added. 3) "table=0,in_port=1,eth_type=0x800,actions=goto_table:1" Is the Flow rule where the flow is to be added in table '0' of the Switch, to packets arriving at port '1' which is mapped to interface s1-eth1, which are of eth type 0x800 (ICMP Packets). The 'actions' value indicates the type of actions to take on such packets.

The Complete Flow configuration for MPLS is as follows:

3.1.3 Testing the MPLS System

After adding the Flow Rules we can initiate the ping test from h1 to h3 by calling the ping command in the Mininet CLI. This would create an ICMP packet at Host H1 which would be forwarded to Switch S1. Switch S1 based on the Flow Rules as described in Diagram will forward the packet to its table '1' where it will push an MPLS label of eth value 0x8847, set its label value to 12 and forward it to switch S2. Switch

S2 will redirect all packets of eth type 0x8847 to Switch S3. S3 based on its flow rules will first forward the data packet to its table 1 where it will check if the MPLS label is a bottom of the stack label, pop the MPLS label if it is and later forward the packet to Host H3. H3 then responds to the ICMP request by sending the ICMP response to switch S3 after which the data packets is subjected to the flow rules as described in the 'Response Flow' part of the diagram.

To test if the packets are being routed correctly using MPLS labels we can inspect the packets at switch 2. This is done by calling the command tcp dump on the terminal of switch 2. Refer diagram

The output displayed by the ping command in diagram indicates that the switches are pushing, inspecting and popping the MPLS labels correctly without any issues.

3.2 Implementation of the OS Encryption functionality

The next part of the experiment involves the implementation of the OS Encryption functionality. Based on the OS designed as mentioned in chapter 2, the 2 additional functionality involved in the MPLS OS is the addition and removal of the CW, Encryption and Decryption of the MPLS Payload. These functionalities should be performed by the OVS Switch involved in the network experiment and as such should be implemented in the OVS code base.

After a thorough study of the OpenVSwitch Codebase, we found out that there is no such inbuilt provision in the OVS codebase using which we can implement either the CW or the Encryption/Decryption of the MPLS payload. The functionality was thus needed to be implemented manually into the OVS codebase and then converted into a kernel module for the Switches to implement the same at the kernel level.

For the sake of development classification we will segregate the OS functionality into 2 categories:

3.2.1 Changes at the Ingress (Entering) Switch

When the ICMP request packet from H1 enters S1, before pushing the MPLS headeron the data packet, S1 first encrypts the payload data using AES-GCM encryption algo-

rithm. The key and IV required for the encryption is currently hardcoded in the code as their values are dependent on the Key Exchange which is not implemented in this project. The AEAD-AES-GCM encryption is carried out using the Linux Kernel Crypto API. After encryption the pseudo wire Code Word is pushed on top of the encrypted data before pushing the MEL on top of it. The structure of the CW is described in section...

Due to certain implementation aspects of the Linux Kernel Crypto AEAD encryption API, instead of encrypting the Payload data first and then adding the CW we will first add the CW and then encrypt the underlying Payload. The details of this aspect will be discussed further in section...

Insertion of CW

We first create the CW data structure in the OVS codebase and assign the nonce value to the CW sequence attribute.

In OVS datapath and Linux kernel in general, all data packets are parsed into a fundamental data structure called a socket buffer (skb). The skb is a series of contiguous memory location storing the data packet information in the form of bytes. The structure of the skb is shown in diagram...

The linux kernel has in-built APIs to manipulate the contents of skb. Using these APIs we manipulate the data packet and insert our CW between the payload and the MAC header of the datapacket. This is done by calling these series of Kernel APIs. refer diagram which depicts the pseudocode implementing the kernel APIs to insert the CW into the data packet.

Encryption of Data using Linux Crypto API

The Linux Kernel Crypto API is a rich set of cryptographic ciphers and data transformation API used for cryptographic operations at the kernel level. To understand and properly use the Crypto API functions one needs to understand their underlying architecture specifications and the functional flow of these APIs.

The Kernel Crypto API refers to all encryption algorithms as 'transformations'. The encryption is carried out by 2 major components, The transformation object and the request handle. The transformation object contains all the settings and configurations

of the given encryption type to use. The request handler as the name suggest handles the encryption request.

Figure ... describes the flow of API calls to initiate the encryption of the packet data.

Once the payload data has been encrypted S1 can now further push the MEL label on top of the packet followed by another normal MPLS packet for routing.

3.2.2 Changes at the Egress (Exiting) Switch

Once the encrypted MPLS OS packet reaches the Egress Switch S3, S3 pops the top MPLS label along with the MEL. It then inspects nonce value in the CW header. If the nonce counter value matches with the counter value of switch S3 then the switch moves ahead with the decryption.

Decryption of Data using Linux Crypto API

Decryption of the MPLS payload follows the same procedure as the one during Encryption process with a slight change in the pseudocode. Using Linux Kernel Crypto API the Switch creates a transformation object and a request handler. It passes the relevant parameters to the transformation object and initiates the asynchronous decryption of the payload data. During decryption it will also validate and authenticate the data based on the authentication tag generated during the encryption.

Figure ... describes the flow of API calls to initiate the decryption of the packet data.

Removal of CW

Just as how we used the in-built SKB manipulation APIs to insert the CW in the Ingress Switch, we will use the same approach while removing the CW from the Data packet. The pseudocode to remove the CW is show in diagram ....

3.3 Testing the MPLS OS System

Once the OS functionality has been implemented in the code base we need to build the system for testing. WE run the following commands to do so. 1) make 2) make install 3) make modules install 4) modprobe openvswitch

3.3.1 Addition of the MPLS OS flows

Compared to the flows passed during the testing of MPLS without OS add flows run ping Display functionality

Chapter 4

Results

4.1 Analysis of the metrics

4.2 Results

Chapter 5

Future Work

Chapter 6

Conclusion

Appendix

...

Bibliography

[1] Denise Grayson, Daniel Guernsey, Jonathan Butts, Michael Spainhower, and Sujeet Shenoi, "Analysis of security threats to mpls virtual private networks", *International Journal of Critical Infrastructure Protection*, vol. 2, no. 4, pp. 146 – 153, 2009.

[2] Sahel Alouneh, Abdeslam En-Nouaary, and Anjali Agarwal, "Mpls security: an approach for unicast and multicast environments", *annals of telecommunications - annales des télécommunications*, vol. 64, no. 5, pp. 391–400, Jun 2009.

Appendix

...