



96 more bits, no more magic

How many of you have
an IPv6 Strategy?



Those of you who don't

Today you'll learn why you
should have one...



Who Are We?

Martyn Ruks

- Researcher
- Consultant
- Technical Director

Daniel Bradberry

- Developer
- veripy Project Lead



- Introduction
- Why does v6 Matter?
- So, about that Strategy....
- Working as a Community
- Demo
- Conclusion
- Going Forwards

96 more bits, no more magic

Introduction



Let's back up a Minute...

If...

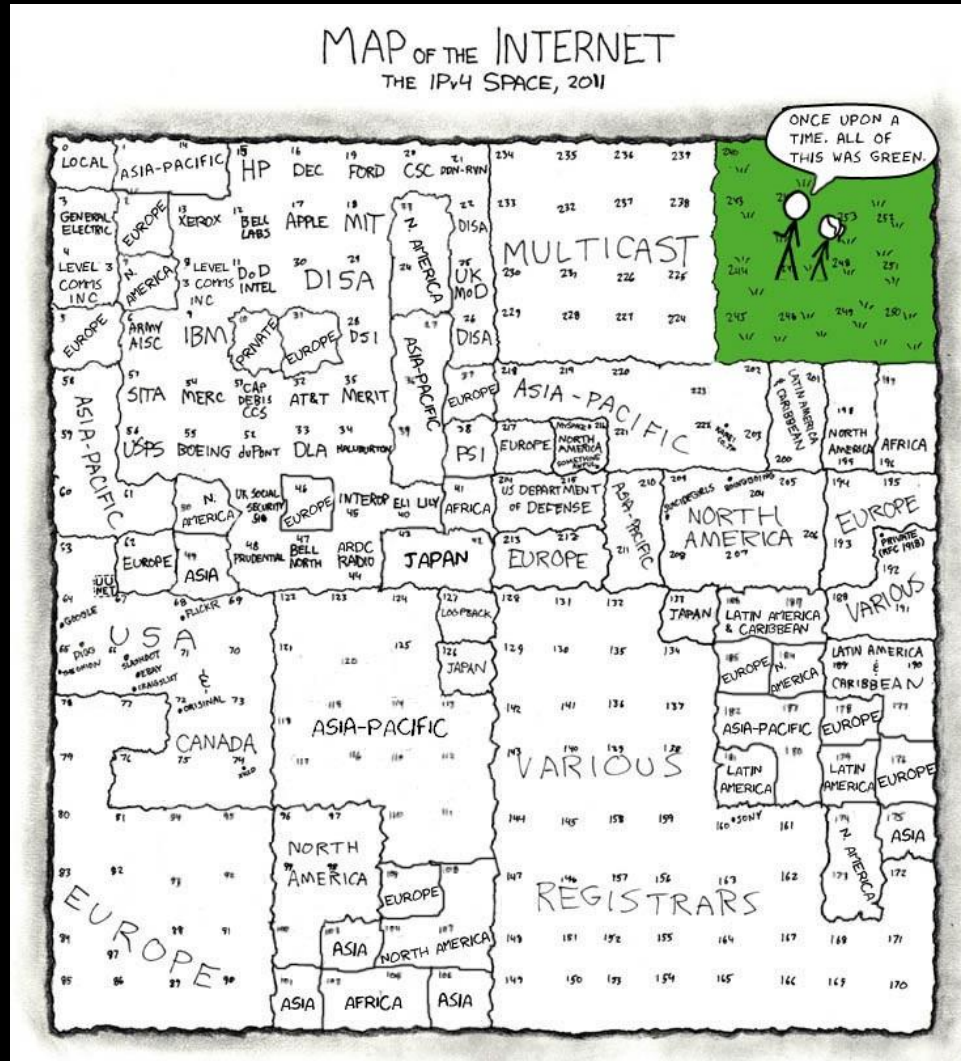
640K of memory should be enough for anybody.

Surely...

4,294,967,296 addresses is enough.



source xkcd.com



source xkcd.com

23
2010
September

~~03
2011
February~~

AfriNIC	Oct 20, 2014
APNIC	Apr 15, 2011
ARIN	Jul 19, 2013
LACNIC	Jan 30, 2014
RIPE NCC	Jul 26, 2012

Updated 2012-01-16, Data from iNetCore.com

IPv4

- 32-bit address
- 192.0.43.10
- 4 294 967 296 addresses

IPv6

- 128-bit address
- 2001:500:88:200::10
- 340 282 366 920 938 463
463 374 607 431 768 211
456 addresses

96 more bits, no more magic

Why does v6 Matter?



Why Does this Matter?

- The end-user adoption of IPv6 is imminent.
- Organisations need to be prepared to roll out IPv6 services.



But it'll never happen.™

IANA can recover some of the class-As wastefully allocated in the early days of IPv4.



We've used NAT to extend the life of IPv4
so far, we can use that again.

[Realm-Specific IP] ... the host can lease some of the gateway's addressing parameters in order to establish a global network presence.



We could introduce a CNAME2 type for DNS, that includes a “real” and “apparent” port.



But...

... what if the naysayers are wrong?

- Organisations will have to roll out IPv6 services, at short notice.
- There will be less time to plan, less time to test and more chance of something going wrong.

How well can your organisation survive a prolonged IT outage?

96 more bits, no more magic

So, about that Strategy...

MWR What goes in the Strategy?

- Training
- Network Design
- Security
- Address Allocation
- Transition
- Equipment



What about Equipment?

Organisations will seek assurance that network equipment works as expected.



“We Support IPv6” (Vendor X)

They Say

We have spent £x million testing every possible eventuality and error condition, and can guarantee it will work.

The Reality

You can configure an IPv6 address on an interface.

It'll probably work...



Every man for himself

- Each organisation defines a standard for their use-case.
- They can build their own toolsets for testing this.

As a community

- The community defines a standard for a typical use-case, with options for customisation.
- They build toolsets for testing this, with peer review.

96 more bits, no more magic

Working as a Community

MWR The Community is Doing it

- Working Groups
- RFCs, IETF standards
- IPv6 Ready Logo program
- RIPE-501

MWR The IPv6 Ready Logo program

- A conformance and interoperability testing program.
- Aims to build user confidence.
- Defines a series of tests a device must withstand to be considered “IPv6 Ready”.



Phase 1

- IPv6 Specification
- Neighbour Discovery
- Address Auto-configuration
- ICMPv6

- 170 tests

Phase 2

- IPsec
- IKEv2
- MIPv6
- NEMO
- DHCPv6
- SIP
- MLDv2

- 450 tests

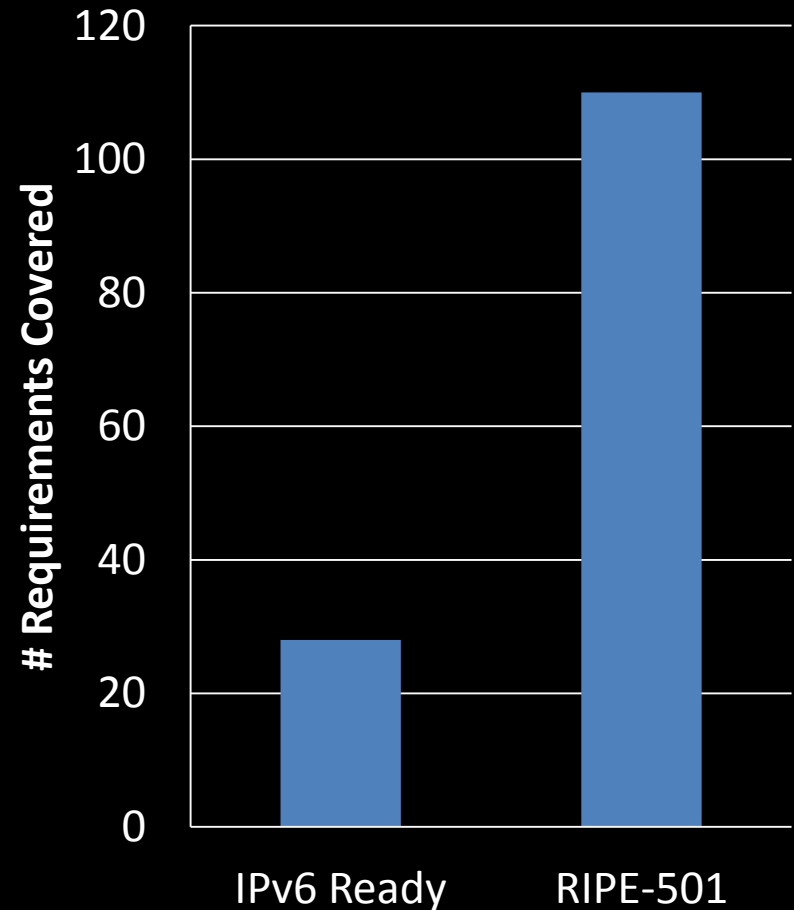


The Trouble with Logos

- It is a process of validation, not a challenge to the community to do better.
- It tests the stack in a contrived configuration, not a real-world configuration.
- It requires testing to be conducted in one of 6 laboratories.

- A best common practice template for use when procuring IPv6-enabled equipment.
- Complementary to the IPv6 Ready Logo program.
- Many more requirements.

RIPE-501



RFC 1772

RFC 1981

RFC 1997

RFC 2080

RFC 2401

RFC 2402

RFC 2406

RFC 2407

RFC 2408

RFC 2409

RFC 2460

RFC 2472

RFC 2473

RFC 2474

RFC 2545

RFC 2597

RFC 2671

RFC 2710

RFC 2711

RFC 2784

RFC 2918

RFC 3041

RFC 3140

RFC 3226

RFC 3246

RFC 3289

RFC 3315

RFC 3319

RFC 3392

RFC 3411

RFC 3412

RFC 3413

RFC 3414

RFC 3484

RFC 3596

RFC 3633

RFC 3736

RFC 3775

RFC 3810

RFC 3971

RFC 3972

RFC 4193

RFC 4271

RFC 4291

RFC 4292

RFC 4293

RFC 4294

RFC 4301

RFC 4302

RFC 4303

RFC 4306

RFC 4360

RFC 4443

RFC 4541

RFC 4552

RFC 4632

RFC 4659

RFC 4718

RFC 4760

RFC 4798

RFC 4807

RFC 4821

RFC 4861

RFC 4862

RFC 4877

RFC 4884

RFC 4891

RFC 5095

RFC 5120

RFC 5308

RFC 5340

RFC 5555

RFC 5969

RFC 6106

RFC 6146

RFC 6147

RFC 6164

RFC 6204

RFC 6333

RFC 6334

- Specifies requirements for a variety of network equipment.
- host (client or server)
- layer-2 switch
- router or layer-3 switch
- network security equipment
- consumer premises equipment
- mobile node
- load balancer



Microsoft Windows

- 1998 Windows 98 gets trial IPv6 Protocol Stack
- 2002 XP SP1 has an optional, production-ready IPv6 stack
- 2004 XP SP2 has additional tools
- 2007 Vista has a proper dual-stack, that is enabled by default

Linux Kernel

- 1996 v2.1.8 contains the first IPv6-related code
- 2000 The USAGI project aims to provide patches for incomplete support in the kernel
- 2006 The USAGI patch is integrated into the vanilla 2.6.x kernel series



96 more bits, no more magic

Testing against the Standard

verity

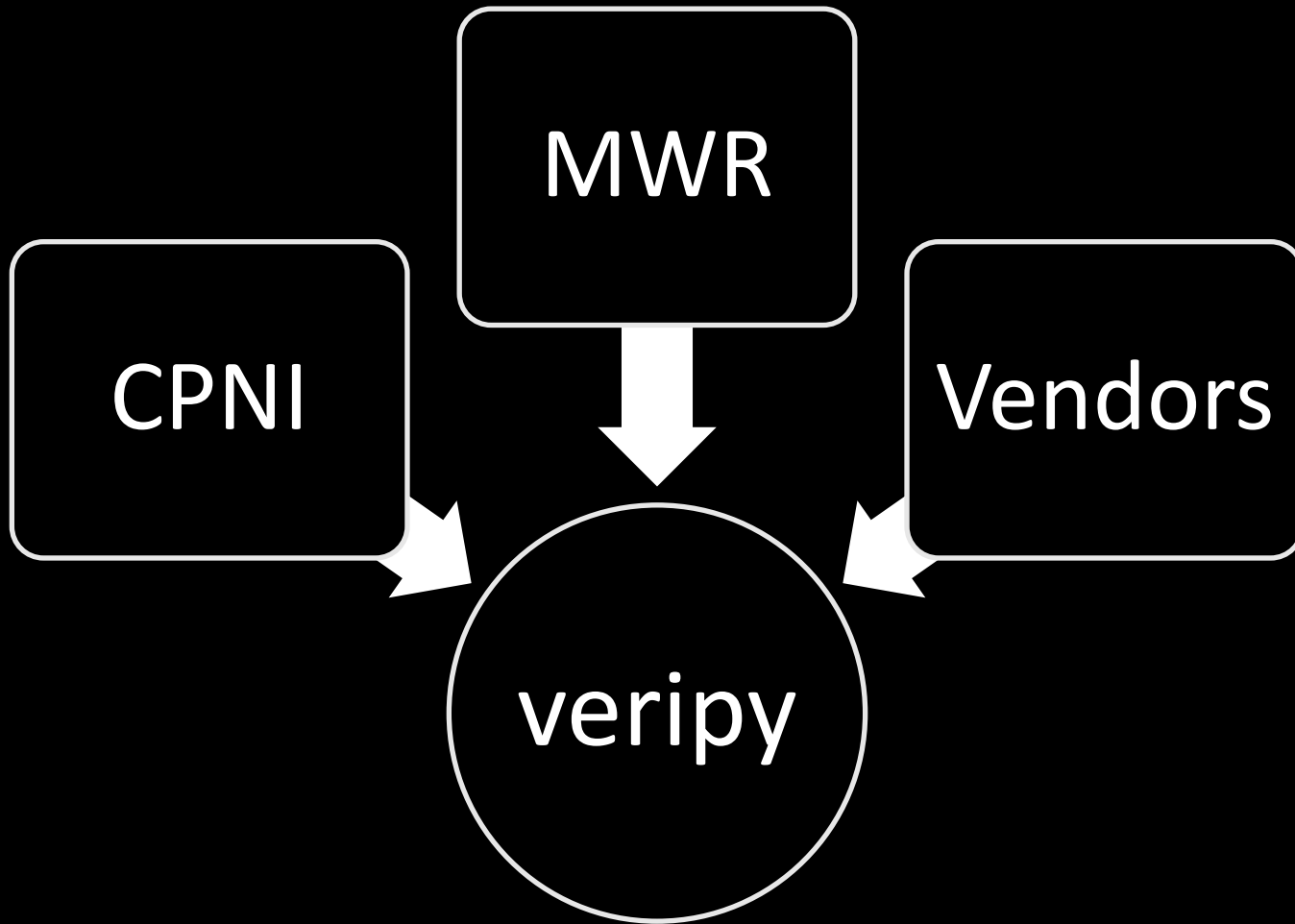
A shield-shaped icon with an orange-to-white gradient. Inside the shield, the letters 'ip' are written in a large, white, lowercase sans-serif font. Below 'ip', the text 'v6' is written in a smaller, white, lowercase sans-serif font. The shield has a white border and a slight drop shadow.



veripy

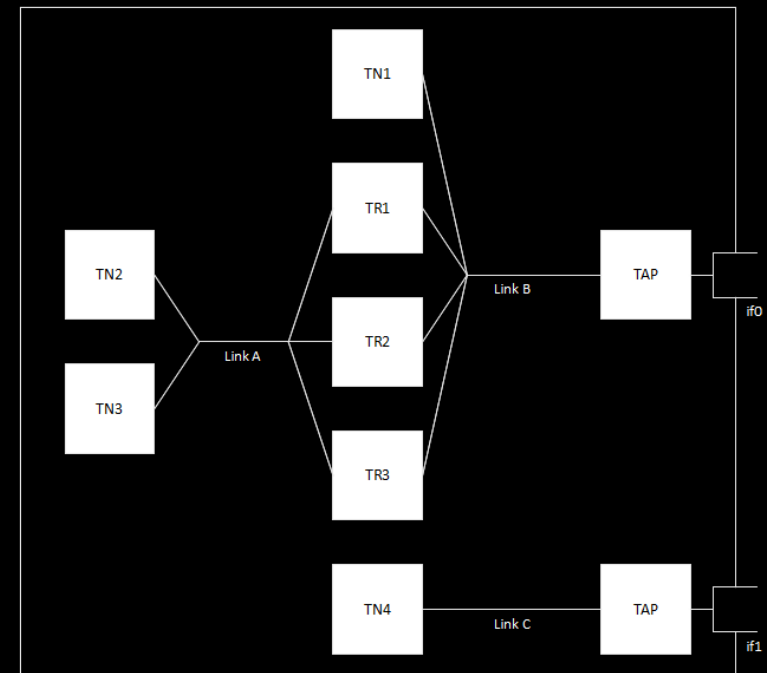
veripy is a tool that methodically checks ICT equipment for compliance with the RFCs specified in RIPE-501

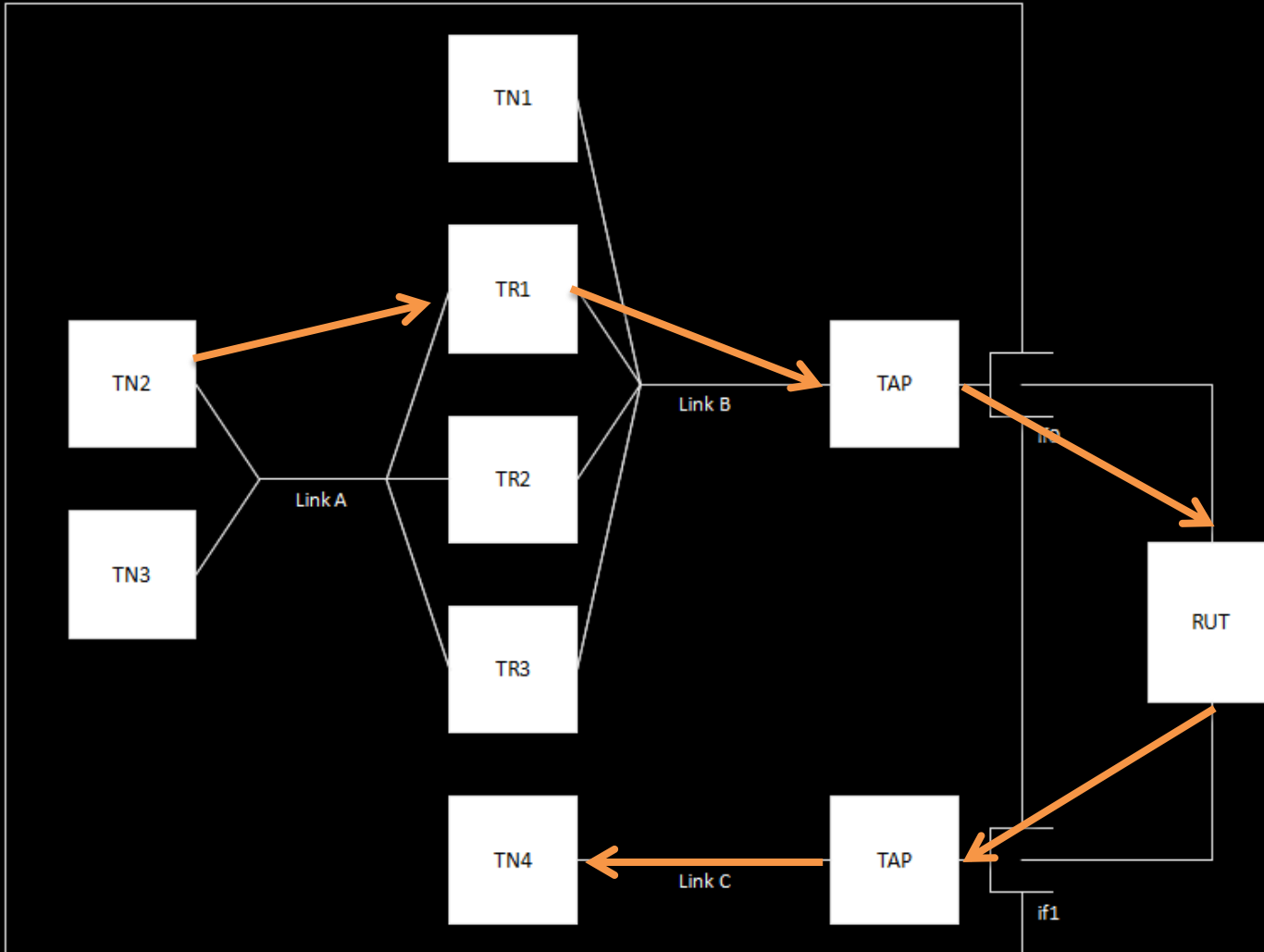
... [to give] confidence and assurance in the hardware and software products that are to support IPv6 networking moving forwards.



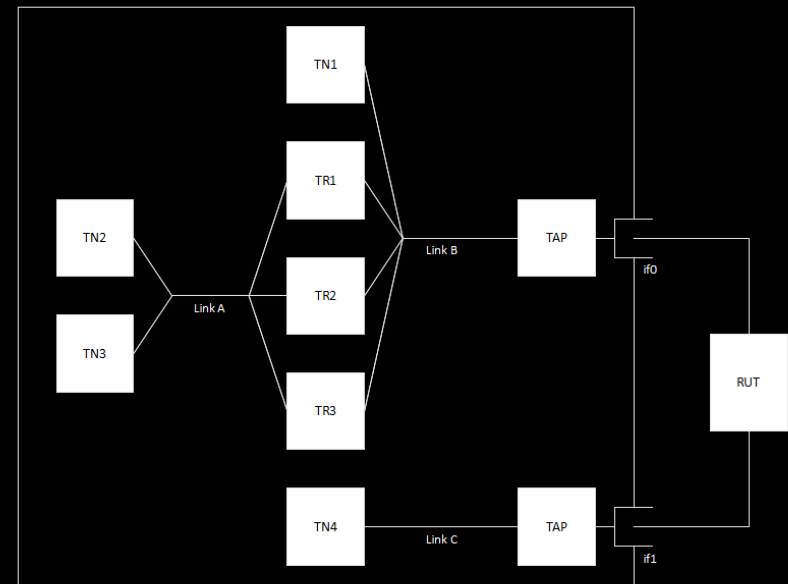
- a Network-centric compliance testing framework;
 - a set of test suites for the various requirements set out in RIPE-501; and
 - a configuration to run the correct tests for each type of device that can be tested.
-
- Free (as in beer)

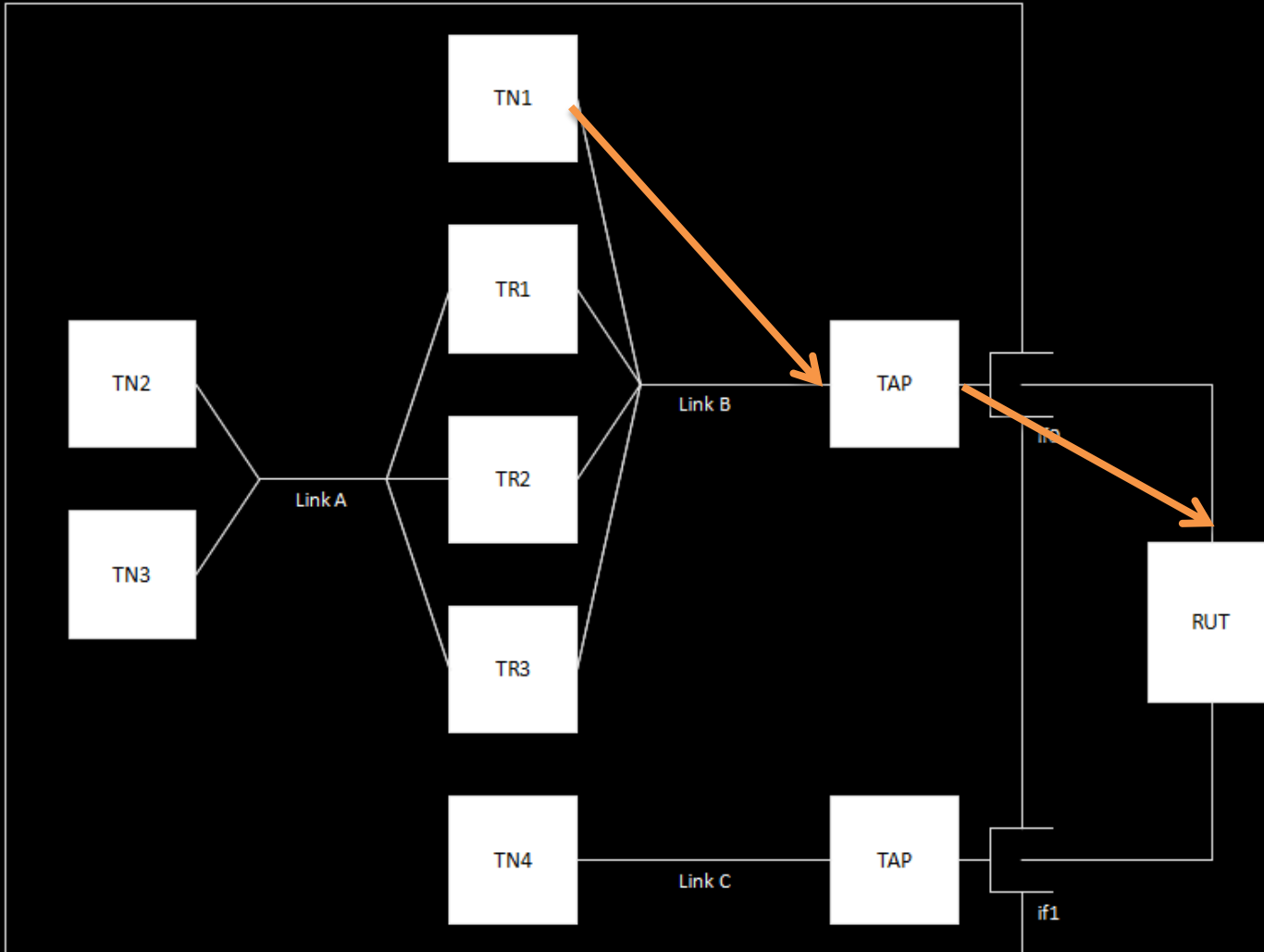
- veripy simulates a simple network internally
- It bridges these internal networks to physical interfaces.
- Nodes on the simulated network send traffic, and monitor the responses.



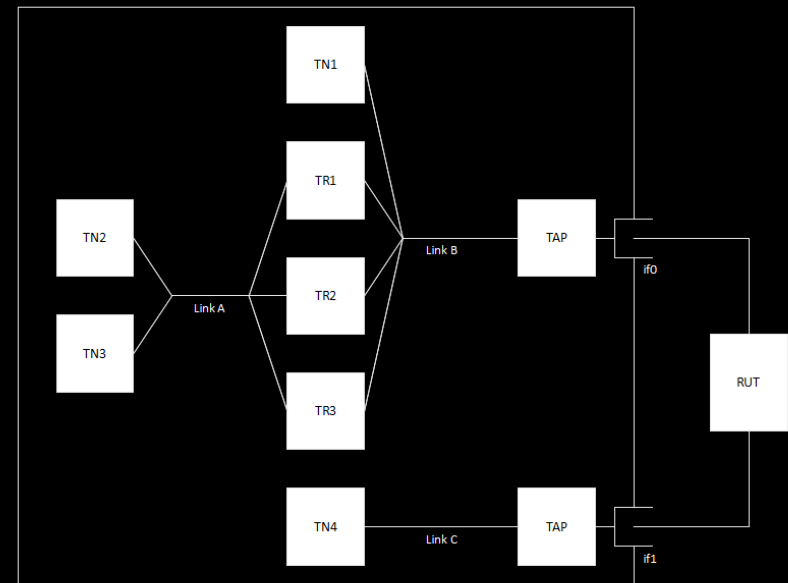


- ✓ TN4 receives an ICMP Echo Request (r).
- ✓ r's source is TN2's global IP address.
- ✓ r's destination is TN4's global IP address.
- ✓ r's IPv6 Hop Limit is 62.





✓ TN4 does not receive an ICMP Echo Request.



```
class DoesNotRouteLinkLocalAddress(ComplianceTestCase):  
    """  
    Router does not forward a packet sent to a link local IP.  
    """  
  
    def run(self):  
        self.node(1).send(  
            IPv6(dst=self.node(4).link_local_ip)/  
            ICMPv6EchoRequest())  
  
        assertNoPacket(self.node(4),  
            src=self.node(1).link_local_ip(),  
            type=ICMPv6EchoRequest)
```

96 more bits, no more magic

Demo

- We are going to run veripy against a Virtual Machine.
- It is a default installation of Ubuntu 11.10, running under VMware Player.
- We will run a few test cases around processing of the IPv6 version field.


```
dbradberry@ubuntu: ~
```

```
collisions:0 txqueuelen:1000  
RX bytes:46308 (46.3 KB) TX bytes:20337 (20.3 KB)
```

```
dbradberry@ubuntu:~$ ifconfig eth2
```

```
eth2      Link encap:Ethernet  HWaddr 00:0c:29:f4:b8:a4  
          inet addr:192.168.79.129 Bcast:192.168.79.255 Mask:255.255.255.0  
          inet6 addr: fe80::20c:29ff:fe4:b8a4/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
          RX packets:137 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:27831 (27.8 KB) TX bytes:10590 (10.5 KB)
```

```
dbradberry@ubuntu:~$ ifconfig eth3
```

```
eth3      Link encap:Ethernet  HWaddr 00:0c:29:f4:b8:ae  
          inet addr:192.168.79.130 Bcast:192.168.79.255 Mask:255.255.255.0  
          inet6 addr: fe80::20c:29ff:fe4:b8ae/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
          RX packets:146 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:25303 (25.3 KB) TX bytes:9296 (9.2 KB)
```

```
dbradberry@ubuntu:~$
```

```

root@yoshi: /home/dbradberry/veripy
root@yoshi:/home/dbradberry/veripy# ./bin/veripy \
> -c demo-vm.cfg \
> -fH /home/dbradberry/Desktop/vm.html \
> --subnet 0 Ethernet fe80::20c:29ff:fef4:b89a fe80::20c:29ff:fef4:b89a 00:0c:
29:f4:b8:9a \
> --subnet 1 Ethernet fe80::20c:29ff:fef4:b8a4 fe80::20c:29ff:fef4:b8a4 00:0c:
29:f4:b8:a4 \
> --subnet 2 Ethernet fe80::20c:29ff:fef4:b8ae fe80::20c:29ff:fef4:b8ae 00:0c:
29:f4:b8:ae \
> --subnet 3 Ethernet fe80::20c:29ff:fef4:b8b8 fe80::20c:29ff:fef4:b8b8 00:0c:
29:f4:b8:b8 \
> host

```

```

37 (20.3 KB)
4:b8:a4
8.79.255 Mask:255.255.255.0
4 Scope:Link
00 Metric:1
runs:0 frame:0
uns:0 carrier:0
90 (10.5 KB)
4:b8:ae
8.79.255 Mask:255.255.255.0
4 Scope:Link
00 Metric:1
runs:0 frame:0
uns:0 carrier:0
TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:25303 (25.3 KB) TX bytes:9296 (9.2 KB)

```

dbradberry@ubuntu:~\$

```

root@yoshi: /home/dbradberry/veripy
2011-11-26 10:07:15,406 INFO Sending ICMPv6 Echo Request, in a valid IPv6 packet
2011-11-26 10:07:15,582 INFO Attempting to retrieve response to ICMPv6 Echo Request.
2011-11-26 10:07:15,583 INFO Executing test case: Version Field (v5)...
2011-11-26 10:07:15,583 INFO Sending IPv6 packet header with invalid version of 5.
2011-11-26 10:07:16,673 INFO Attempting to retrieve response to invalid IPv6 packet.
2011-11-26 10:07:16,674 INFO Sending ICMPv6 Echo Request, in a valid IPv6 packet
2011-11-26 10:07:16,825 INFO Attempting to retrieve response to ICMPv6 Echo Request.
2011-11-26 10:07:16,827 INFO Executing test case: Version Field (v7)...
2011-11-26 10:07:16,828 INFO Sending IPv6 packet header with invalid version of 7.
2011-11-26 10:07:18,145 INFO Attempting to retrieve response to invalid IPv6 packet.
2011-11-26 10:07:18,148 INFO Sending ICMPv6 Echo Request, in a valid IPv6 packet
2011-11-26 10:07:18,325 INFO Attempting to retrieve response to ICMPv6 Echo Request.
Written /home/dbradberry/Desktop/vm.html.
root@yoshi:/home/dbradberry/veripy#

```

```

37 (20.3 KB)
4:b8:a4
8.79.255 Mask:255.255.255.0
4 Scope:Link
00 Metric:1
runs:0 frame:0
uns:0 carrier:0
90 (10.5 KB)
4:b8:ae
8.79.255 Mask:255.255.255.0
4 Scope:Link
00 Metric:1
runs:0 frame:0
uns:0 carrier:0

```

```

TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:25303 (25.3 KB) TX bytes:9296 (9.2 KB)

```

dbradberry@ubuntu:~\$

vm.html_files
















IIERCT Test Report

Configuration

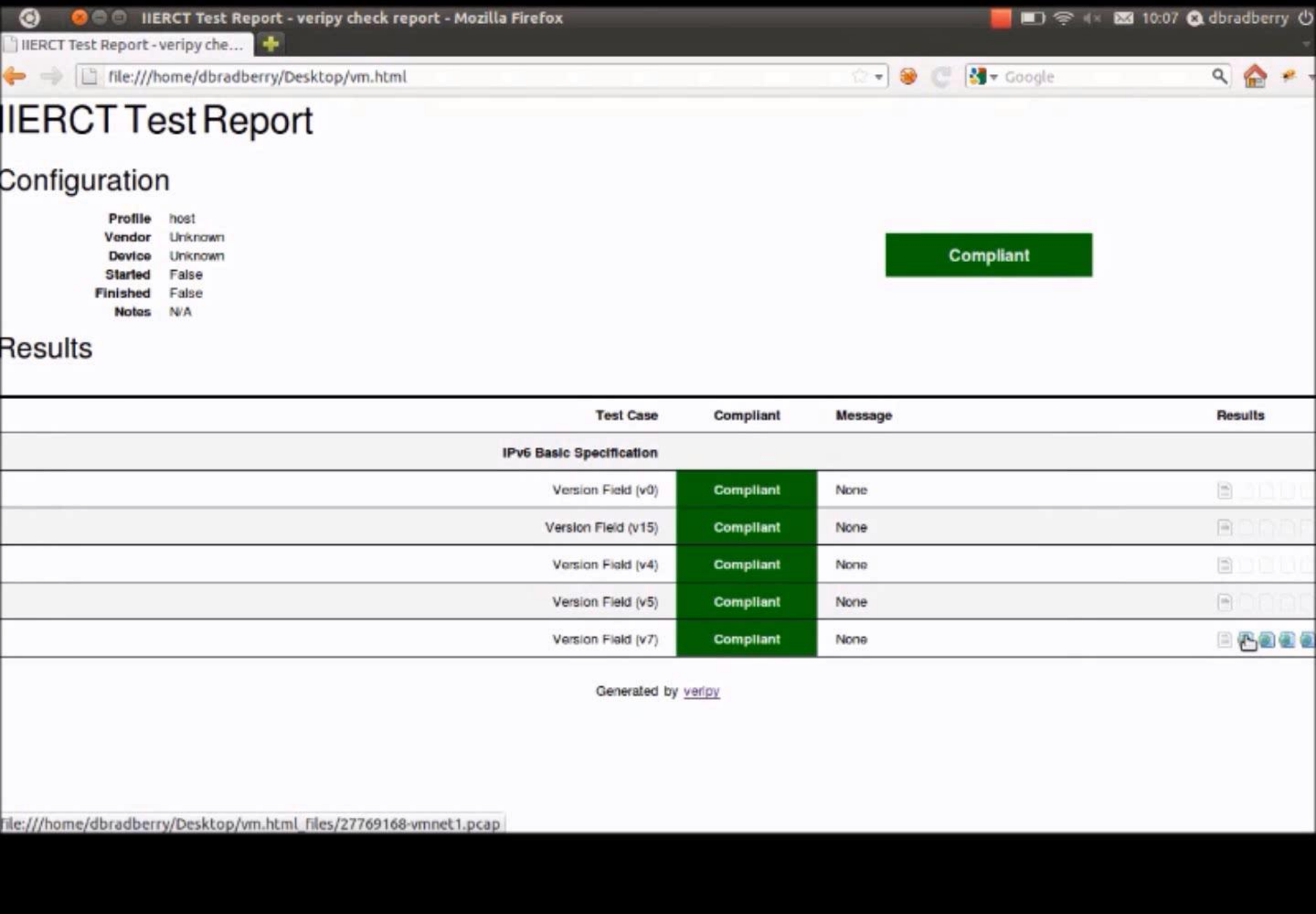
Profile host
Vendor Unknown
Device Unknown
Started False
Finished False
Notes N/A

Compliant

Results

Test Case	Compliant	Message	Results
IPv6 Basic Specification			
Version Field (v0)	Compliant	None	  
Version Field (v15)	Compliant	None	  
Version Field (v4)	Compliant	None	  
Version Field (v5)	Compliant	None	  
Version Field (v7)	Compliant	None	  

Generated by [veripy](#)


















IIERCT Test Report

Configuration

Profile host
Vendor Unknown
Device Unknown
Started False
Finished False
Notes N/A

Compliant

Results

Test Case	Compliant	Message	Results
IPv6 Basic Specification			
Version Field (v0)	Compliant	None	  
Version Field (v15)	Compliant	None	  
Version Field (v4)	Compliant	None	  
Version Field (v5)	Compliant	None	  
Version Field (v7)	Compliant	None	  

Generated by [veripy](#)

What does it mean?

Compliant

- The device behaved as prescribed in the relevant standards.
- If something goes wrong, all you need is your knowledge of IPv6.

Not Compliant

- The device did not behave as expected.
- If something goes wrong, you need in-depth knowledge of the operation of your equipment as well as the IPv6 standards.



96 more bits, no more magic

Conclusion

- The IPv6 transition will happen.
- Organisations need to plan for it now, even if there is no implementation yet.
- Your IPv6 strategy should first focus on training and ensuring that your network equipment will work in an IPv6 environment.

- veripy is a tool you should use as a part of implementing that strategy.

96 more bits, no more magic

Going Forwards



veripy isn't ready for primetime yet

10th October Project Started

3rd November First ICMPv6 Echo Request Sent

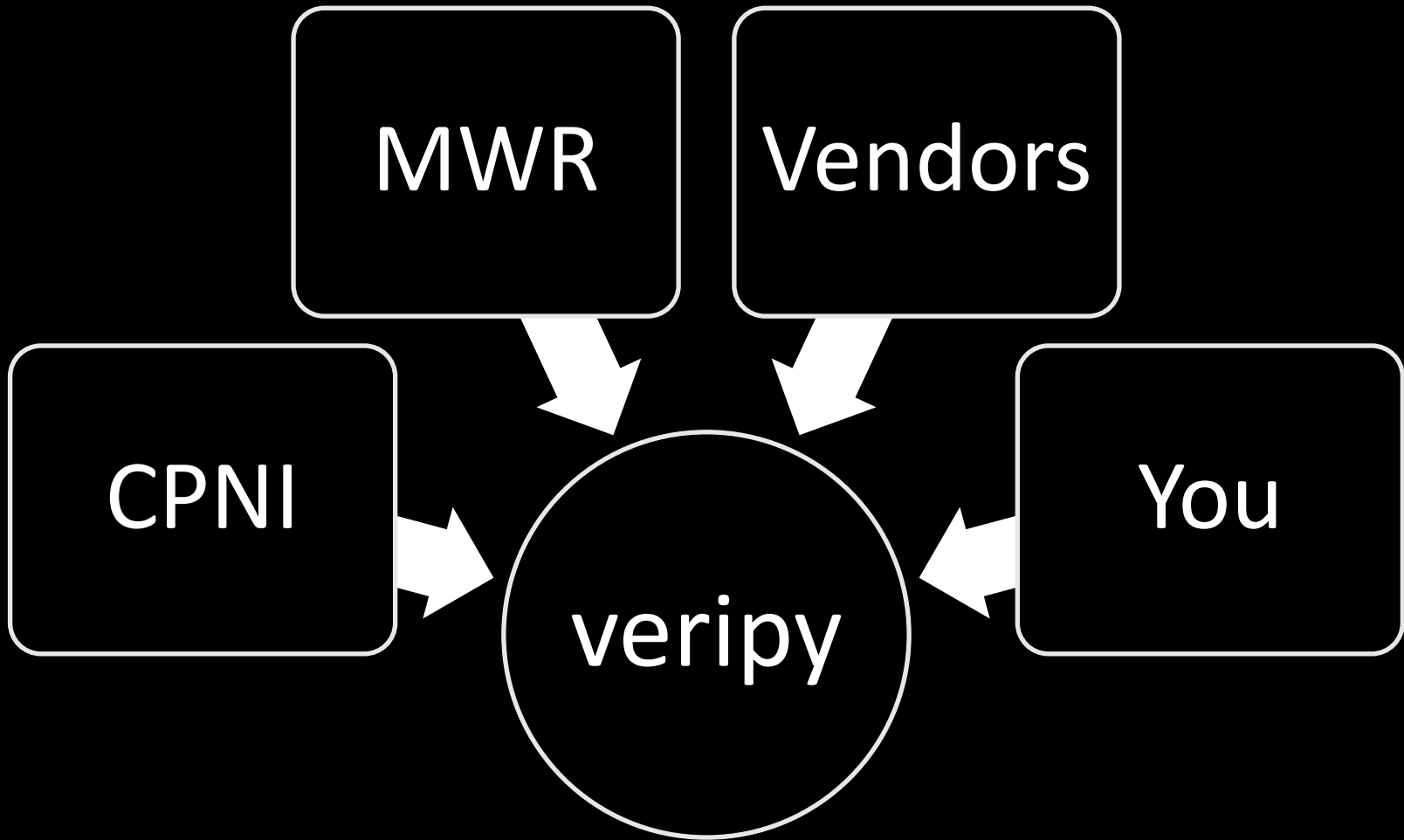
21st November First Completed Test Suite

30th November Demonstration to Software
Vendors

18th January Launch at MSoR

further development, demonstrations,
testing and promotion

16th March First Release

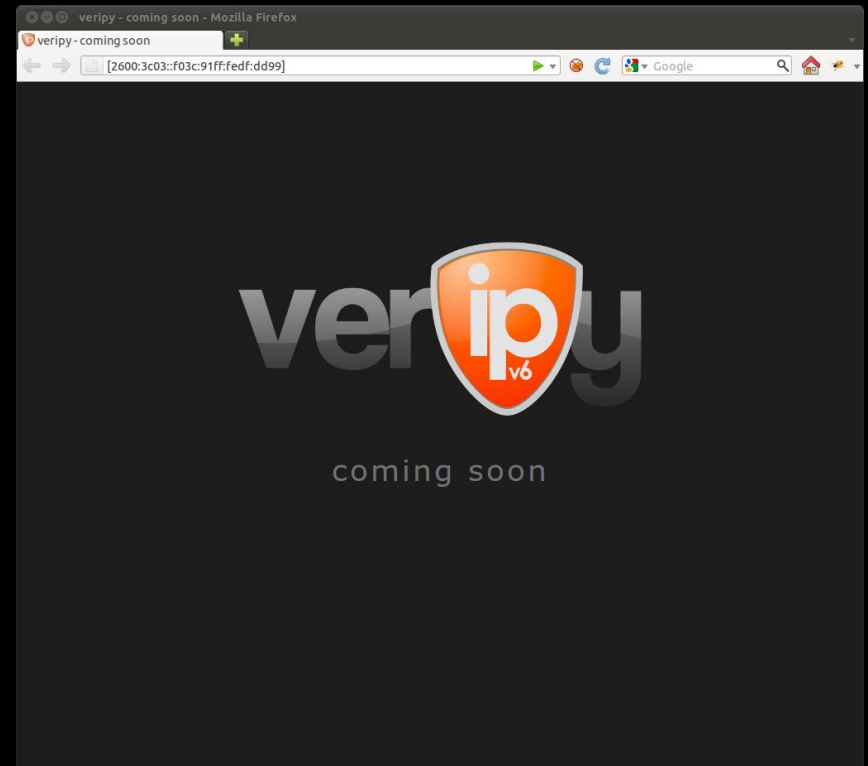




<http://veripy.org>

[2600:3c03::f03c:91ff:fedf:dd99]*

* also available over IPv4





Any Questions?

