Migration de l'infrastructure d'un établissement en double pile IPv4 / IPv6

Jérôme Berthier, Guillaume Cassonnet

Migration de l'infrastructure d'un établissement en double pile IPv4 / IPv6

Guillaume Cassonnet
Inria DSI Service Production
Centre de recherche Inria Bordeaux – Sud-Ouest
200 avenue de la Vieille Tour
33405 Talence Cedex

Jérôme Berthier
Inria DSI Service Conception d'Infrastructure
Centre de recherche Inria Bordeaux – Sud-Ouest
200 avenue de la Vieille Tour
33405 Talence Cedex

Résumé

Ce document présente les étapes suivies et les choix techniques effectués afin de déployer le protocole IPv6 à un niveau d'usage équivalent d'IPv4 sur l'infrastructure du centre de recherche Inria Bordeaux – Sud-Ouest.

Il aborde dans un premier temps les aspects liés à l'infrastructure réseau et le volet sécurité lié.

Nous décrivons ensuite les prérequis à traiter avant l'activation du protocole tant au niveau de l'organisation que de l'outillage technique pour l'exploiter.

Le cheminement de migration des serveurs et services est détaillé avant d'aborder la mise en œuvre de la connectivité des utilisateurs.

Nous proposons ensuite un focus rapide sur les dysfonctionnements identifiés avant de conclure par un bilan de ce déploiement.

L'activation du protocole IPv6 s'est étalée entre 2007 pour l'activation du préfixe RENATER de site et à fin 2018 pour sa généralisation aux services et utilisateurs. Trois intervenants Inria se sont succédés sans jamais être plus de deux à travailler sur le sujet. La problématique est longtemps restée un souhait d'ingénieur sans rencontrer de soutien officiel.

Il reste encore à ce jour divers éléments à activer : Wifi utilisateurs, parc Windows...

Mots-clefs

routage, WAN, IPv4, IPv6, sécurité, FHS, SLAAC, DHCPv6

1 Introduction

1.1 Pourquoi déployer le protocole IPv6 ?

Spécifié en 1998, le protocole IPv6 est l'alternative au protocole IPv4 dont l'espace d'adressage arrive à saturation.

Depuis 2017, l'ensemble des registres régionaux d'allocation d'adresses (RIR) ont entamé leur dernier bloc IPv4 de longueur 8 bits.

L'ARCEP[1] estime maintenant que le registre RIPE-NCC[2] atteindra la *date d'épuisement de ses ressources IPv4 le 05 novembre 2019 [1]*.

Cette situation se heurte à la croissance exponentielle des besoins en connectivité.

Ne rien engager pour supporter IPv6 reste une option pour les entités disposant de réserves IPv4 suffisantes. A l'échelle de l'Internet, la pénurie d'IPv4 implique une dégradation de la situation :

- ces ressources sont soumises à une situation de marché où l'offre est bien inférieure à la demande. L'explosion des prix associés constituerait un frein au développement.
- techniquement, l'usage des mécanismes de traduction d'adresses à grande échelle (Carrier Grade NAT (CGN)) [2] risque de se généraliser. Les utilisateurs partagent alors un espace restreint d'adresses IPv4 publiques. Leur trafic est parfois soumis à plusieurs couches de traduction, rompant toute possibilité de connectivité de bout en bout ou d'usage libre de sa connexion (nombre de sessions simultanées limitées, impossibilité d'héberger soi-même un service...).

L'adoption du protocole IPv6 reste lente. Beaucoup d'entités l'associent à un centre de coût et de complexité.

Point positif, on constate une réelle accélération de la prise en compte du protocole IPv6 ces dix dernières années au niveau de l'état mais surtout au niveau des grands fournisseurs d'accès et de contenu (voir Annexe 1).

D'un point de vue technique, les équipements et solutions intègrent majoritairement une compatibilité IPv6[3].

Depuis presque dix ans, les systèmes d'exploitation classiques[4] utilisent le protocole IPv6 lorsqu'il est disponible.

Par défaut, le protocole IPv6 est actif. Se pose alors le choix :

- de ne pas en tenir compte au risque de créer une dette technique et d'impacter la sécurité
- de mettre de l'énergie à désactiver le protocole[5]
- ou mieux, de mettre des moyens à gérer le protocole

Après une première phase d'implémentation réseau, il nous est apparu possible de généraliser un support d'IPv6 dans notre établissement.

L'objectif retenu est d'assurer la connectivité de nos expérimentations, de nos utilisateurs et de nos services publiés sur Internet (et même plus au final).

Afin de conserver une connectivité de bout en bout, nous avons privilégié un adressage IPv6 natif en excluant toute forme de traduction d'adresse type « NAT64 ».

Le double adressage IPv4 et IPv6 s'est imposé naturellement.

2 Ressources et infrastructure réseau

2.1 Allocation d'un préfixe IPv6 global

La préparation du déploiement commence par une demande d'attribution de préfixe IPv6 auprès du LIR RENATER[6] dans notre contexte.

1. Autorité de régulation des communications électroniques et des postes (ARCEP)
2. Registre régional pour l'Europe, le Moyen-Orient, la Russie...
3. Il est impossible de référencer ici l'ensemble des éléments concernés.
4. Exemples : Windows 7, MacOS 10.7, Linux : Fedora 13, Debian Squeeze...
5. Microsoft recommande de ne pas désactiver IPv6 par exemple ; https://blogs.technet.microsoft.com/netro/2010/11/24/arguments-against-disabling-ipv6/
6. Demande de préfixes IP. Renater; https://services.renater.fr/services_ip/allocation_d_adresses

Les échanges avec le GIP permettent de fiabiliser la cible technique (site concerné, NR…) avant d'engager la mise en production auprès du NOC et la délégation de la zone DNS reverse.

2.2 Plan d'adressage

L'élaboration d'un plan d'adressage permet de planifier la manière dont le préfixe attribué sera consommé.

Sur la base des documents RFC 6164 [3], RFC 6177 [4] et RFC 7421[5], nous implémentons trois longueurs de masque différentes :

- 128 bits pour les interfaces de type loopback
- 127 bits pour les réseaux d'interconnexion point à point entre équipements (y compris Ethernet)
- 64 bits pour tous les autres usages Ethernet

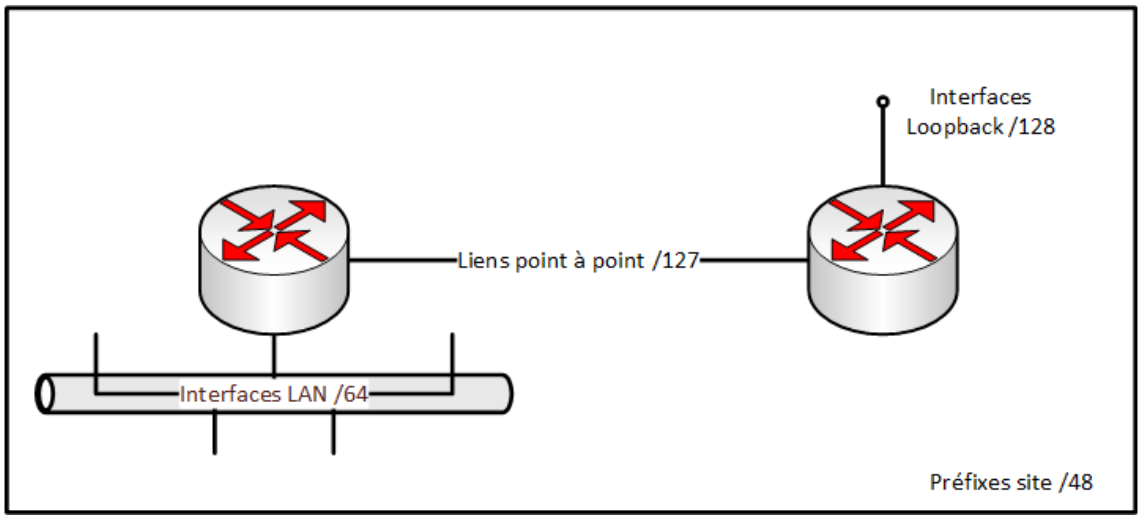


Figure 1: Masques standards d'adressage IPv6

Il existe de nombreux documents[7][8] détaillant des pratiques d'adressage IPv6.

En 2011, nous avons référencé pour chaque site Inria la structure d'adressage « 2001:660:<network_id_site_Inria>:AGTT:: »[9].

Nous appliquons la logique suivante :

- effectuer de l'allocation progressive par bloc /52 (site principal, antenne distante, grande expérimentation autonome…) en faisant varier (A) sur 4 bits
- regrouper les réseaux ayant un profil proche par bloc /56 (groupe de réseaux de confiance, groupe de réseaux externes, ...) (G) sur 4 bits
- provisionner 256 préfixes /64 dans une allocation donnée (A) et pour un groupe de réseau donné (G) pour les réseaux instanciés (T) sur 8 bits

7. Exemple : SurfNet, «PREPARING AN IPV6 ADDRESS PLAN » ; https://www.ripe.net/support/training/material/IPv6-for-LIRs-Training-Course/Preparing-an-IPv6-Addressing-Plan.pdf

8. Exemple : Infoblox, «IPv6 Addressing Plan Basics » ; https://www.infoblox.com/wp-content/uploads/2016/04/infoblox-whitepaper-ipv6-addressing-plan-basics_1.pdf

9. Proposition de Philippe Lubrano, architecte réseau au sein de la DSI Inria

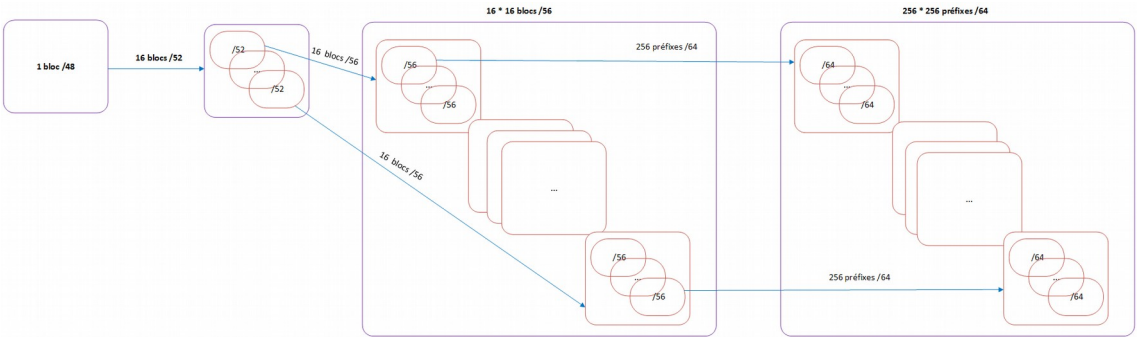


Figure 2: Expansion plan adressage IPv6

Après ce premier déploiement de site, il nous apparaît utile de revalider ce plan, notamment sur la gestion des groupes (G) que nous devrions calquer sur notre modèle de sécurité réseau décrit en 2018.

Le plan d'adressage retenu est ensuite à intégrer au référentiel SI à travers un outil de gestion d'adresses IP (IPAM) .

2.3 Routage et filtrage IP

Les fonctions de routage et filtrage réseau sont les premières briques à rendre compatible avec IPv6.

Dans notre contexte, les configurations proposées sont installées :

- sur des routeurs WAN Cisco famille ASR1000
- sur des pare-feux Juniper SRX1400

Les mécanismes utilisés sont standardisés donc a priori utilisables sur d'autres solutions propriétaires ou libres (exemples projet FRRouting[10] ou VyOS[11]).

10. Projet FRRouting (FRR) ; https://frrouting.org/

11. Projet VyOS ; https://vyos.io/

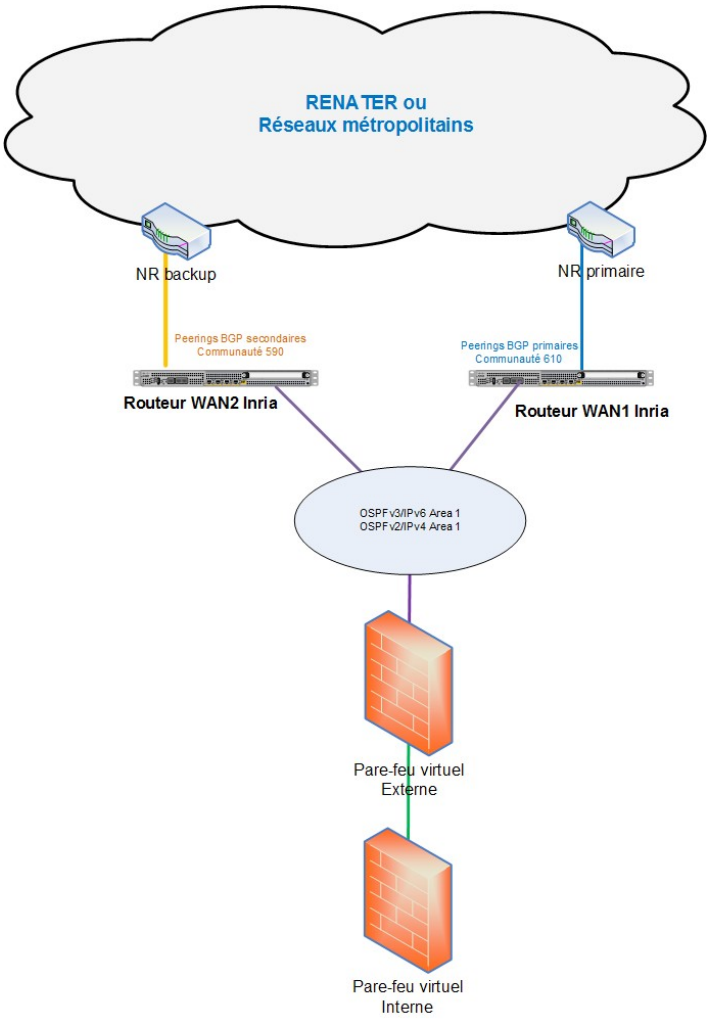


Figure 3: Architecture réseau IP site Inria niveau 3

2.3.1 Routage externe WAN RENATER

Comme pour IPv4, la présence RENATER du site est assurée en IPv6 par l'annonces de préfixes via BGP.

Une nouvelle interconnexion IPv6 est réalisée avec le NR sur un nouveau vlan afin d'établir un nouveau peering[12].

L'activation de la famille d'adresses IPv6 dans la fonction de routage BGP permet ensuite de traiter des annonces IPv6 sur ce nouveau point d'échange[13].

```
router bgp <ASN_Inria>
 neighbor 2001:660:<IPv6_NR_Renater> remote-as 2200
 neighbor 2001:660:<IPv6_NR_Renater> description RENATER IPv6
 neighbor 2001:660:<IPv6_NR_Renater> password 7 <secret>
  address-family ipv6
```

12. Supervision BGP IPv6 – voir Annexe 2

13. Le transport des annonces IPv6 pourrait être assuré par une interconnexion IPv4 et inversement.

```
  network 2001:660:<network_id_site_Inria>::/48
  neighbor 2001:660:<IPv6_NR_Renater> soft-reconfiguration
inbound
  neighbor 2001:660:<IPv6_NR_Renater> prefix-list prefix-v6 out
  neighbor 2001:660:<IPv6_NR_Renater> maximum-prefix 200 warning-
only
  neighbor 2001:660:<IPv6_NR_Renater> filter-list 1 out
  neighbor 2001:660:<IPv6_NR_Renater> activate
  neighbor 2001:660:<IPv6_NR_Renater>  filter-list 2 in
 exit-address-family
```

Les attributs de paramétrage BGP manipulés en IPv6 (network, neighbor, community, ASN filter list, route-map…) sont identiques à ceux manipulés en IPv4.

Sur le site Inria Bordeaux – Sud-Ouest, cette connectivité BGP/IPv6 a été initiée dès 2008 et revue en 2012 pour intégrer une architecture redondée par deux routeurs WAN.

2.3.2 Routage interne Inria

Dans la topologie WAN Inria, les échanges entre les deux routeurs du site et l'instance pare-feu externe sont assurés par une aire de routage OSPFv2 en IPv4.

Nous avons activé le protocole OSPFv3 qui permet de traiter les préfixes IPv6 en plus d'IPv4 de manière agnostique à la version du protocole.

Son transport est basé sur des échanges IPv6 au niveau du lien local. La sécurisation est à effectuer par un chiffrement IPSEC qui s'applique au lien.

La fonctionnalité Cisco OSPFv3 s'active de manière similaire à OSPFv2 avec l'ajout de la notion de familles d'adresses :

```
router ospfv3 1
 router-id <Ipv4_router_id>
 area 1 authentication ipsec spi 256 md5 7 <secret>
 !
 address-family ipv6 unicast
  passive-interface default
  no passive-interface Port-channel1.6
  no passive-interface Port-channel1.240
  distribute-list prefix-list ospf-filter-v6-in in
  distribute-list prefix-list ospf-filter-v6-out out connected
  distribute-list prefix-list ospf-filter-v6-out out static
  default-information originate
  redistribute connected
  redistribute bgp <ASN_Inria>
 exit-address-family
!
interface Port-channel1.x
ospfv3 1 ipv6 area 1
```

Afin de séparer les deux protocoles, nous avons conservé une répartition entre les deux familles d'adresses entre OSPFv2/IPv4 et OSPFv3/IPv6[14].

2.4 Sécurité au niveau 3

2.4.1 Filtrage réseau au niveau 4

Comme pour IPv4, il est incontournable d'appliquer une politique de sécurité réseau en implémentant le même cloisonnement pour IPv6 (voir Annexe 3).

Nous opérons deux niveaux de filtrage IP : l'un générique en bordure sur nos routeurs WAN et l'autre plus spécialisé sur nos instances pare-feux.

Le premier niveau consiste à épurer tout trafic clairement anormal car sourcées de nos propres préfixes, de classes d'adresses réservées pour des tests, utilisant des extensions dépréciées…

Le second niveau applique une politique de sécurité spécifique.

Nos pare-feux Juniper ont été déployés pour traiter le protocole IPv6 par défaut :

```
set security forwarding-options family inet6 mode flow-based
```

Ainsi, l'utilisation de la directive « any » dans les politiques s'applique à tout trafic IPv4 et IPv6.

Au démarrage, nous avons intégré les filtres de base notamment concernant les messages de contrôle ICMPv6 (RFC 4890 [6]). Les filtrages spécifiques ont ensuite été adaptés à chaque besoin au cours de l'implémentation des services en double pile.

2.4.2 Protection des équipements

L'activation d'IPv6 sur les équipements réseau de routage requiert quelques précautions en terme de sécurité (voir Annexe 3).

– Saturation cache ND

Un segment Ethernet est normalement associé à un préfixe /64. Cet immense espace d'adressage facilite la saturation du cache de voisinage sur les équipements.

L'usage de préfixe /127 pour les interconnexions point à point supprime ce risque.

En parallèle, il est possible d'affiner l'expiration du cache (fixée à 4 heures par défaut sur un routeur Cisco ou à 20000 entrées sur un pare-feu Juniper).

– Extensions IP dépréciées

Comme pour IPv4, les options de routage par la source ne sont pas souhaitables.

Le filtrage des paquets avec option « Routing header 0 » et la désactivation globale sont à appliquer sur les équipements.

En complément des protections IPS basiques (IP spoofing, TCP SYN flood, blocage des balayages de ports et machines…), il peut être intéressant d'activer la détection des paquets IPv6 malformés.

2.5 Sécurité à l'accès (First Hop Security)

2.5.1 Motivations

Fin 2014, suite à un rappel lors d'une formation IPv6 donnée par RENATER, nous décidons de corriger la vulnérabilité de nos réseaux à différentes attaques IPv6 existantes (problématique identifiée dès les premiers travaux sur le protocole).

14. Il a été tenté de traiter la famille d'adresses IPv4 aussi via ospfv3. Nous avons catégorisé le bug Cisco CSCue82043 - « OSPFv3 IPv4 neighborship goes down w/ IPsec authentication configured ». Ce bug est clos sans résolution prévue.

Après avoir mis en évidence certaines attaques dans un laboratoire avec *THC-IPv6 Attack Toolkit*[15], nous avons déployé certaines directives *First Hop Security (FHS)*[16] sur nos commutateurs de desserte Cisco.
Des fonctionnalités équivalentes existent chez d'autres constructeurs[17][18].

2.5.2 Vulnérabilités et mitigations

– **Rogue RA**

Par négligence ou malice, la présence de messages RA illégitimes sur le réseau peut conduire à une mauvaise configuration des clients (RFC 6104 [7]).

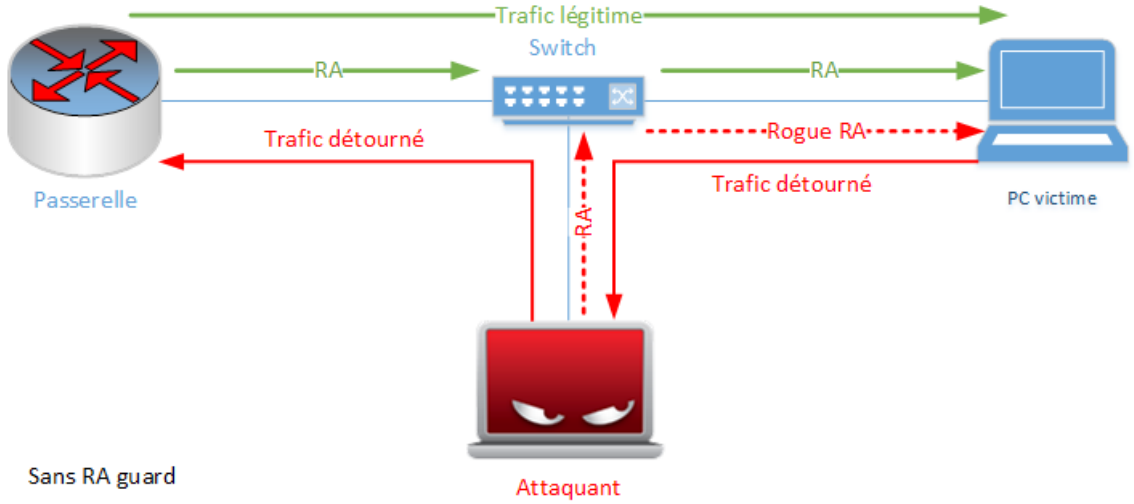


Figure 4: Attaque "rogue RA"

La contre mesure à adopter est *RA Guard* (RFC 6105 [8]) , activable en *FHS* avec la directive :

```
ipv6 nd raguard
```

Cette fonctionnalité empêche les hôtes de se comporter comme un routeur fournissant un adressage « *stateless* ».

15. © 2005-2019 vh@thc.org https://github.com/vanhauser-thc/thc-ipv6
16. Cisco IPv6 First Hop Security (FHS) https://www.cisco.com/c/en/us/products/ios-nx-os-software/ipv6-first-hop-security-fhs
17. Exemple HP, « IPv6 network defense ND snooping and detection » ; https://techhub.hpe.com/eginfolib/networking/docs/switches/WB/16-01/5200-0135_wb_2920_ipv6/content/index.html
18. Exemple Juniper, « IPv6 Acces Security » ; https://www.juniper.net/documentation/en_US/junos/information-products/pathway-pages/system-basics/security-services.html

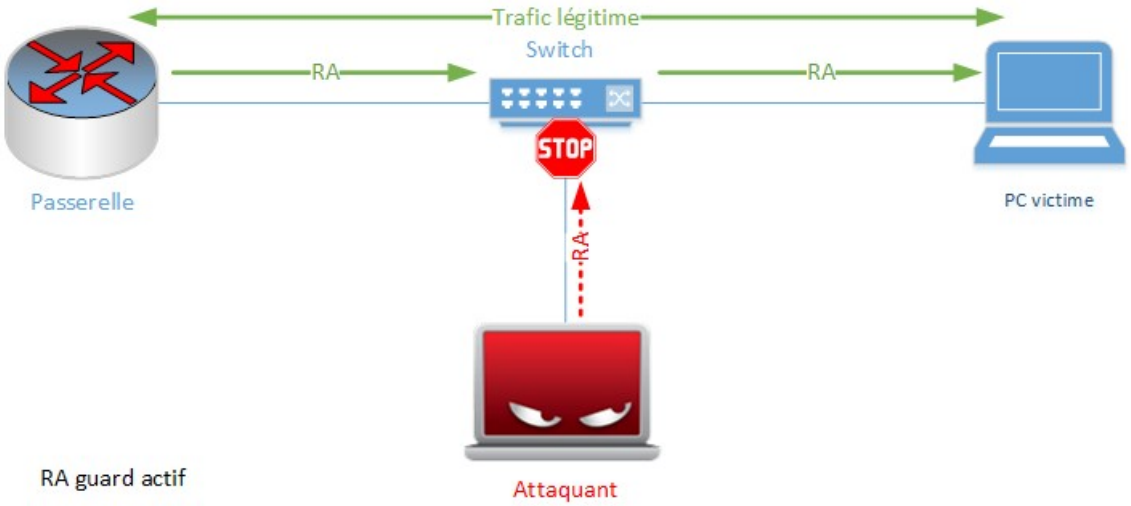


Figure 5: Attaque "rogue RA" - IPv6 FHS ra guard

Cette sécurité, si elle est indispensable, n'est pas parfaite (cf. RFC 7113 [9]).

– **Rogue DHCPv6 Server**

La présence sur le réseau d'un serveur DHCPv6 illégitime est un autre vecteur d'attaque indépendant du choix d'auto-configuration attendu (« stateful » dans ce cas).

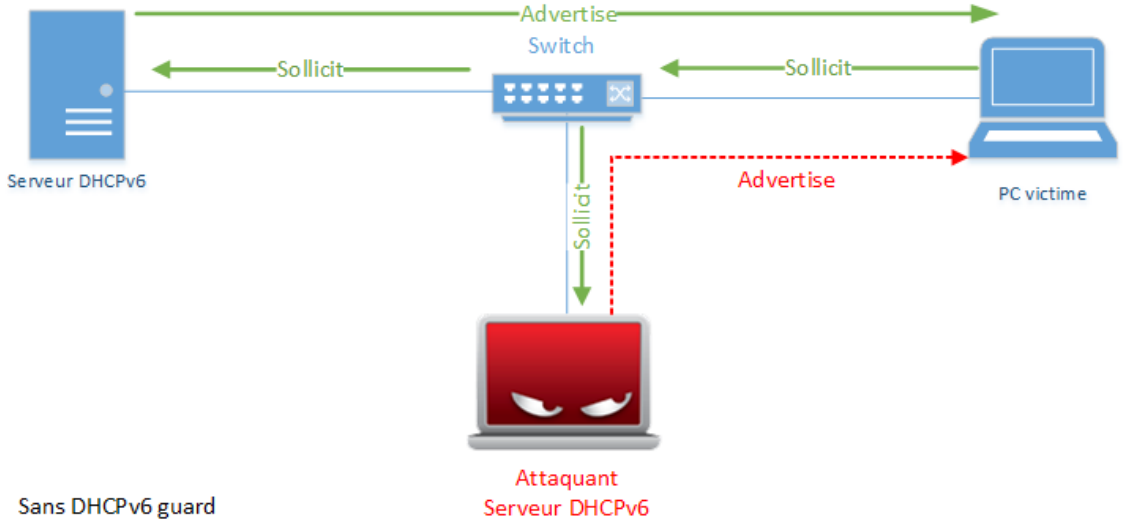


Figure 6: Attaque "rogue DHCPv6 server"

Afin de s'en prémunir, il faut appliquer une protection semblable *au « DHCP snooping IPv4 ». Cette protection est appelée « DHCPv6 shield » (RFC 7610 [10])* et implémentée sous Cisco FHS par la directive :

```
ipv6 dhcp guard
```

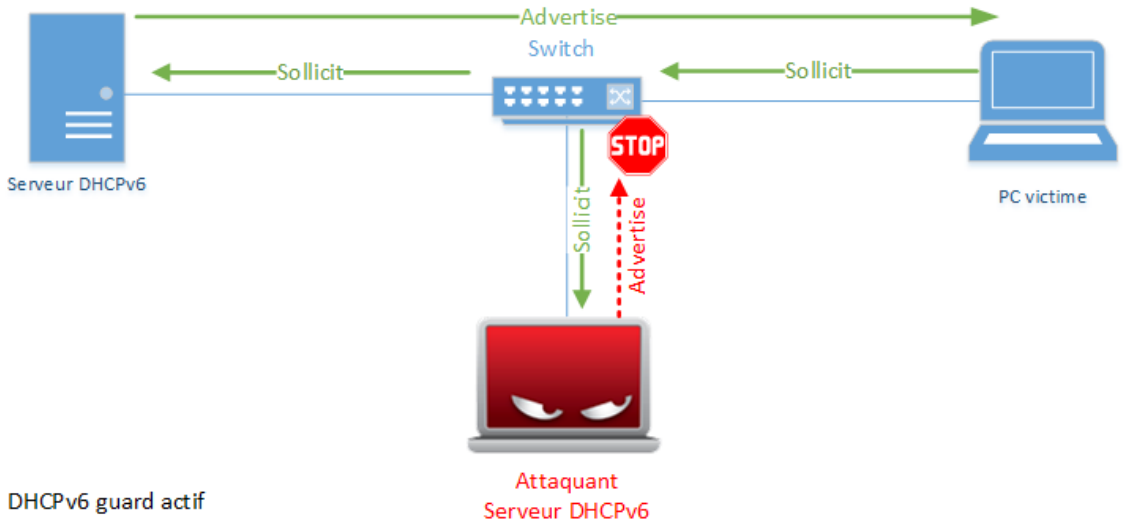


Figure 7: Attaque "rogue DHCPv6 server" - IPv6 FHS DHCPv6 guard

– **Neighbor Discovery Spoofing**

Le protocole NDP (RFC 4861 [11]) est vulnérable par pollution du cache du voisinage sur les hôtes. Cette technique permet d'usurper l'adresse d'un hôte pouvant conduire à une attaque type « Homme du milieu ».

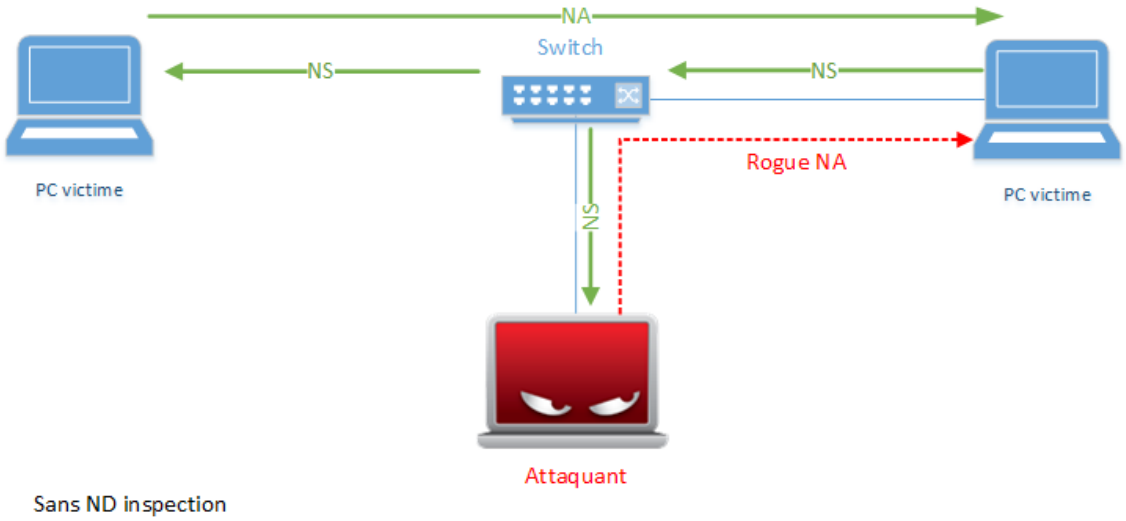


Figure 8: Attaque "pollution cache ND"

La contre-mesure consiste à activer une analyse dynamique des messages NS et NA :

```
ipv6 nd inspection
```

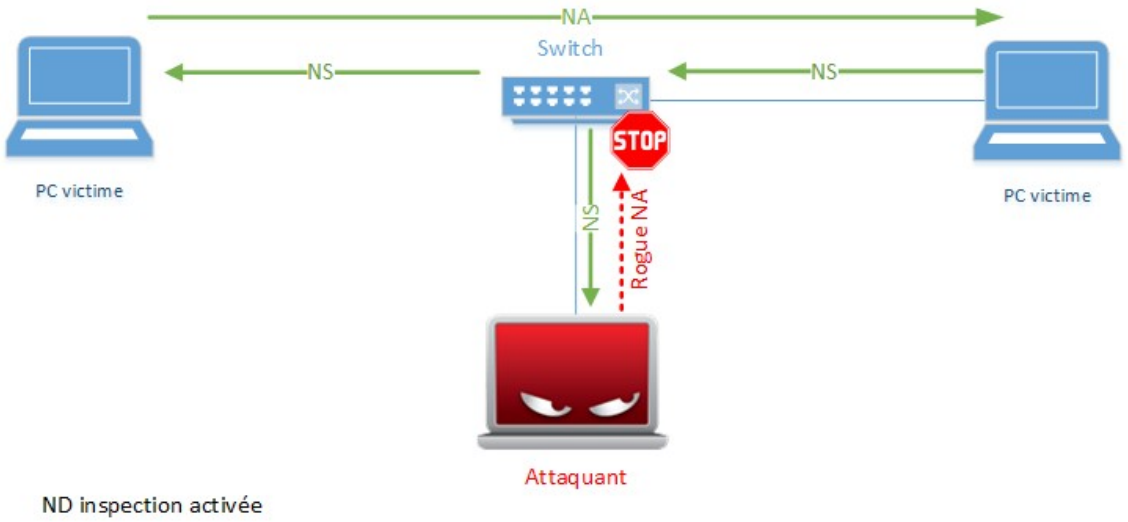


Figure 9: Attaque "pollution cache ND" - IPv6 FHS IPv6 nd inspection

– **DAD DoS**

Le protocole NDP est aussi vulnérable par détournement de son mécanisme DAD à un déni de service IPv6, empêchant toute auto-configuration sur le réseau. La directive *FHS* correspondante est :

```
ipv6 snooping
```

Cette fonction va apprendre les adresses IPv6 négociées au travers de chaque port du commutateur, et empêcher une machine d'émettre des messages frauduleux pour une autre adresse.

2.5.3 Déploiement

Avant de pouvoir utiliser les directives *FHS* sur les commutateurs Cisco, il faut utiliser un *Switch Database Management template (SDM)*[19] qui alloue suffisamment de ressources à IPv6. Sur IOS 12.2 et 15.0, nous avons activé le profil « dual-ipv4-and-ipv6 » :

```
sdm prefer dual-ipv4-and-ipv6 default
```

Nous utilisons déjà des *macro* Cisco pour faciliter la configuration des ports de desserte. L'intégration des directives de sécurité IPv6 s'est fait en complétant ces macros. Pour la branche 12.2 de Cisco IOS, ne supportant pas les FHS, nous avons traité les protections avec des filtres IPv6 appliqués sur le port :

19. Profil d'attribution des ressources du commutateur.

– IOS 12.2

```
ipv6 access-list inria-desktop_ipv6-in
  deny icmp any any router-advertisement
  deny udp any eq 547 any eq 546
  permit ipv6 any any

macro name inria-desktop
  !... # macro existante
  ipv6 traffic-filter inria-desktop_ipv6-in in
@
```

– IOS 15.0

```
macro name inria-desktop
 !... # macro existante
 ipv6 nd inspection
 ipv6 nd raguard
 ipv6 dhcp guard
 ipv6 snooping
@
```

3 Prérequis à la généralisation

Avec un peu de recul sur notre activation IPv6, nous vous proposons une liste de prérequis utiles afin de faciliter le projet.

3.1 Considérations organisationnelles

Il existe quelques points à prendre en compte avant de se lancer dans les aspects techniques :

– Un déploiement IPv6 à grande échelle doit se faire avec l'*approbation de la direction*[20]**.** En effet, IPv6 n'est pas juste un nouveau projet pour le catalogue de services, mais une nouvelle composante que tous les personnels de la DSI devront manipuler et comprendre. Il est important de sensibiliser et d'obtenir le soutien de la direction pendant toute la durée de mise en œuvre de l'accès aux services en *double pile*.

– Sauf en cas d'incompatibilité, *tout nouveau projet* informatique doit intégrer la *composante IPv6 dès sa conception*.

– Pour les *services déjà existants*, il faut *mesurer les impacts*, l'opportunité et les difficultés de les porter sur IPv6.

– Des *formations* sont nécessaires pour les *personnels de la DSI* manipulant de près où de loin IP dans le cadre de leur travail. On y abordera, entre autres :

  – Le concept de *double pile* et le comportement des applications clientes.
  – La logique associée au plan d'adressage IPv6 (préfixe et sous-réseau).

20. Dans notre cas, elle est venue très tardivement. L'aspect réseau a été avant tout une problématique traité entre ingénieurs.

  - La structure des adresses IPv6 unicast, notamment la partie interface.
  - Les idées fausses couramment associées à IPv6 : sécurité réduite par absence de NAT, la différence entre SLAAC et DHCPv6, etc.
- En complément du point précédent, les *documentations* et *référentiels* impactés devront être mises à jour. De nouvelles documentations propres à IPv6 sont à créer pour compléter la formation des personnels : outils de vérification, de debug, etc.

3.2 Outils d'administration

Avant de démarrer, il a été nécessaire d'opérer une revue de l'outillage technique sur lequel intégrer IPv6.

Le système d'information technique manipulant les adresses IP est un élément clé à vérifier : réservation IP, DNS, filtrages IP...

- Monitoring et visibilité réseau

Les protocoles de collecte d'empreintes réseaux IPFIX ou NetFlow sont à activer explicitement pour IPv6. L'équipement réseau et le collecteur doivent être compatibles *NetFlow 9* (RFC 3954 [12]) ou *IPFIX* (RFC 7011 [13] et 7012 [14]). Nous planifions d'utiliser *nfdump*, remplaçant *flow-capture* qui ne supportait pas la spécification *NetFlow 9*.

Les outils de *monitoring* doivent pouvoir surveiller les hôtes et services en IPv6. Nous utilisons N*agios*, qui ne permet pas de renseigner d'hôte double pile. Il faut dupliquer la configuration en créant un nouvel hôte associé à l'adresse IPv6 du serveur.

- Outil de gestion des adresses IP et zones DNS

Il faut que les outils et processus d'exploitation du DNS soient compatibles avec IPv6.

Au démarrage du projet, le référencement des adresses IP était directement réalisé par la gestion des zones DNS. Des allocations statiques étaient gérées dans les fichiers de configuration Bind. Ces changements étaient versionnés dans un dépôt git et orchestrés via puppet.

Nous utilisons maintenant *phpIPAM et PowerDNS*[21] qui supportent IPv6.

Côté *DNS opérationnel*, nous avons récupéré auprès de RENATER la gestion de notre zone reverse IPv6[22].

- Gestion des filtrages pare-feux

Nous exploitons les *filtrages pare-feux spécifiques* à travers l'outil Juniper *JunOS Space Security Director*.

Celui-ci est compatible IPv6 mais ne permet pas de manipuler d'objets *double pile*. Nous avons dû créer en parallèle les objets IPv6 correspondant aux objets IPv4 existants, les associer dans un « groupe », et utiliser ce groupe dans les règles. Ceci alourdit l'exploitation au quotidien.

- Orchestration système

Un dernier point critique pour la réussite de l'intégration IPv6 dans une infrastructure est l'outillage de gestion automatisé des serveurs.

Pour nos *déploiements* GNU/Linux *CentOS*, nous avons adapté les fichiers *kickstart* pour constituer un nouveau profil par défaut double pile.

21. Pierre Bénard, Inria DSI. Article « "PHPIpam/PowerDNS" : une solution IPAM/DNS moderne et open source », Décembre 2017; https://2017.jres.org/fr/presentation?id=93

22. Le transport IP d'une zone DNS IPv6 n'est pas dépendant du protocole et peut être assuré en IPv4 et inversement.

Nous utilisons *Ansible* et *Puppet* comme *gestionnaires de configuration* des serveurs GNU/Linux. Ces outils se sont révélés d'une grande aide pour réaliser les activations IPv6, notamment des services, de façon aisée et maîtrisée.

Pour les machines Windows, la politique Inria est de désactiver globalement IPv6 sur tous les postes de travail, mais une simple modification par *GPO Active Directory* permettra d'activer IPv6 en toute simplicité.

4 Activation IPv6 sur les réseaux de serveurs

Cette partie détaille le processus d'activation IPv6 sur un réseau de serveurs. Elle se compose de deux étapes séquentielles :

- la configuration d'un réseau (routeur et pare-feu)
- la configuration des serveurs et des services qu'ils hébergent.

4.1 Adressage statique de VLAN

Nous avons choisi d'utiliser des adresses IPv6 statiques pour les serveurs, comme c'était le cas en IPv4. Le routeur du VLAN n'envoie pas de messages RA. On ajoute simplement une adresse IPv6 globale au routeur Juniper sur son interface dans le VLAN :

```
set interfaces reth0 unit 555 family inet6 address
2001:660:<network_id_site_Inria>:555::1/64
```

Pour les adresses IPv6 des serveurs, par défaut, nous reportons les 8 derniers bits de l'IPv4 (écriture décimale) dans les 16 derniers bits de l'IPv6 (écriture hexadécimale)[23]. Ainsi, un serveur en *x.x.x.156* devient *2001:600:<network_id_site_Inria>:x::156* .

4.2 Mise à niveau filtrages pare-feu du VLAN

Après activation du routage, s'il en existe, on met à jour les filtrages pare-feu applicables globalement au réseau IPv4 du VLAN en y ajoutant le préfixe IPv6 correspondant.

Le filtrage fin par hôte est adapté lors de l'étape suivante de migration du serveur et services.

4.3 Activation IPv6 d'un serveur et de ses services

Nos serveurs hébergent chacun un seul service. Nous proposons une procédure pour y activer IPv6 en minimisant le temps d'indisponibilité.

1. Déterminer l'adresse IPv6 du serveur. *Ne pas la déclarer dans le DNS à ce stade.*
1. Configurer l'adresse IPv6 sur le système d'exploitation du serveur (voir Annexe 4).
2. Configurer le service en IPv6 (voir Annexe 5).
3. Valider la connectivité IPv6 au service :
   - En local, vérifier les sockets TCP/UDP d'écoute, les *TCP Wrappers*, le pare-feu et les possibles filtrages applicatifs par IP.
   - Depuis une machine cliente située dans un réseau privilégié (accédant à tout), vérifier la connectivité IPv6 au service : par capture réseau, ou plus simplement sans pile IPv4. Le serveur étant absent du DNS, on utilise directement son adresse IPv6 pour joindre le

23. Un hôte adressé uniquement en IPv6 le serait aléatoirement en l'absence de correspondance IPv4.

service. Pour les services dépendant du nom DNS (certificats SSL, VirtualHost, etc.), on modifie le fichier *hosts* de la machine cliente.

4. Dans les mêmes conditions, valider le bon fonctionnement du service à l'aide d'un jeu de test client le plus complet possible : connexion, utilisation courante, fonctionnalités avancées.

À ce stade, selon les résultats des tests, on fait le choix de rendre le service disponible en IPv6.

5. Mettre à niveau les filtrages pare-feu concernant le serveur et le service pour offrir une *connectivité identique sur les deux protocoles.*
6. Intégrer l'adresse IPv6 du serveur dans les outils de supervision. *Tous les services surveillés en IPv4 doivent également l'être en IPv6.*
7. Enfin, *déclarer l'adresse IPv6 du serveur ou service sur le DNS faisant autorité (enregistrement AAAA et PTR en général)*. Cette action va « mettre en production » le service en IPv6. Les requêtes IPv6 devraient commencer à peupler les logs d'accès.
8. Mettre à jour les documentations et référentiels concernant le service.

4.4 Bilan Serveurs

À la fin du projet, sur 106 serveurs dans les réseaux où IPv6 a été activé,

- 70 % sont désormais fonctionnels en *double pile* IPv4 / IPv6.
- 19 % n'ont pas été migrés car trop complexes, bien que compatibles avec IPv6.
- 7 % étaient incompatibles avec IPv6
- 4 % ont été jugés non pertinents à migrer (fin de vie, etc.)

C'est donc 89 % de nos services qui sont compatibles avec IPv6, prouvant la maturité de la spécification dans une grande diversité d'applications. Les boites noires type « appliances » et services infogérés constituent une bonne partie des 11 % restants.

5 Activation IPv6 sur les réseaux utilisateurs

5.1 Méthode d'adressage sans état

Par simplicité, nous avons choisi de déployer les réseaux utilisateurs en auto-configuration IPv6 sans état par la méthode « SLAAC ».

Ce mécanisme automatise l'attribution d'identifiant d'interface au niveau de l'hôte en fonction du préfixe global reçu du routeur.

Le format d'adresse générée est libre mais nous recommandons l'usage de la méthode RFC 7217[15] (voir Annexe 6) .

Ce type d'adressage dit « stable-privacy » allie la stabilité d'une adresse IPv6 unique par préfixe global avec la discrétion d'une adresse aléatoire.

Cet aléa inhibe la traçabilité d'un hôte unique repérable par son adresse EUI64[24] dérivée de son adresse MAC. Cela permettrait aussi de limiter l'efficacité des scans réseaux ciblés.

5.2 Impératifs de traçabilité

Afin de répondre aux impératifs de traçabilité d'usage de nos réseaux, nous avons besoin de conserver une association IP – terminal (ie adresse MAC).

Pour cela nous activons sur les commutateurs de desserte Cisco le suivi des périphériques IPv6 avec la directive globale :

24. http://livre.g6.asso.fr/index.php/Identifiant_d'interface#EUI-64

```
ipv6 neighbor binding logging
```

Les entrées envoyées au serveur syslog associent les attributs « A = adresse IPv6, V = vlan, I = port commutateur, M = adresse MAC » :

```
Sep 27 14:49:26 commut001.bordeaux.inria.fr 296677: Sep 27
14:49:25.608 MET: %SISF-6-ENTRY_CHANGED: Entry changed
A=2001:660:xxxx:xxxx:54DB:2B9:725A:1B5C V=814 I=Gi6/0/31 P=0005
M=685C.329A.DBB5
```

5.3 Information aux partenaires

Certains services hébergés en dehors de Inria sont basés sur des listes d'accès IPv4. Nous avons donc fourni à des partenaires et prestataires les plages réseau IPv4 hébergeant nos utilisateurs.

Si ces services extérieurs sont joignables en IPv6, il est important de fournir à ces partenaires les plages IPv6 correspondantes. A défaut, le service ne sera pas correctement rendu à nos utilisateurs connectés en double pile.

5.4 Activation IPv6 sur un réseau utilisateurs

Le déploiement est d'abord initié de façon statique, sans messages RA, afin de valider le bon fonctionnement depuis une machine de test.

1. Ajouter au routeur une adresse IPv6 globale sur son interface dans le VLAN. *Ne pas activer les messages RA à ce stade.*
2. Mettre à niveau les filtrages pare-feu concernant les réseaux IPv4 du VLAN : y ajouter le préfixe IPv6 correspondant.
3. Depuis une machine configurée en IPv6 statique dans le VLAN, vérifier le bon fonctionnement du routage IPv6 et de différent services internes et externes accessibles en IPv6.

Si tout est fonctionnel, on peut activer IPv6 pour tous les utilisateurs du VLAN.

Pour activer l'auto-configuration SLAAC sur nos routeurs internes Juniper, il faut explicitement ajouter l'interface du VLAN à la directive globale « router-advertisement » , avec le réseau à annoncer :

```
set protocols router-advertisement interface reth0.777 prefix
2001:660:<network_id_site_Inria>:777::/64.
```

Nos routeurs Juniper SRX1400[25] ne permettent pas de transporter des configuration DNS resolvers dans les annonces RA comme spécifié dans le RFC 8106 [15].

Nos clients reçoivent la configuration DNS complémentaire via DHCP IPv4 afin de pouvoir effectuer les résolutions de noms.

5.5 Bilan utilisateurs

25. Limitation logicielle non existante sur les gammes plus récentes utilisées sur d'autres sites Inria.

Nous avons choisi de réaliser l'activation finale en informant les utilisateurs, les incitant à contacter leur support informatique pour tout souci de connectivité.

L'opération s'est déroulée en toute transparence. On compte environ 150 postes uniques quotidiens sur ces réseaux.

6 Dysfonctionnements possibles

6.1 Cohérence de l'offre de services

L'implémentation du protocole IPv6 doit être pensée de manière transverse afin d'être compatible avec l'ensemble de l'offre de services de l'établissement.

Dans le cas du centre Inria Bordeaux – Sud-Ouest, la migration a été opérée de manière large et exploratoire en incluant aussi des services purement locaux et internes.

Des dépendances avec des services Inria nationaux n'ont pas été identifiées initialement.

C'est par exemple le cas de l'offre VPN SSL utilisateurs qui ne supporte pas encore IPv6.

Une publication d'un service double pile (DNS A et AAAA par exemple) peut poser problème si le client dispose d'une connexion IPv6 native hors VPN et d'une connexion VPN IPv4.

Selon l'application cliente, le protocole IPv6 sera préféré, routé via la connexion native sans possibilité de joindre la ressource interrogée (puisque non accédée par le VPN).

Il convient donc d'être très vigilant sur l'horizon de présentation de services en IPv6 afin de s'assurer qu'ils seront joignables.

L'usage de services externes type « SaaS » peut aussi poser problème s'ils disposent eux même d'une publication double pile IPv4 / IPv6.

C'est notamment le cas d'un service PeopleDoc qui s'appuie sur le fournisseur CloudFlare en double pile.

Nos préfixes IPv6 n'ont pas été intégré lors de la mise en œuvre du contrat. Le comportement attendu depuis nos préfixes IPv4 n'est donc pas constaté en IPv6 ce qui dégrade l'expérience utilisateur[26].

6.2 Cohabitation IPv4 et IPv6

Comme on vient de l'évoquer, l'usage parallèle des deux protocoles peut ne pas être sans conséquence.

Concernant l'usage Web, les navigateurs par exemple implémentent tous l'algorithme « Happy Eyeballs » (RFC 6555 [17]) qui permet de tester en parallèle le fonctionnement IPv6 et IPv4 d'un service.

Si la connectivité IPv6 est correctement établie, elle est préférée. Dans le cas contraire, la connectivité IPv4 est retenue dans un délai suffisamment court pour éviter tout désagrément à l'utilisateur.

Si un service hébergé sur un serveur double pile n'est pas joignable en IPv6 (routage impossible ou perte de paquets...), certains clients peuvent rencontrer des difficultés.

À l'opposé des navigateurs Web utilisant les « Happy Eyeballs », on constate des comportements inadaptés à la problématique double pile. Parmi les cas les plus extrêmes, on peut citer :

- OpenSSH client, avec un timeout de 2 minutes avant de passer en IPv4.
- Le client NFS sur GNU/Linux (nfs-utils), avec un timeout de 3 minutes avant de passer en IPv4, et en partant de la version la plus haute vers la plus basse du protocole, multipliant potentiellement l'attente...

7 Bilan

26. Ce dysfonctionnement a été provisoirement contourné par le blocage du trafic IPv6 à destination des ressources CloudFlare correspondantes.

7.1 Empreinte sur le trafic

Sur le site Inria Bordeaux – Sud-Ouest, nous mesurons actuellement sur le protocole IPv6 : 13,2 % du trafic IP entrant et 9,3 % du trafic IP sortant.

Ces chiffres sont en augmentation par rapport à nos dernières mesures en octobre 2018. Ils devraient encore progresser significativement avec une future activation IPv6 sur certains réseaux Wi-Fi, doublant le nombre de périphériques utilisateurs.

7.2 Impact sur le support

Un an après la fin de ce déploiement, aucune complication liée à IPv6 sur les postes de travail n'a été remontée à travers le support.

Un cas de filtrage applicatif a dû être ajusté sur un serveur Apache, en ajoutant le préfixe IPv6 d'un partenaire pour leur permettre l'accès.

Enfin, un serveur de fichiers NetApp, dont la résilience de configuration IPv6 n'avait pas pu être confirmée par un reboot, s'est retrouvé sans configuration IPv6 après un arrêt électrique.

7.3 Perspectives d'évolution

Ce premier déploiement fonctionnel a permis de jouer une migration représentative des enjeux au niveau utilisateurs et fournisseurs de service.

Une priorité va maintenant être de capitaliser sur ces éléments pour les généraliser au niveau de l'institut.

Fournir par défaut une connectivité IPv6 pour nos utilisateurs et une présence IPv6 pour nos services nous permettra d'accompagner sereinement la saturation de l'espace d'adressage IPv4.

Les deux protocoles vont certainement cohabiter très longtemps sur Internet. Implémenter IPv6 permet d'être prêt à répondre aux usages à venir.

Annexe

Annexe 1 – Un peu plus de contexte IPv6

Il est difficile de dresser un état des lieux exhaustifs de la prise en compte du protocole IPv6. Certains points ci-dessous peuvent constituer des signaux positifs à prendre en compte.

Du point de vue de l'*État français* par exemple, une circulaire[27] du 08 décembre 2011 co-signée par les Ministres de l'Intérieur, de l'Économie et de l'Industrie demande à *intégrer la compatibilité IPv6 dans les services de l'état, notamment les sites Internet à destination du public.*

Entre 2007 et 2017, la question de la pénurie d'IPv4 et du déploiement d'IPv6 a été abordée neuf fois à l'Assemblée nationale (a priori aucune mention depuis 2017).

Plus dernièrement, l'ARCEP a décidé d'initier la création d'un groupe de travail sur la transition IPv6[28], co-piloté avec Internet Society, afin d'associer les acteurs qui le souhaitent (opérateurs, hébergeurs, entreprises, secteur public, etc.).

Sur le plan technique, en juin 2012, l'initiative « World IPv6 Launch » a prouvé à grande échelle que l'activation permanente du protocole IPv6 est possible au niveau des constructeurs de matériels, des fournisseurs d'accès et des opérateurs de sites Web. A cette occasion, RENATER a activé le support IPv6 sur son service Antispam[29].

Ainsi, depuis plusieurs années, de grands fournisseurs de contenu (Google, Facebook...) portent un adressage en double pile IPv4 / IPv6.

De même, la majorité des fournisseurs d'accès Internet fixes et mobiles propose maintenant de manière transparente un adressage en double pile IPv4 / IPv6 à leurs abonnés.

Dans le contexte de la recherche et de l'enseignement supérieur, nous avons depuis longtemps l'opportunité de profiter des services du réseau RENATER[30] qui implémente et fournit une connectivité IPv6 depuis 2002 !

Enfin, il existe de très nombreux supports de formation notamment en ligne. Citons le travail historique de l'association G6[31] ou plus récemment l'occurrence d'un MOOC « Objectif IPv6 »[32] proposé sur la plateforme FUN.

Tous ces éléments sont autant d'encouragements à intégrer IPv6 dans les solutions déployées.

Annexe 2 – Supervision BGP IPv6

La supervision du service est réalisée via le protocole SNMPv3 AuthPriv pour vérifier l'état de chaque peering.

Le serveur doit disposer de la MIB CISCO-BGP4-MIB :

- lien : ftp://ftp.cisco.com/pub/mibs/v2/CISCO-BGP4-MIB.my
- à copier sous /usr/share/snmp/mibs/ sous CentOS 7

L'état de chaque peering BGP est récupérable via une requête snmpget sur oid 1.3.6.1.4.1.9.9.187.1.2.5.1.3.<chaine_a_recuper> pour les peerings IPv6.

Pour les peerings IPv6, l'identifiant du peering ne correspond donc pas à l'adresse du peer. Il est nécessaire d'en extraire l'oid cible :

27. https://ciscocollectivitesblog.files.wordpress.com/2012/05/ipv6-circulaire-8-dec2011-cir_34250.pdf

28. Arcep « Task-Force IPv6 »; https://www.arcep.fr/la-regulation/grands-dossiers-internet-et-numerique/lipv6/suivi-epuisement-adresses-ipv4/appel-a-candidatures-pour-former-une-task-force-ipv6-en-france.html

29. https://services.renater.fr/antispam/projet_ipv6

30. La connectivité IPv6 est aussi garantie au-delà à travers le réseau pan-européen Geant.

31. http://livre.g6.asso.fr/

32. https://www.fun-mooc.fr/courses/course-v1:MinesTelecom+04012+session05/about

– faire une requête snmpwalk sur oid cbgpPeer2State.ipv6 afin d'identifier les peerings IPv6 établis. Pour identification, l'IP du peer est formatée par octet.

```
snmpwalk -v3 -a SHA -A <auth> -l authPriv -u <user> -x AES -X
<priv> <IP_routeur> cbgpPeer2State.ipv6


CISCO-BGP4-
MIB::cbgpPeer2State.ipv6."20:01:06:60:XX:XX:00:XX:00:01:00:00:00:
00:00:01" = INTEGER: established(6)
```

– relancer cette requête snmpwalk sur oid cbgpPeer2State.ipv6 en ajoutant l'option « On » afin d'identifier l'oid spécifique de chaque peering IPv6 établi.

Cette valeur sera la cible de l'interrogation snmpget.

```
snmpwalk -v3 -a SHA -A <auth> -l authPriv -u <user> -x AES -X
<priv> <IP_routeur> cbgpPeer2State.ipv6 -On


.1.3.6.1.4.1.9.9.187.1.2.5.1.3.2.16.32.1.6.96.121.4.0.134.0.1.0.0
.0.0.0.1 = INTEGER: established(6)
```

Un peering BGP peut vérifier les états suivants :

1 -> Idle

2 -> Connect

3 -> Active

4 -> OpenSent

5 -> OpenConfirm

6 -> Established

Chaque peering doit être établi donc renvoyer la valeur numérique 6 lors de l'interrogation snmp.

Annexe 3 – Sécurité niveau 3

Exemple – équipement Cisco - filtrage sanitaire entrant [18] - bordure RENATER

```
ipv6 access-list de-INTERNET-v6
 remark anti-spoofing
 deny ipv6 2001:660:<network_id_site_Inria>::/48 any log
 remark Deny loopback address
 deny ipv6 host ::1 any log
 remark Deny IPv4-compatible addresses
 deny ipv6 ::/96 any log
 remark Deny IPv4-mapped addresses and obsoletes
```

```
 deny ipv6 ::FFFF:0.0.0.0/96 any log
 remark Deny auto tunneled packets w/compatible addresses (RFC
4291)
 remark Deny other compatible addresses
 deny ipv6 ::224.0.0.0/100 any log
 deny ipv6 ::127.0.0.0/104 any log
 deny ipv6 ::/104 any log
 deny ipv6 ::255.0.0.0/104 any log
 remark Deny false 6to4 packets
 deny ipv6 2002:E000::/20 any log
 deny ipv6 2002:7F00::/24 any log
 deny ipv6 2002::/24 any log
 deny ipv6 2002:FF00::/24 any log
 deny ipv6 2002:A00::/24 any log
 deny ipv6 2002:AC10::/28 any log
 deny ipv6 2002:C0A8::/32 any log
 remark Deny Site-Local deprecated
 deny ipv6 FEC0::/10 any log
 remark Deny Unique-Local packets
 deny ipv6 FC00::/7 any log
 remark Deny multicast packets (source seulement)
 deny ipv6 FF00::/8 any log
 remark Deny Documentation Address
 deny ipv6 2001:DB8::/32 any log
 remark Deny 6Bone addresses deprecated
 deny ipv6 3FFE::/16 any log
 remark Deny RH0 packets
 deny ipv6 any any routing-type 0 log
 remark ICMPv6 RFC4890
 permit icmp any any echo-reply
 permit icmp any any echo-request
 permit icmp any any destination-unreachable
 permit icmp any any port-unreachable
 permit icmp any any packet-too-big
 permit icmp any any time-exceeded
 permit icmp any any parameter-problem
 permit icmp FE80::/10 any nd-na
 permit icmp FE80::/10 any nd-ns
 remark ICMPv6 multicast
 permit icmp FE80::/10 FF02::/16
 deny icmp any any log
 remark trafic global depuis peer Renater
 permit tcp host <pair_Renater> host <pair_Inria> eq bgp
 remark protection adresses routeurs
 deny ipv6 any host <routeur_IPv6_globale1> log
 deny ipv6 any host <routeur_IPv6_globale2> log
 remark trafic vers plages IPv6 site
 permit ipv6 any 2001:660:<network_id_site_Inria>::/48
 remark fermeture finale
 deny ipv6 any any log
```

Exemple – équipement Cisco - désactivation routage par la source

```
no ipv6 source-route
```

Exemple – équipement Juniper SRX – protections IPS IPv6

```
set security screen ids-option <nom_protection> icmp icmpv6-
malformed
set security screen ids-option <nom_protection> ip ipv6-
extension-header routing-header
set security screen ids-option <nom_protection> ip ipv6-
malformed-header
```

Exemple – équipement Junipe SRX – logs d'activation protection IPS balayage IP

```
May 28 16:55:18 fw1 RT_IDS: RT_SCREEN_ICMP_LS: [lsys: lsys1]
Address sweep! source: 2a03:b0c0:2:d0:0:0:b42:8001, destination:
2001:660:xxxx:xxx:xxxx:0:0:65, zone name: untrust, interface
name: reth4.666, action: drop

May 28 16:55:18 fw1 RT_IDS: RT_SCREEN_ICMP_LS: [lsys: lsys1]
Address sweep! source: 2a03:b0c0:2:d0:0:0:b42:8001, destination:
2001:660:xxxx:xxx:xxxx:0:0:65, zone name: untrust, interface
name: reth4.666, action: drop
```

Annexe 4 : configuration IPv6 système d'exploitation

GNU/Linux CentOS 6 et 7

Sur un système CentOS déjà en production, pour activer IPv6, il faut tout d'abord modifier les fichiers d'interface **/etc/sysconfig/network-scripts/ifcfg-interface** :

```
IPV6INIT=yes
IPV6ADDR=2001:660:<network_id_site_Inria>:333::196/64
IPV6_DEFAULTGW=2001:660:<network_id_site_Inria>:333::30
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
```

Il est possible que **IPV6INIT** existe déjà et soit configuré à **no**.

Il convient de vérifier aussi dans /etc/sysctl* :

```
## Serveur avec ipv6 désactivé. Mettre une IP sur l'interface ne
suffira pas.
grep -r ipv6 /etc/sysctl.*
/etc/sysctl.conf:net.ipv6.conf.all.disable_ipv6=1
```

Pour valider le bon fonctionnement, utiliser :

```
ip -6 addr
ip -6 route
ping6 whatever
# Pour les services :
ss -ntlp
```

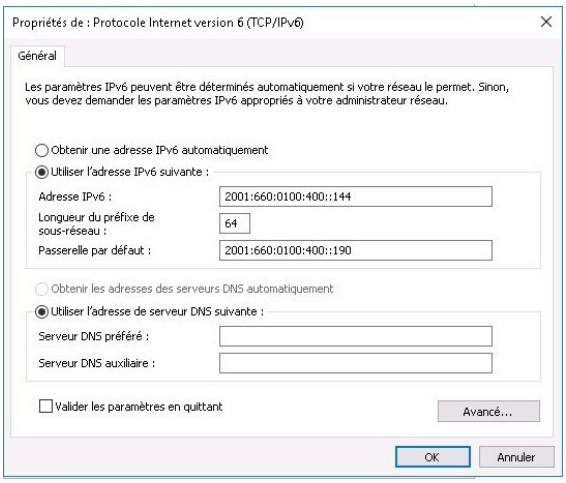
Un bug assez original que nous avons rencontré concerne d'ailleurs la directive *net.ipv6.conf.all.disable_ipv6*. Sur CentOS 6, à chaque installation de kernel est exécuté *dracut*, pour générer un initramfs. Dracut tient compte de l'état actuel du système (IPv6 activé ou désactivé dans sysctl) pour générer un initramfs supportant, ou pas, IPv6. Si l'on vient d'une configuration où sysctl désactivait globalement IPv6, il est possible que l'initramfs en porte encore la trace, et empêche IPv6 de fonctionner même si sysctl prétend que IPv6 est actif. Si le problème se présente, il faut relancer :

```
dracut -f
```

Avec IPv6 activé dans sysctl pour corriger le problème.

Windows Server 2016

Pour activer IPv6, configurer l'*adresse IP,* la *longueur du préfixe de sous-réseau* et la *passerelle par défaut* sur l'interface :



Si malgré ces paramètres, l'adresse IPv6 n'est pas visible sur l'interface, il se peut qu'une GPO (active ou passée) l'ai désactivé dans le registre. Il est possible de faire sauter ce verrou en supprimant la clé *DisabledComponents* située dans le registre à l'emplacement :

```
\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TCPIP6\
Parameters
```

Si vous souhaitez « pinger » la machine pour valider le bon fonctionnement de l'adresse IP, il est possible que le pare-feu Windows bloque le trafic ICMPv6. La règle à activer est *Partage de fichiers et d'imprimantes (Demande d'écho - ICMPv6 entrant).*

Annexe 5 : configuration IPv6 services

Apache (httpd) – GNU/Linux

Versions testées : httpd-2.2.15-69.el6 (CentOS 6.10) et httpd-2.4.6-80.el7 (CentOS 7.5)

Le démon httpd, de base, écoute sur toutes les interfaces. Il est possible de le vérifier avec la directive Listen dans **/etc/httpd/conf/httpd.conf** :

```
#Exemple commenté dans la configuration livrée de base
#Listen 12.34.56.78:80
# sans spécifier d'IP, « Listen 80 » fera écouter httpd sur
toutes les interfaces.
Listen 80
```

Il faut aussi s'assurer que les directives *VirtualHost* ne sont pas limitées à une adresse IP. Par exemple :

```
<VirtualHost 192.168.1.72:443>
```

doit être remplacé par :

```
<VirtualHost *:443>
```

Enfin, il faut penser à mettre à jour les ACL Allow/Deny dans les configurations : on ajout pour chaque hôte ou réseau IPv4, l'IPv6 correspondante si elle existe :

```
  allow from 192.168.10.0/24
  allow from 2001:660:<network_id_site_Inria>:333::/64
```

En complément, nous utilisons le module externe *mod_evasive* pour protéger nos serveurs Apache. De façon similaire aux ACL vues précédemment, il faut penser à ajouter le préfixe IPv6 du site Inria à la liste des réseaux autorisés. Cela se fait dans le fichier */etc/httpd/conf.d/mod_evasive.conf* :

```
    ...
    DOSWhitelist   192.168.1.*
    DOSWhitelist   192.168.2.*
    DOSWhitelist   2001:660:<network_id_site_Inria>:*
```

Puis redémarrer le service.

Plus d'informations :

- https://httpd.apache.org/docs/2.2/bind.html
- https://httpd.apache.org/docs/2.4/bind.html
- https://github.com/jzdziarski/mod_evasive

BIND (named) – GNU/Linux

Version testée : bind-chroot-9.9.4-61.el7_5.1 (CentOS 7.5)

Dans le fichier de configuration */etc/named.conf*, renseigner :

```
# insérer dans la section « options »
options {
   # autres options
```

```
    listen-on-v6 { any; };
}
```

Puis redémarrer le service.

Plus d'informations : https://deepthought.isc.org/article/AA-00821/0/Is-it-possible-to-configure-BIND-to-use-both-IPv6-and-IPv4-on-the-same-server.html

Codec Visio Cisco – Cisco

Matériel testé :

- Cisco TelePresence SX20 TC7.3.0.8cb420c
- Cisco TelePresence SX20 TC7.3.6.ea51021
- Cisco Codec C40 TC7.2.1.cb31c3d
- Cisco Codec C20 TC7.2.1.cb31c3d

Pour tous ces codecs visio, la procédure est la même : se rendre dans **Configuration -> System Configuration -> Network** et s'assure en haut que **IPStack** est configuré à **Dual**. Plus bas dans les options, on configure dans la section IPv6 les options : Address, Assignement = Static, DHCPOptions = Off et Gateway.

Puis sauvegarder les changements. L'adresse IPv6 est prise en compte immédiatement.

Plus d'informations :

- https://www.cisco.com/c/dam/en/us/td/docs/telepresence/endpoint/ce91/sx20-administrator-guide-ce91.pdf
- https://www.cisco.com/c/dam/en/us/td/docs/telepresence/endpoint/profile-series/tc6/administration_guide/profile-c60-c40-and-codec-c60-c40-administrator-guide-tc62.pdf

CrashplanPro (CPServer) – GNU/Linux

Version testée : 6.5.2 – 1508734800652 (sur CentOS 7.5)

Pour activer le support IPv6 de Crashplan, suivre la procédure officielle détaillée[33].

Ce que nous avons fait, en suivant la procédure ci-dessus :

```
Double-click the logo in the upper-left corner of the
administration console to open the administration console CLI.
Run the following commands:
1 address.bind peer ::
2 address.bind superpeer ::
3 address.bind space ::
Restart the Code42 server.
```

Puis dans Administration → Settings → Server, nous avons configuré comme suit :

33. https://support.code42.com/Administrator/6/Configuring/How_to_use_both_IPv4_and_IPv6_in_your_Code42_environment

```
Website protocol, host and port: http://monservice.domaine:4280
Primary network address: <adresse_IPv6>, TLS port: 4287
Secondary network address: <adresse_IPv4>, TLS port: 4287
```

Puir redémarrer le daemon depuis l'interface. La documentation officielle est assez complète, n'hésitez pas à la consulter

Plus d'informations : https://support.code42.com/Administrator/6/Configuring/How_to_use_both_IPv4_and_IPv6_in_your_Code42_environment

CUPS (cupsd) – GNU/Linux

Version testée : cups-1.6.3-35.el7 (CentOS 7.5)

Avec la configuration par défaut, cupsd écoute sur toutes les interfaces, y compris IPv6. Si ce n'est pas le cas, vérifier les directives Listen dans **/etc/cups/cupsd.conf** :

```
Listen *:631
Listen /var/run/cups/cups.sock
```

Il faut également mettre à jour les ACL (allow/deny) dans ce même fichier */etc/cups/cupsd.conf* : pour chaque hôte ou réseau IPv4 renseigné, ajouter l'adresse IPv6 correspondante. Attention à la syntaxe IPv6, qui utilise des crochets :

```
  Allow from 192.168.10.0/24
  Allow from [2001:660:<network_id_site_Inria>:333::]/64
```

Enfin, il y a un *bug* connu dans l'interface d'administration, qui génère une URL de la forme https://[v1.2001.660...]:631/admin, incompatible avec les navigateurs. Il devrait être corrigé dans les versions les plus récentes de cupsd[34].

Plus d'informations : man 5 cupsd.conf

DATA OnTap – NetApp

Version testée : 8.2.5P1 7-Mode

Pour rendre un filer NetApp compatible avec IPv6, il y a différents aspects à couvrir : l'OS lui même, les protocoles NFS et CIFS, et les spécificités liées au vFiler.

OS

Pour que le filer NetApp lui même puisse utiliser IPv6, il faut d'abord paramétrer les options suivantes :

34. https://github.com/apple/cups/issues/3991

```
options ip.v6.enable on
options ip.v6.ra_enable off # IPv6 statique
options ip.v6.autoconf_enable off # idem
```

Ensuite, il faut attribuer une adresse IPv6 au NetApp, avec les commandes :

```
ifconfig vif-global-333 2001:660:<network_id_site_Inria>:333::157
# interface existante, IPv6 du filer.
# Le préfixe n'est pas précisé car c'est /64 par défaut.
route add inet6 default 2001:660:<network_id_site_Inria>:333::190
1 # routeur du réseau et sa distance 1.
```

Ces deux lignes sont à mettre dans le fichier */etc/rc* pour la persistance au reboot.

Il est à noter que même si le filer répond en IPv6, ses services de base (SNMP, SSH, etc.) ne fonctionneront que en IPv4.

vFiler

Lorsqu'il s'agit d'un vFiler, son IPv6 doit lui être associé en tant que ressource. Depuis le filer qui fait actuellement tourner le vfiler, exécuter :

```
# vfiler add vfilername [-f] [-i ipaddr  [-i  ipaddr]...]
vfiler add bor-nas1 -i 2001:660:<network_id_site_Inria>:333::132
```

Puis il faut activer l'IP elle même sur une interface existante :

```
ifconfig vif-global-810 alias
2001:660:<network_id_site_Inria>:300::132 # IPv6 du vfiler.
```

Vu que c'est une ressource associée au vFiler, elle sera basculée avec les volumes et autres IPv4 en cas de transfert du vFiler sur un autre hôte.

CIFS

Cette section est valide autant pour un filer qu'un vFiler.

Une simple option est nécessaire pour activer le support IPv6 de CIFS :

```
options cifs.ipv6.enable on
```

Cela suffit à rendre les partages CIFS accessibles en IPv6.

NFS

Cette section est valide autant pour un filer qu'un vFiler.

Il y a une option pour activer le support IPv6 de NFS :

```
options nfs.ipv6.enable on
```

Toutefois, les partages ne seront pas automatiquement accessibles : le daemon *portmap*, qui annonce les partages et ses restrictions d'accès, a besoin d'être relancé. Pour se faire il faut désactiver puis réactiver NFS. Attention : ceci va interrompre les connexions NFS en cours.

```
nfs off
nfs on
```

Le service NFS est désormais accessible en IPv6, mais il faut mettre à jour les exports pour que les clients habituellement autorisés en IPv4 bénéficient du même accès. Le fichier */etc/exports* est à vérifier :

Si des exports sont fait sur des IP ou plages d'IP, ajouter les IPv6 correspondantes :

```
# exemple pour une hôte
/vol/vol0       -sec=sys,root=192.168.1.136:
[2001:660:<network_id_site_Inria>:333::136]
# exemple pour un réseau
/vol/bo_fzxl     -sec=sys,root=192.168.0.128/28:
[2001:660:<network_id_site_Inria>:335::]/64
```

Enfin, en interactif depuis le filer, recharger les exports :

```
exportfs -rv
```

Les exports utilisant des noms d'hôtes n'ont pas à être modifiés, mais nous avons constaté que le filer NetApp utilise un cache pour ses résolutions DNS et il est possible que les accès soient refusés pendant quelques minutes avant de fonctionner.

DHCP (dhcpd) – GNU/Linux

Version testée : dhcp-4.2.5-68.el7 (CentOS 7.5)

Nous utilisons le service DHCPD uniquement en IPv4. La seule chose à signaler est que la fonctionnalité de *Failover* ne supporte pas d'adresses IPv6 : *« There isn't a failover protocol defined for DHCPv6 yet and so dhcpd doesn't attempt to implement failover for DHCPv6. »*

La fonctionnalité restera donc opérationnelle en IPv4 (les IP sont précisées en dur dans les fichiers de configuration).

Plus d'informations :

- https://kb.isc.org/docs/aa-01356
- http://isc-dhcp-users.2343191.n4.nabble.com/DHCP-Failover-with-IPv6-on-dhcpd-4-3-2-td424.html

Dovecot (dovecot) – GNU/Linux

Version testée : dovecot-2.2.10-8.el7 (CentOS 7.5)

De base dovecot écoute déjà en IPv4 et IPv6, si ce n'est pas le cas, régler l'option *listen* dans le fichier */etc/dovecot/dovecot.conf* :

```
# A comma separated list of IPs or hosts where to listen in for
connections.
# "*" listens in all IPv4 interfaces, "::" listens in all IPv6
interfaces.
# If you want to specify non-default ports or anything more
complex,
# edit conf.d/master.conf.
#listen = *, ::
listen = *, ::
```

Puis redémarrer le service.

Plus d'informations : https://wiki.dovecot.org/Services#Service_listeners

matlab flexlm (MLM, lm_TMW.ld) – GNU/Linux

Version testée : r2018a

Nous utilisons pour serveur de licence MATLAB leur solution basée sur FLEXlm. Aucune configuration n'a été nécessaire pour le rendre compatible IPv6, les ports d'écoute se sont bien « bindé » sur toutes les interfaces. On observe avec *tcpdump* que le client matlab (version r2018b) privilégie IPv4 même en configuration *double pile*. IPv6 n'est utilisé que si c'est la seule connectivité disponible, mais elle fonctionne bien.

memcached (memcached) – GNU/Linux

Version testée : memcached-1.4.4-5.el6 (CentOS 6.10)

memcached écoute de base sur toutes les interfaces. Si ce n'est pas le cas, vérifier si l'option *-l* du daemon est présente. En son absence, le comportement par défaut est d'écouter sur *INADDR_ANY* (toutes les interfaces). Sur CentOS, les options du daemon sont gérées dans */etc/sysconfig/memcached* .

Plus d'informations : man 1 memcached

Munin (munin-node) – GNU/Linux

Versions testées :

- munin-node-2.0.40-4.el6 (CentOS 6.10)
- munin-node-2.0.40-4.el7 (CentOS 7.5)

Il est nécessaire d'installer le package *perl-IO-Socket-INET6* pour que munin-node écoute en IPv6. En son absence, il ne fera pas d'erreur mais écoutera uniquement en IPv4.

Dans le fichier */etc/munin/munin-node.conf*, on édite comme suit (détails dans les commentaires) :

```
# Si on a des ACL spécifiques, on les complète avec leur
équivalent IPv6
allow ^127\.0\.0\.1$ # localhost IPv4
allow ^::1$ # localhost IPv6
allow 192.168.1.175 # supervision IPv4
allow 2001:660:<network_id_site_Inria>:33::175 # supervision IPv6
# La directive host ne prend qu'une seule option.
# Pour écouter sur les interfaces IPv4 et IPv6, host * est la
seule option.
host *
```

Puis redémarrer le service.

Plus d'informations : man 5 munin-node.conf

Nagios Remote Plugin Executor (nrpe) – GNU/Linux

Versions testées :

- nrpe-3.2.1-6.el6 (CentOS 6.10)
- nrpe-3.2.1-6.el7 (CentOS 7.5)

De base, sans configuration particulière, nrpe écoute bien en IPv4 et IPv6. Si ce n'est pas le cas vérifier la présence de l'option *server_address* dans le fichier */etc/nagios/nrpe.cfg* :

```
# SERVER ADDRESS
# Address that nrpe should bind to in case there are more than
one interface
# and you do not want nrpe to bind on all interfaces.
# NOTE: This option is ignored if NRPE is running under either
inetd or xinetd
#server_address=127.0.0.1
```

Plus d'informations : https://github.com/NagiosEnterprises/nrpe/blob/master/sample-config/nrpe.cfg.in

mmm_mysql (mmm_mond / mmm_agentd) – GNU/Linux

Version testée : mysql-mmm-2.2.1-3.el6 (CentOS 6.10)

mmm, « Multi-Master Replication Manager for MySQL » est une solution simple pour gérer un cluster de serveurs MySQL (tous master). Hélas, elle n'est plus activement développée depuis l'intégration de Galera Cluster au projet MariaDB (solution désormais recommandée). Même si l'application est maintenue, IPv6 n'y sera pas intégré.

Plus d'informations : https://bugs.launchpad.net/mysql-mmm/+bug/1736962

Nginx (nginx) – GNU/Linux

Version testée : nginx-1.12.2-2.el7 (CentOS 7.5)

Pour que nginx écoute en IPv6, il faut ajuster les directives listen dans */etc/nginx/nginx.conf* :

```
http {
  server {
    listen 80;
    listen [::]:80;
    # ...
  }
  server {
    listen       443 ssl http2 default_server;
    listen       [::]:443 ssl http2 default_server;
   # ...
  }
}
```

Puis redémarrer le service.

Plus d'informations : http://nginx.org/en/docs/http/ngx_http_core_module.html#listen

ntopng (ntopng) – GNU/Linux

Version testée : ntopng Community Edition v.3.4.180712

Interface web : La version testée ne permet pas de faire écouter l'interface web en IPv6, mais cela n'est pas problématique car nous utilisons ntopng derrière un reverse-proxy, pour palier à son contrôle d'accès limité. Le reverse-proxy *Apache* étant lui accessible en IPv6, *ntopng* l'est également. Plus récemment, des mesures ont été prises par les développeurs du projet pour permettre de spécifier l'adresse à utiliser pour l'interface web[35].

Capture : sur cet aspect ntopng gère nativement IPv6. Nous avons juste eu à ajouter notre préfixe IPv6 pour qu'il soit vu comme un réseau interne, dans */etc/ntopng/ntopng.conf* :

```
#       -m|--local-networks
-m "...,2001:660:<network_id_site_Inria>::/48"
```

35. https://github.com/simonemainardi/ntopng/commit/b6eb2bc36c0c1a51a58dcb7031f8a958f3136663#diff-31ebf7cc04d6eef2420ff06d32158cbbR182

Puis redémarrer le service.

Plus d'informations : man 8 ntopng

NUT – Network UPS Tools (upsd) – GNU/Linux

Version testée : nut-2.6.5-2.el6 (CentOS 6.10)

upsd supporte nativement IPv6, il faut juste ajouter une directive *LISTEN* à */etc/ups/upsd.conf* :

```
# plusieurs directives LISTEN sont possibles
LISTEN 0.0.0.0 3493 # toutes interfaces IPv4
LISTEN :: 3493 # toutes interfaces IPv6
```

Puis redémarrer le service.

Plus d'informations : man 5 upsd.conf

OSSEC serveur (ossec-remoted) – GNU/Linux

Version testée : ossec-hids-server-2.9.4-5177.el7 (CentOS 7.5)

Le serveur *ossec-remoted*, servant à échanger avec les clients ossec en UDP, écoute de base sur toutes les interfaces. Penser toutefois à mettre à jour les «whitelist » avec leur équivalents IPv6 dans */var/ossec/etc/ossec-server.conf* :

```
<global>
  <white_list>127.0.0.1</white_list>
  <white_list>::1</white_list>
  <white_list>^localhost.localdomain$</white_list>
  <white_list>192.168.0.128/28</white_list>
  <white_list>2001:660:<network_id_site>:333::/64</white_list>
</global>
```

Puis redémarrer le service.

Plus d'informations : https://ossec-docs.readthedocs.io/en/latest/syntax/head_ossec_config.global.html

Postfix (master) – GNU/Linux

Version testée : postfix-2.10.1-6.el7 (CentOS 7.5)

Dans le fichier */etc/postfix/main.cf*, doivent appraître :

```
inet_interfaces = all # toutes les interfaces
inet_protocols = all # IPv4 et IPv6
```

Puis redémarrer le service.

Plus d'informations : man 5 postconf

rsyslog (rsyslogd) – GNU/Linux

Version testée : rsyslog-8.24.0-16.el7_5.4 (CentOS 7.5)

De base, rsyslog écoute sur toutes les interfaces. Ceci est dû au fait que les directives spécifiant les adresses IP à « binder » sont absentes ou commentées. Dans le fichier principal de configuration, */etc/rsyslog.conf* , les sections suivantes entrent en jeu :

```
# provides TCP syslog reception
$ModLoad imtcp
#$InputTCPServerAddress 127.0.0.1 #si absent, écoute sur toutes
les interfaces.
$InputTCPServerRun 514

# Provides UDP syslog reception
$ModLoad imudp
#$UDPServerAddress 127.0.0.1 #si absent, écoute sur toutes les
interfaces.
$UDPServerRun 514
```

Toujours dans ce fichier, penser à mettre à jour les règles de filtrage des logs si elles sont basées sur des IP. Chaque réseau ou hôte IPv4 qui a un équivalent IPv6 doit être configuré. Par exemple :

```
# Hôte unique
:fromhost-ip, isequal,
"192.168.1.72" :ommysql:127.0.0.1,syslog_infra,rsyslogdb,motdepas
se;log
& stop
:fromhost-ip, isequal,
"2001:660:<network_id_site>:312::72" :ommysql:127.0.0.1,syslog_in
fra,rsyslogdb,motdepasse;log

# Réseau de VLAN
:fromhost-ip, startswith,
"192.168.10." :ommysql:127.0.0.1,syslog_heberg,motdepasse,YVr1D6q
vYeNZ;log
& stop
:fromhost-ip, startswith, "2001:660:<network_id_site_Inria>:333:"
:ommysql:127.0.0.1,syslog_heberg,motdepasse,YVr1D6qvYeNZ;log
& stop
```

Puis redémarrer le service.

Plus d'informations :

- https://www.rsyslog.com/doc/master/configuration/modules/imudp.html

– https://www.rsyslog.com/doc/v8-stable/configuration/modules/imtcp.html

Squid et SquidGuard (squid et squidGuard) – GNU/Linux

Versions testées : squid-3.5.20-12.el7 (CentOS 7)

Pour squid, la directive http_port dans le fichier de configuration principal, */etc/squid/squid.conf*, fait foi. Si aucune IP n'est précisée, il écoutera sur toutes les interfaces disponibles.

```
#http_port 127.0.0.1:3128
http_port 3128
```

Dans ce même fichier, penser à mettre à jour les ACL, c'est à dire ajouter les versions IPv6 de toute hôte ou réseau qui est renseigné. Exemple :

```
# réseau entier
acl localnet src 192.168.0.128/28
acl localnet src 2001:660:<network_id_site_Inria>:333::/64
# hote unique
acl localnet src 192.168.0.136/32
acl localnet src 2001:660:<network_id_site_Inria>:333::136/128
```

Même chose pour squidGuard si des hôtes ou réseaux IPv4 y sont présents. C'est dans le fichier /etc/squid/squidGuard.conf :

```
src test-clients {
        ip              192.168.0.128/28
        ip              2001:660:<network_id_site_Inria>:333::/64
}
```

Puis redémarrer le service.

Plus d'informations : https://wiki.squid-cache.org/Features/IPv6

SSH Daemon (sshd) – GNU/Linux

Versions testées :

– openssh-server-5.3p1-123.el6_9 (CentOS 6.10)

– openssh-server-7.4p1-16.el7 (CentOS 7.5)

Dans le fichier de configuration */etc/ssh/sshd_config*, il faut spécifier les options :

```
AddressFamily any # écoute en IPv4 et IPv6 (default)
ListenAddress 0.0.0.0 # sur toutes les interfaces IPv4
ListenAddress :: # et toutes les interfaces IPv6
```

Puis redémarrer le service.

Plus d'informations : man 5 sshd_config

vsftpd (vsftpd) – GNU/Linux

Versions testées :

- vsftpd-2.2.2-24.el6 (CentOS 6.10)
- sftpd-3.0.2-22.el7 (CentOS 7.5)

vsftpd supporte IPv4 et IPv6, mais pas dans un seul daemon. Il faut dupliquer le fichier de configuration et lancer deux instances du daemon. Toutefois, les distributions *RHEL6/RHEL7* ont prévu le nécessaire pour simplifier la procédure.

Il faut d'abord, copier le fichier */etc/vsftpd/vsftpd.conf* en */etc/vsftpd/vsftpd-ipv6.conf* . Puis modifier afin que les directives *listen* et *listen_ipv6* soient configurées ainsi :

/etc/vsftpd/vsftpd.conf	/etc/vsftpd/vsftpd-ipv6.conf
`listen=YES` `listen_ipv6=NO`	`listen=NO` `listen_ipv6=YES`

Ensuite la procédure diverge :

RHEL 6

Il n'y a rien de plus à faire, le script d'init va lancer une instance de *vsftpd* pour chaque fichier *.conf* dans */etc/vsftpd* .

RHEL 7

Un mécanisme a été prévu dans systemd pour lancer plusieurs instances du service. Exécuter :

```
# vsftpd-ipv6 pointe automatiquement sur /etc/vsftpd/vsftpd-
ipv6.conf
systemctl enable vsftpd@vsftpd-ipv6.service
systemctl enable vsftpd.service
systemctl enable vsftpd.target
```

Puis redémarrer le service.

Autre point de vigilance, si vous utilisez *TCP_WRAPPERS=yes* (dans le fichier de configuration), il faudra ajouter les IPv6 aux hôtes IPv4 déjà présents dans */etc/hosts.allow* et */etc/hosts.deny*. La syntaxe pour une IPv6 est [IPv6] et pour un réseau [IPv6]::/mask

Plus d'informations :

- https://access.redhat.com/solutions/1566213
- https://forums.fedoraforum.org/showthread.php?97374-Common-vsftp-problems-and-likely-solutions

Annexe 6 – Support adressage IPv6 RFC 7217

- Linux depuis noyau 4.1[36]

Exemple - activé par défaut sur Fedora 29 - valeurs kernel :

```
# désactivation des adresses temporaires
net.ipv6.conf.all.use_tempaddr = 0
net.ipv6.conf.default.use_tempaddr = 0
net.ipv6.conf.<interface>.use_tempaddr = 0
net.ipv6.conf.all.addr_gen_mode = 0
net.ipv6.conf.default.addr_gen_mode = 0
# RFC 7217
net.ipv6.conf.<interface>.addr_gen_mode = 1
```

NetworkManager[37] génère un fichier de configuration d'interface incluant la valeur suivante - /etc/sysconfig/network-scripts/ifcfg-<interface> :

```
IPV6_ADDR_GEN_MODE=stable-privacy
```

- Mac OSX depuis version 10.12

Valeurs kernel :

```
# activation des adresses temporaires
net.inet6.ip6.use_tempaddr : 0
net.inet6.ip6.prefer_tempaddr: 0
# RFC 7217
net.inet6.send.opmode=1
```

Le comportement par défaut reste à valider car il semble que les adresses temporaires soient utilisées en premier lieu.

36. https://linuxfr.org/news/sortie-du-noyau-linux-4-1#impl%C3%A9mentation-de-la-rfc7217-ipv6
37. La connexion doit intègrer la valeur « ipv6.addr-gen-mode: stable-privacy ». Voir commande « nmcli connection ».

– Windows depuis la version 10[38]

38. Pas de documentation Microsoft identifiée. Cité sur le site support informatique de l'université de Cambridge ; https://help.uis.cam.ac.uk/service/network-services/ip/ipv6-at-cambridge#intro-addr-secured

Bibliographie

[1] ARCEP, Suivi de l'épuisement des adresses IPv4, 29 septembre 2019; https://www.arcep.fr/la-regulation/grands-dossiers-internet-et-numerique/lipv6/suivi-epuisement-adresses-ipv4.html

[2] IETF, RFC 7021 Assessing the Impact of Carrier-Grade NAT on Network Applications, septembre 2013 ; https://tools.ietf.org/html/rfc7021

[3] IETF, RFC 6164 Using 127-Bit IPv6 Prefixes on Inter-Router Links, avril 2011 ; https://tools.ietf.org/html/rfc6164

[4] IETF, RFC 6177 IPv6 Address Assignment to End Sites, mars 2011 ; https://tools.ietf.org/html/rfc6177

[5] IETF, RFC 7421 Analysis of the 64-bit Boundary in IPv6 Addressing, janvier 2015 ; http://www.rfc-editor.org/rfc/rfc7421.txt

[6] IETF, RFC 4890 Recommendations for Filtering ICMPv6 Messages in Firewalls, mai 2007 ; https://tools.ietf.org/html/rfc4890

[7] IETF, RFC 6104 Rogue IPv6 Router Advertisement Problem Statement, février 2011 ; https://tools.ietf.org/html/rfc6104

[8] IETF, RFC 6105 IPv6 Router Advertisement Guard, février 2011 ; https://tools.ietf.org/html/rfc6105

[9] IETF, RFC 7113 Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard), février 2014 ; https://tools.ietf.org/html/rfc7113

[10] IETF, RFC 7610 DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers, août 2015 ; https://tools.ietf.org/html/rfc7610

[11] IETF, RFC 4861 Neighbor Discovery for IP version 6 (IPv6), septembre 2007 ; https://tools.ietf.org/html/rfc4861

[12] IETF, RFC 3954 Cisco Systems NetFlow Services Export Version 9, octobre 2004 ; https://tools.ietf.org/html/rfc3954

[13] IETF, RFC 7011 Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, septembre 2013 ; https://tools.ietf.org/html/rfc7011

[14] IETF, RFC 7012 Information Model for IP Flow Information Export (IPFIX), septembre 2013 ; https://tools.ietf.org/html/rfc7012

[15] IETF, RFC 7217 A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC), avril 2014 ; https://tools.ietf.org/html/rfc7217

[16]IETF, RFC 8106 IPv6 Router Advertisement Options for DNS Configuration, mars 2017 ; https://tools.ietf.org/html/rfc8106

[17] IETF, RFC 6555 Happy Eyeballs: Success with Dual-Stack Hosts, avril 2012 ; https://tools.ietf.org/html/rfc6555

[18] Scott Hogg, Eric Vyncke. IPv6 Security. Cisco Press, 2009