
Développement de l'outil Ndmon

du projet Madynes - LORIA

Rapport de stage 2A - ESIAL

septembre 2006

Stagiaire : Thibault C

École supérieure d'Informatique et Applications de Lorraine (ESIAL)

Université Henri Poincaré Nancy 1

Campus Scientifique - B.P. 239

54506 V - ` -N Cedex

Tél. +33 (0)3 83 68 26 00

Site Web : <http://www.esial.uhp-nancy.fr>

Laboratoire et équipe d'accueil : Laboratoire Lorrain de Recherche en Informatique et Applications (LORIA)

Équipe MADYNES

Campus scientifique - 615, rue de Jardin Botanique - B.P. 101

54600 V - ` -N

Site Web : <http://www.loria.fr>

Encadrant Universitaire et maître de stage : Monsieur Olivier F

Directeur de Recherche INRIA

Responsable Scientifique du projet MADYNES au LORIA

Courriel : olivier.festor@loria.fr

Avant-propos

Ce rapport présente le travail réalisé au cours du stage de deux mois effectué au sein de l'équipe Madynes. Ce stage de type technicien supérieur est demandé dans le cursus de l'ESIAL, il est nécessaire à l'obtention de la deuxième année. Il permet aux élèves-ingénieurs de mettre en pratique dans un cadre professionnel les connaissances, les méthodes et les techniques acquises durant les deux premières années du cursus.

Je tiens à remercier particulièrement Olivier Festor pour la confiance qu'il m'a accordée lors de cette mission ainsi que Frederic Beck pour ses précieux conseils et sa disponibilité. Je remercie également le reste de l'équipe pour m'avoir si bien accueilli.

Sommaire

Avant-propos	2
Introduction	4
I Présentation du LORIA	5
I.1 Présentation générale	5
I.2 Les thèmes de recherche	5
I.3 Équipe Madynes	7
II Outils et protocoles support	8
II.1 Présentation d'Arpwatch	8
II.2 Présentation d'IPv6	9
II.2.1 Généralités	9
II.2.2 Neighbor Discovery	10
II.3 Exemple d'attaque utilisant ND	10
III Présentation de Ndmon	12
III.1 Monitoring	12
III.2 Utilisation	13
III.3 Fonctionnement général	13
IV Développement de Ndmon	15
IV.1 Organisation	15
IV.2 Utilisation des bibliothèques	16
IV.2.1 libpcap	16
IV.2.2 libxml2	16
IV.3 Analyse des paquets	17
IV.4 Tests	18
IV.5 Difficultés	18
Conclusion	20
Glossaire	21
Bibliographie	22
A Tests réalisés	23
B Manuel de Ndmon	26
C Résumé	28

Introduction

Le stage réalisé s'inscrit parfaitement dans le cadre de ma formation puisqu'il avait pour objectif le développement d'une application dans le domaine des réseaux, plus précisément d'un outil de surveillance des paquets de configuration utilisés par le protocole IPv6¹ appelé Ndmon pour *Neighbor Discovery Monitor*.

Ndmon a pour principale fonction de collecter et d'analyser les différents paquets circulant sur le réseau afin de déceler une attaque éventuelle ou plus simplement la mauvaise configuration d'un poste. Un tel outil est donc utilisé par l'administrateur d'un réseau afin d'avoir une meilleure connaissance de son évolution et des problèmes qui peuvent survenir. Ceci est particulièrement vrai pour un réseau utilisant le protocole IPv6 car celui-ci permet une configuration automatique par l'échange de messages entre les différents noeuds*, alors qu'Ipv4 nécessitait une configuration manuelle. Si la mise en oeuvre semble facilitée, il apparaît qu'un poste mal configuré peut nuire à l'ensemble du réseau et des attaques peuvent être facilement réalisées. L'intérêt de Ndmon est donc de fournir un outil de monitoring limitant les effets d'une mauvaise configuration en avertissant l'administrateur en cas de problème détecté, comme le montre la figure 1.

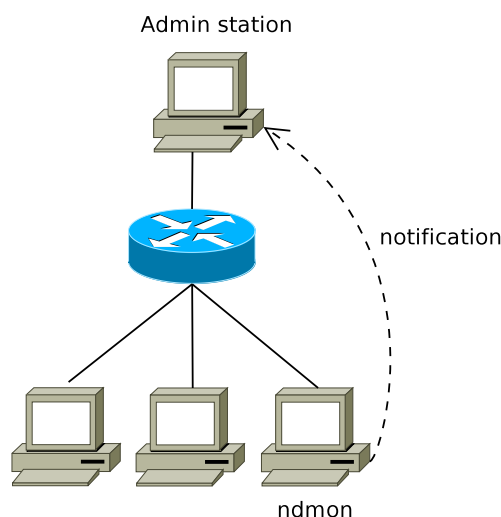


Figure 1 – Principe de l'outil Ndmon

Après avoir présenté le laboratoire et l'équipe de recherche qui m'ont accueilli, je décrirai précisément les outils et protocoles utilisés pour développer Ndmon puis le résultat finalement obtenu. Enfin, je parlerai plus précisément de la création du programme en mettant en évidence la démarche de travail réalisée et en entrant dans des considérations techniques.

¹Internet Protocol version 6

I Présentation du LORIA

I.1 Présentation générale

Le LORIA, Laboratoire Lorrain de Recherche en Informatique et ses Applications, est une Unité Mixte de Recherche - UMR 7503 - commune à plusieurs établissements

- CNRS, Centre National de Recherche Scientifique
- INPL, Institut National Polytechnique de Lorraine
- INRIA, Institut National de Recherche en Informatique et en Automatique
- UHP, Université Henri Poincaré, Nancy 1
- Nancy 2, Université Nancy 2

La création de cette unité a été officialisée le 19 décembre 1997 par la signature du contrat quadriennal avec le Ministère de l'Éducation Nationale, de la Recherche et de la Technologie et par une convention entre les cinq partenaires. Cette unité, renouvelée en 2001, succède ainsi au CRIN (Centre de Recherche en Informatique de Nancy), et associe les équipes communes entre celui-ci et l'Unité de Recherche INRIA Lorraine.

Le LORIA est un Laboratoire de plus de 450 personnes parmi lesquelles :

- 150 chercheurs et enseignants-chercheurs
- un tiers de doctorants et post doctorants
- des ingénieurs, techniciens et personnels administratifs organisés en équipes de recherche et services de soutien à la recherche

Chaque année le Loria compte aussi :

- une trentaine de chercheurs étrangers invités
- des coopérations internationales avec des pays des cinq continents
- une quarantaine de contrats industriels

Le laboratoire assume les missions suivantes :

- Recherche fondamentale et appliquée au niveau international dans le domaine des Sciences et Technologies de l'Information et de la Communication
- Formation par la recherche en partenariat avec les Universités lorraines
- Transfert technologique par le biais de partenariats industriels et par l'aide à la création d'entreprises

I.2 Les thèmes de recherche

La recherche au sein du Loria est organisée en équipes spécifiques à un domaine particulier. Chaque équipe regroupe autour d'un projet de recherche régulièrement évalué :

- des chercheurs
- des enseignants-chercheurs
- des doctorants
- des post-doctorants

A travers ses équipes de recherche, le LORIA possède des compétences reconnues en Sciences et Technologies de l'Information et de la Communication. Les équipes se structurent dans 6 thèmes de recherche et offrent des compétences scientifiques variées dans plusieurs

domaines d'application.

Les différents thèmes de recherche abordés sont :

- Calculs, simulation et visualisation à haute performance
- Qualité et sûreté des logiciels
- Systèmes parallèles, distribués et communicants
- Modèles et algorithmes pour les sciences du vivant
- Traitement de la langue naturelle et communication multimodale
- Représentation et gestion des connaissances

Les principaux domaines d'application sont :

- Réseaux, internet, Web
- Sécurité des systèmes informatiques
- Réalité virtuelle
- Robotique
- Bioinformatique
- Santé

Voici la liste exhaustive des différentes équipes composant le Loria :

- ADAGIo : Algorithmique Discrète et ses Applications à la Génomique et à l'Imagerie
- ALGORILLE : Algorithmes pour la Grille
- ALICE : Géométrie et Lumière
- CALLIGRAMME : Logique Linéaire, Réseaux de Démonstration et Grammaires Catégorielles
- CASSIS : Combinaison d'Approches pour la Sécurité des Systèmes Infinis
- CORTEX : Intelligence neuromimétique
- DEDALE : Développement de spécifications
- ECOO : Environnements pour la COOpération
- LANGUE ET DIALOGUE : Informatique linguistique pour le dialogue homme-machine multimodal
- MACSI : Modélisation, Analyse et Conduite des Systèmes Industriels
- MADYNES : Supervision des Réseaux et des Services Dynamiques
- MAGRITTE : Augmentation visuelle d'environnements complexes
- MAIA : MACHine Intelligente Autonome
- MERLIN : Méthodes pour l'Ergonomie des Logiciels Interactifs
- MODBIO : MODÈles informatiques en BIOlogie moléculaire
- MOSEL : Méthodes formelles et applications
- ORPAILLEUR : Extraction et représentation de connaissances
- PAROLE : Analyse, Perception et Reconnaissance automatique de la parole
- PROTHEO : Contraintes, déduction automatiques et preuves de propriétés de logiciels
- QGAR : Navigation dans les documents graphiques par l'analyse et la reconnaissance
- READ : Reconnaissance de l'écriture Et Analyse de Documents
- SITE : Modélisation et Développement de Systèmes d'Intelligence économique
- SPACES : Systèmes Polynomiaux, Arithmétiques, Calculs Efficaces et Sûrs
- TRIO : Temps Réel et InterOpérabilité

- TYPES : Logique, Théorie de la Démonstration et Programmation
- VEGAS : Algorithmes géométriques effectifs pour la visibilité et les surfaces

I.3 Équipe Madynes

L'équipe Madynes a pour thème de recherche la Supervision des Réseaux et des Services Dynamiques sous la direction d'Olivier Festor.

L'équipe de recherche MADYNES vise la conception, la validation et la mise en oeuvre de nouveaux paradigmes et architectures de supervision et de contrôle capables de maîtriser la dynamique croissante des services et résistantes au facteur d'échelle induit par l'Internet ubiquitaire.

Les axes thématiques de l'équipe sont :

- élaboration de méthodes d'auto-organisation des entités de gestion,
- conception, évaluation et mise en oeuvre d'architectures de supervision exploitant le modèle pair-à-pair, le routage applicatif, et de nouvelles approches pour la représentation de l'information de gestion,
- modélisation et benchmarking des infrastructures de supervision.
- sécurité : nouveaux protocoles de distribution de clefs et infrastructures pour le respect de l'anonymat et de la vie privée,
- la configuration et la provision de services,
- la mesure, l'analyse et l'instrumentation automatique des services.

L'Internet nouvelle génération est le domaine d'application principal des résultats des axes précédents. Son architecture ainsi que les services qui s'y déploient offrent toutes les caractéristiques de dynamique et de besoin de passage à l'échelle que Madynes aborde dans les autres axes du projet.

L'équipe travaille en relation avec d'autres partenaires scientifiques et industriels tels que :

- Partenaires académiques : LAAS-CNRS, LIP6, ENST, INSA de Lyon, LSR-IMAG, Twente University, Concordia University, UQAM, Macquarie University
- Partenaires industriels : 6Wind, Alcatel, France Telecom R&D, IBM, Thalès, Cisco Systems
- Ces coopérations sont renforcées par notre participation à de nombreux programmes nationaux (RNRT, RNTL, ACI, AS CNRS) et internationaux (projets 6Net, EUNICE) ainsi qu'à des groupes de recherche et de standardisation comme le NMRG à l'IRTF

Le développement de l'outil Ndmmon s'intègre dans les recherches et développements que l'équipe réalise au sujet d'IPv6. En particulier, cet outil de monitoring intéresse un partenaire industriel de l'équipe à savoir Cisco Systems.

II Outils et protocoles support

Cette partie est consacrée à la présentation détaillée des outils et protocoles étudiés pour concevoir Ndmn. Il y sera mis en évidence le contexte technologique de ce développement afin de mieux comprendre ensuite l'intérêt et le fonctionnement de l'outil.

II.1 Présentation d'Arpwatch

Si aucun logiciel de monitoring n'a été développé pour IPv6, il en existe en revanche pour la version actuelle du protocole IP. Ainsi le programme Arpwatch a servi dans un premier temps de cahier des charges à Ndmn qui reprend ses fonctionnalités en les adaptant au nouveau protocole IPv6. Je vais donc décrire dans cette partie l'outil de monitoring Arpwatch développé par *Lawrence Berkeley National Laboratory* entre 1990 et 2004.

Arpwatch analyse les paquets de type ARP² permettant de connaître l'adresse physique d'une carte réseau correspondant à une adresse IP et inversement via les paquets RARP³. Un poste transmettant des paquets ARP erronés entraîne donc un mauvais routage des paquets ce qui pose des problèmes de sécurité. Arpwatch relève ainsi l'activité de ces paquets et note les comportements suspects.

La notification des activités observées se fait de deux manières selon leur importance. Un premier niveau d'avertissement note l'information dans le journal de la machine */etc/var/log/syslog* comme le montre l'exemple ci-dessous. Si l'activité relevée par arpwatch est plus sérieuse, l'écriture dans le journal est complétée par l'envoi, via la commande mail de linux, d'un courrier électronique avertissant l'administrateur.

```

1 exemple de log: (date / machine / programme / situation / @IP / @MAC)
2 Jul 4 12:33:51 krusty arpwatch: bogon      152.81.11.15  0:13:72:51:fd:ef
3 Jul 4 12:33:51 krusty arpwatch: new station 152.81.114.47 0:8:74:92:b9:80

```

Pour détecter les dysfonctionnements (erreurs ou attaques), arpwatch construit sa base de données dans le fichier arp.dat qui a la forme suivante :

```

1 adresse MAC / adresse IP / date / DNS
2 0:13:5f:89:14:0      152.81.112.1      1152011624      loria112-gw
3 0:2:a5:63:15:68     152.81.114.180   1152011293      guest-180

```

Arpwatch est alors en mesure de comparer les informations contenues dans les paquets ARP analysés avec cette base de données pour détecter certains changements.

Les sources d'arpwatch étant libres, il a été très instructif d'observer son code source pour concevoir Ndmn. Ndmn reprend donc cette architecture en adaptant les analyses au protocole Ipv6 que nous allons maintenant décrire.

²Address Resolution Protocol

³Reverse Address Resolution Protocol

II.2 Présentation d'IPv6

II.2.1 Généralités

Le protocole IPv6 se charge comme son prédécesseur d'assurer l'élaboration et le transport des datagrammes IP (les paquets de données) tout en apportant de nouvelles fonctionnalités. Le besoin de faire évoluer le protocole IP s'est fait ressentir alors que le développement d'Internet laissait entrevoir rapidement une pénurie d'adresses ainsi qu'une augmentation des tables de routage. Les dernières spécifications du protocole sont définies dans le RFC⁴ 2460 de l'IETF⁵ [7].

Les principaux objectifs définis par l'IETF pour IPv6 sont les suivants :

- Pouvoir adresser un nombre beaucoup plus élevé d'ordinateurs, en se libérant de l'inefficacité de l'espace des adresses IP actuelles
- Réduire la taille des tables de routage
- Permettre une analyse plus rapide des datagrammes
- Améliorer la sécurité (authentification et confidentialité)
- Permettre différents types de services (notamment les services temps réel)
- Faciliter la diffusion multidestinataire
- Permettre une configuration automatique des réseaux
- Faciliter la mobilité des postes

Parmi toutes ces caractéristiques, certaines ont dû être étudiées pour développer Ndmn, particulièrement tout ce qui a trait à la configuration du réseau via le protocole Neighbor Discovery.

L'extension du nombre d'adresses est souvent présentée comme la motivation principale de l'évolution du protocole. Une adresse IPv6 est codée sur 128bits contre 32bits auparavant ce qui donne un nombre possible d'adresses de l'ordre de 3.4×10^{38} . Les 64 premiers bits appelés préfixe identifient le réseau, les 64 derniers bits sont appelés suffixe et identifient la machine sur le réseau. Le regroupement s'effectue par groupe de 16bits si bien qu'une adresse IPv6 a la forme suivante :

1234:FCBA:1024:AB45:6C5B:156:24:FE3

Plusieurs paquets de 16bits consécutifs mis à 0 peuvent être simplifiés par la notation " : : ". Comme pour IPv4 l'adressage IPv6 est hiérarchique et définit plusieurs classes d'adresses. La classe d'adresse commençant par FE80 est particulièrement utilisée par Ndmn car il s'agit d'une adresse de liaison locale, utilisée par les postes d'un même sous-réseau pour communiquer entre eux. Ainsi un poste ajouté à un réseau constitue son adresse de lien local à partir de son adresse MAC* en suivant un algorithme particulier.

⁴Request For Comment

⁵Internet Engineering Task Force

II.2.2 Neighbor Discovery

Le Neighbor Discovery Protocol[9] fait partie d'ICMPv6⁶ défini par l'IETF dans le RFC 2463[8]. ICMPv6 regroupe l'ensemble des messages de configuration nécessaires à l'organisation et au bon fonctionnement d'un réseau utilisant le protocole IPv6. ND⁷ remplace ainsi ARP, ICMPv4 Router Discovery, ICMPv4 Redirection, et apporte des fonctions supplémentaires.

Les différentes fonctionnalités d'autoconfiguration fournies par le protocole Neighbor Discovery sont les suivantes :

- Découverte des routeurs locaux
- Découverte des préfixes locaux
- Découverte des paramètres du réseau (ex : MTU*⁸)
- Configuration automatique des adresses (en l'absence de DHCP)
- Fonction de redirection (amélioration du routage)
- Détermination du tronçon suivant (routage des paquets)
- Résolution d'adresse (remplace ARP, construction d'une table de correspondance)
- Détection de l'inaccessibilité des voisins

Neighbor Discovery ainsi défini dans le RFC 2461 utilise une série de 5 messages ICMPv6 :

Router Solicitation (RS) : demande d'information sur les routeurs accessibles émise par un nœud désirant se configurer.

Router Advertisement (RA) : annonce périodique ou en réponse à un RS émise par le routeur concernant la configuration du réseau.

- adresse du routeur par défaut
- validité du routeur
- liste des préfixes utilisés sur le lien
- valeur possible du nombre de sauts
- valeur du MTU

Neighbor Solicitation (NS) : demande d'information sur un voisin.

- pour déterminer son adresse MAC(ex ARP)
- pour tester l'inaccessibilité
- pour tester si l'adresse désirée est libre (DAD⁹)

Neighbor Advertisement (NA) : annonce périodique d'un nœud sur le réseau ou en réponse à un NS, avertit le changement d'une adresse physique.

Redirect (Rd) : utilisé par un routeur pour informer d'une meilleure route.

II.3 Exemple d'attaque utilisant ND

Afin de mettre en évidence les problèmes que peuvent poser une mauvaise utilisation des messages de Neighbor Discovery, ce paragraphe explique concrètement un exemple de

⁶Internet Control Message Protocol version 6

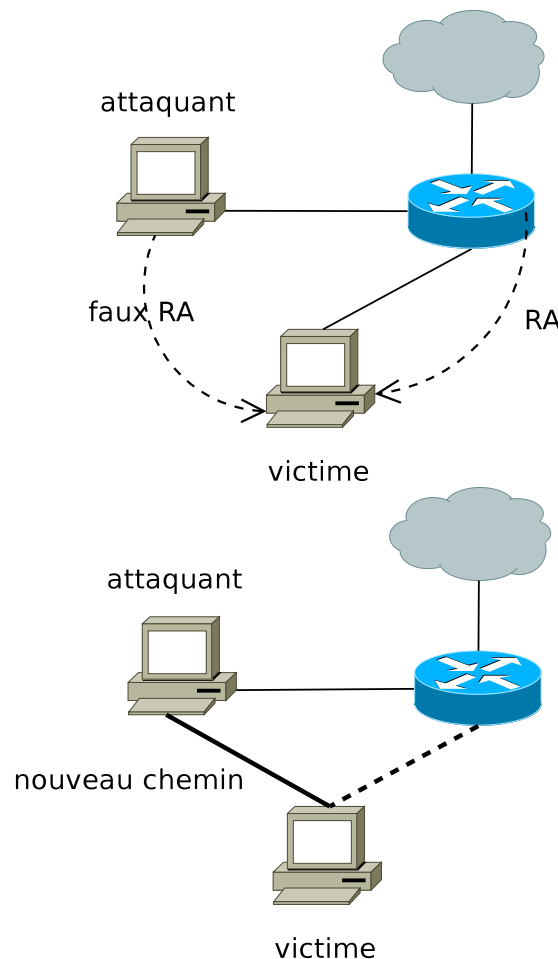
⁷Neighbor Discovery

⁸Maximum Transmission Unit

⁹Duplicate Address Detection

mauvaise configuration aboutissant à une attaque de type "Man in the middle".

On a vu que les messages de type RA fournissent des informations sur le routeur d'un réseau. Il suffit qu'un autre poste émette des RA bien formés avec un degré de priorité plus élevé (via les attributs Router Lifetime ou Reachable Time) pour que ceux-ci écrasent l'ancienne configuration valide. L'assaillant reçoit alors tout le trafic sortant du réseau ce qui pose des problèmes de sécurité évidents. Si le poste devenu routeur ne redirige pas effectivement les paquets reçus, il réalise une attaque de type DOS¹⁰ et les noeuds du réseau ne peuvent plus communiquer avec l'extérieur. Ceci est résumé par le schéma 2.



F . 2 – Exemple d'attaque utilisant NA

D'autres utilisations de ND peuvent permettre de réaliser très facilement des attaques. Ces différentes configurations, développées dans différents papiers[1] [2], ont été prises en compte, dans la mesure du possible, lors du développement de Ndmmon.

¹⁰Deny of service

III Présentation de Ndmon

Le contexte d'application de Ndmon étant maintenant défini, il est possible de présenter plus en détails l'outil en décrivant d'abord ses fonctions de monitoring, puis la manière dont il doit être utilisé. Enfin, nous verrons le fonctionnement général de Ndmon, en particulier les fichiers de configuration dont il se sert.

III.1 Monitoring

Cette partie présente maintenant les différentes activités pouvant être observées par Ndmon. Une partie des détections est implantée dans Arpwatch et d'autres spécifiques à Neighbor Discovery sont inédites.

Les messages suivant font l'objet d'une notification dans le journal du système :

new station : l'adresse MAC de la source du message ICMP n'a jamais été vue sur le réseau

new activity : le paquet provient d'un noeud n'ayant pas montré de signe d'activité depuis plus d'un mois

ethernet broadcast : l'adresse MAC de la source est une adresse spécifique de broadcast

ip broadcast : l'adresse IP de la source est une adresse spécifique de multicast

bogon : l'adresse IP de la source n'est pas locale au lien

ethernet mismatch : l'adresse MAC spécifiée en option du message ICMP ne concorde pas avec l'adresse MAC de la source (donnée par l'entête ethernet)

Les messages suivant posent davantage de problèmes pour la configuration du réseau et font en plus l'objet d'une alerte par courrier électronique.

wrong router mac : la source du RA possède une adresse MAC ne faisant pas partie des routeurs officiels (spécifiés dans le fichier de configuration)

wrong router ip : la source du RA possède une adresse IP ne faisant pas partie des routeurs officiels

wrong prefix : le préfixe annoncé par le RA n'est pas spécifié dans le fichier de configuration

wrong router redirect : la source du message de redirection ne fait pas partie des routers officiels

NA router flag : le NA comporte le flag spécifiant un routeur alors que la source n'est pas un routeur officiel

DAD DOS : un NA répond à un NS dans le cadre du mécanisme DAD pour l'empêcher d'obtenir une adresse IP

changed ethernet address : le noeud a changé d'adresse MAC en gardant la même adresse IP

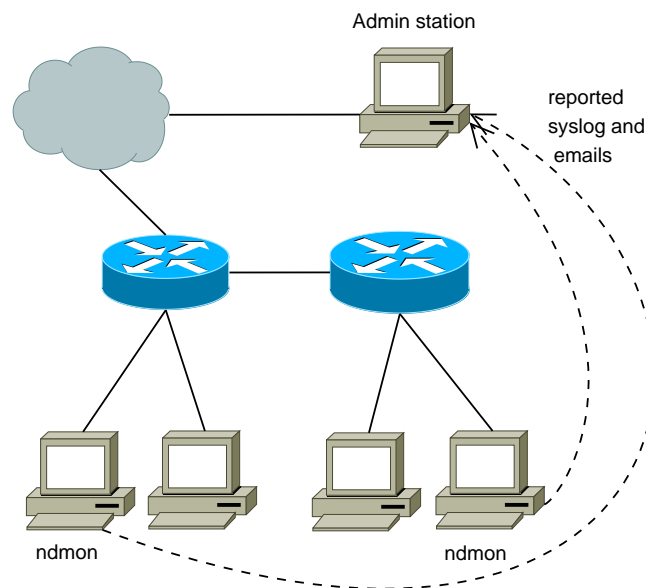
flip flop : un noeud alterne entre 2 adresses MAC

reused address : un noeud réutilise une ancienne adresse MAC

III.2 Utilisation

Ndmon ne nécessite pas de droits particuliers pour fonctionner et se contente d'analyser les paquets transmis en multicast sur le lien local. Il n'est donc pas nécessaire de configurer le réseau de manière à ce que Ndmon ait une vision privilégiée du trafic. En revanche, il doit être exécuté avec les droits d'administrateur car l'interface capturant les paquets a besoin de ces privilèges.

Concernant le déploiement du logiciel, Ndmon doit être installé sur un poste de chaque sous-réseau. La notification peut ensuite être centralisée en configurant d'une part la même adresse électronique pour chaque instance de Ndmon et d'autre part en configurant syslog pour écrire à distance sur la machine de l'administrateur. Ceci est résumé par le schéma 3.



F . 3 – Déploiement de Ndmon

Ndmon est prévu pour fonctionner en tant que démon sur un ordinateur, ce qui nécessite une attention toute particulière quant aux ressources qu'il demande et à sa stabilité. Pour cela, les outils Valgrind et KCacheGrind permettant d'analyser l'exécution d'un programme afin de connaître précisément sa consommation en ressources processeur et mémoire ont été d'une aide très précieuse.

III.3 Fonctionnement général

Ndmon reprend en grande partie le mode de fonctionnement d'Arpwatch à quelques différences près. Le protocole Neighbor Discovery implantant beaucoup plus de fonctionnalités que le protocole ARP, Ndmon tire davantage d'informations des paquets ICMP, étant ainsi en mesure de détecter des événements supplémentaires.

Pour pouvoir détecter certains problèmes, Ndmon a besoin de connaître la configuration valide du réseau. L'administrateur doit écrire ces données dans le fichier XML *config_ndmon.xml* dont voici un exemple :

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <!DOCTYPE config_ndmon SYSTEM "config_ndmon.dtd">
3 <config_ndmon>
4   <admin_mail>cholezth@loria.fr</admin_mail>
5   <authorised_routers>
6     <mac>0:30:b6:51:d4:1c</mac>
7     <ip>fe80:0:0:0:230:b6ff:fe51:d41c</ip>
8   </authorised_routers>
9   <authorised_prefixes>
10    <prefix>2001:660:4501:1:0:0:0:0</prefix>
11  </authorised_prefixes>
12 </config_ndmon>
```

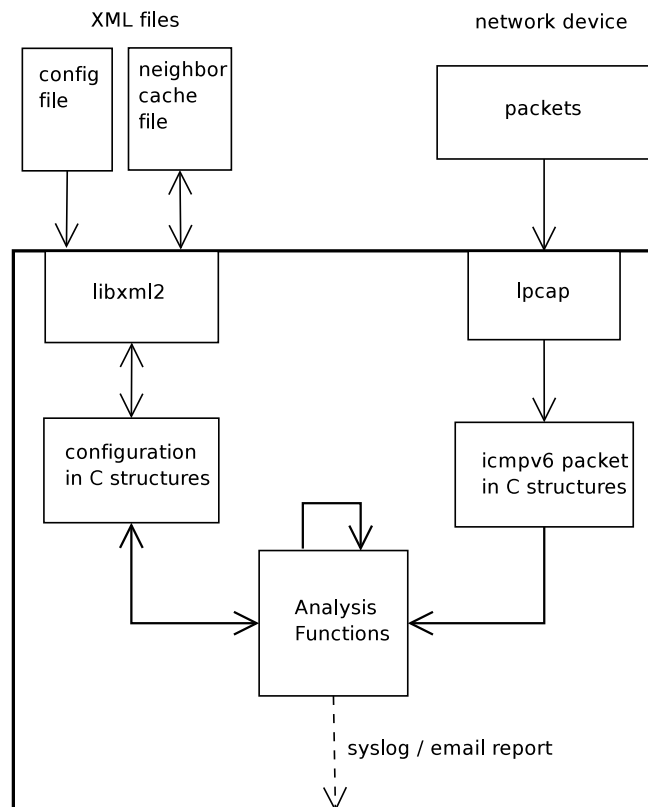
On note ainsi les adresses MAC et IP des routeurs officiels ou encore les préfixes pouvant être annoncés sur le réseau de manière à pouvoir détecter par la suite l'émission de paquets perturbant cette configuration. L'adresse électronique de l'administrateur est utilisée quant à elle pour pouvoir envoyer une notification.

Tout comme Arpwatch, Ndmon se constitue une base de données notant pour chaque noeud rencontré sur le réseau ses adresses MAC et IP ainsi que l'heure de réception du paquet. Cette base de données permet ensuite de détecter par comparaison l'évolution des noeuds présents sur le réseau local et de leur configuration. Ce fichier est par défaut nommé *neighbor_list.xml* dont voici un extrait :

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <!DOCTYPE neighbor_list
3 SYSTEM "neighbor_list.dtd">
4 <neighbor_list>
5   <neighbor>
6     <mac>0:2:a5:63:1a:66</mac>
7     <ip>fe80:0:0:0:202:a5ff:fe63:1a66</ip>
8     <time>1156781131</time>
9   </neighbor>
10  [...]
11 </neighbor_list>
```

Cette liste des voisins pouvant être difficilement écrite à la main par l'administrateur, Ndmon dispose d'une option permettant de constituer cette liste sans lever d'alertes. Une fois cette phase d'apprentissage effectuée, Ndmon dispose de sa base de données et peut efficacement surveiller le réseau.

Finalement, le fonctionnement général de l'application peut être résumé par le schéma 4.



F . 4 – Implantation de Ndmmon

IV Développement de Ndmmon

Cette partie aborde le développement de l’outil à proprement parler. Les différentes sections de cette partie retracent chronologiquement les étapes de l’élaboration du logiciel durant les 2 mois de stage.

IV.1 Organisation

La première étape a consisté en une phase de documentation afin d’avoir une meilleure idée sur l’outil à développer. Il a donc été nécessaire de se renseigner dans un premier temps sur Arpwatch, observer son utilisation et ses fonctionnalités car Ndmmon est assez semblable, puis se documenter sur les protocoles IPv6 et plus particulièrement Neighbor Discovery sur lesquels reposent l’outil.

La seconde étape a consisté à me familiariser avec la librairie C de capture de paquets qui est présentée dans la partie suivante. Les premières lignes de code ont consisté à utiliser cette librairie pour capturer les paquets ICMPv6 qui forment la matière de l’outil. Une fois les paquets capturés, il a fallu extraire les données brutes des paquets et les organiser dans différentes structures C de manière à les rendre facilement accessibles pour les analyses futures.

Ensuite la libxml a été utilisée afin d’importer dans le programme les données entrées par l’administrateur dans le fichier de configuration. Ceci fait, l’étape suivante a consisté à créer

les structures représentant la base de données des voisins sur le réseau puis faire l'interface entre cette structure et le fichier XML sauvegardant ces informations sur le disque.

Le programme ayant désormais tous les éléments pour effectuer son travail de monitoring, une grosse partie du développement a été d'utiliser ces informations et d'écrire les algorithmes permettant de détecter les activités du réseau.

Les dernières tâches effectuées ont été de valider toutes les fonctionnalités de l'application sur un réseau de test. La fin du travail réalisé fut la documentation de l'application en rédigeant le manuel (annexe B), un résumé (annexe C) et un rapport technique.

IV.2 Utilisation des bibliothèques

IV.2.1 libpcap

La bibliothèque nommée libpcap pour "packet capture" fournit différents services dans le domaine de la capture de paquets. Cette bibliothèque est également utilisée par Arpwatch. Elle a été réalisée initialement pour le programme tcpdump, un sniffer évolué en ligne de commande. Elle fournit différentes fonctions permettant de lire les paquets arrivant sur une interface et s'utilise de la manière suivante :

- choisir l'interface à écouter : en spécifiant son nom (ex : eth0), ou en laissant pcap choisir (fonction pcap_lookupdev).
- initialiser pcap avec cette interface : fonction pcap_open_live()
- filtrer le trafic pour garder les informations souhaitées : pour cela un filtre doit être créé puis compilé (pcap_compile(), pcap_setfilter()). La nature du filtre est exprimée grâce à des expressions dans une chaîne de caractères ex : icmp6 pour filtrer les paquets icmp6 dans le cas de Ndmon.
- capturer les paquets : les fonctions pcap_loop() ou pcap_dispatch() sont appelées à chaque fois qu'un paquet passant le filtre est reçu. Un pointeur indique alors l'emplacement des données brutes en mémoire. Le contenu du paquet peut ensuite être organisé dans des structures (en-têtes IP, TCP...) et peut être manipulé.

IV.2.2 libxml2

Ndmon utilise le langage XML¹¹ pour décrire la base des données des voisins et le fichier de configuration. Le langage XML se définit comme étant un format texte de balisage dont l'objectif est de faciliter l'échange de contenus entre différents systèmes d'informations. XML permet ainsi d'organiser les informations, de les rendre lisibles et réutilisables. La vérification de la validité des fichiers XML se fait en écrivant la structure des fichiers dans un fichier DTD correspondant. Si le fichier analysé n'est pas conforme à son schéma DTD, il ne pourra être utilisé. Voici les schémas des 2 fichiers XML utilisés par Ndmon :

config_ndmon.dtd :

```
1 <!ELEMENT config_ndmon (admin_mail, authorised_routers, authorised_prefixes)>
2 <!ELEMENT admin_mail (#PCDATA)>
3 <!ELEMENT authorised_routers (mac*, ip*)>
```

¹¹Extensible Markup Language

```
4 <!ELEMENT authorised_prefixes (prefix*)>
5 <!ELEMENT ip (#PCDATA)>
6 <!ELEMENT mac (#PCDATA)>
7 <!ELEMENT prefix (#PCDATA)>
```

neighbor_list.dtd :

```
1 <!ELEMENT config_neighbor_list (neighbor_list)>
2 <!ELEMENT neighbor_list (neighbor*)>
3 <!ELEMENT neighbor (mac, ip, time)>
4 <!ELEMENT ip (#PCDATA)>
5 <!ELEMENT mac (#PCDATA)>
6 <!ELEMENT time (#PCDATA)>
```

L'utilisation des fichiers XML dans le programme se fait grâce à l'interface DOM fournie par libxml2 qui permet d'exécuter une requête sur un fichier XML pour en extraire une partie. Il est alors possible de parcourir le résultat de la requête sous la forme d'un arbre (noeuds/feuille) suivant l'organisation du fichier.

IV.3 Analyse des paquets

Cette partie a pour but d'expliquer simplement la manière dont procède Ndmmon pour détecter les différentes activités sur le réseau.

Analyse générale : Une première série d'analyses est réalisée quelque soit le type de paquets de Neighbor Discovery reçu. La fonction *watch_eth_broadcast* regarde dans un premier temps l'entête ethernet du paquet afin de vérifier que la source n'utilise pas une adresse spécifique de broadcast. Les fonctions *watch_ip_broadcast* et *watch_bogon* réalisent des tests sur l'adresse source contenue dans l'entête IP. Enfin, *watch_eth_mismatch* vérifie que l'adresse source du paquet et celle annoncée en option du message sont bien identiques. Les comparaisons d'adresses sont de simples comparaisons d'octets en mémoire.

Annnonce d'un voisin : Une autre fonction *neighbor_announced* est commune aux messages RS, RA, NS, NA car ces quatre messages peuvent être utilisés, si l'on suit les recommandations du protocole IPv6, pour constituer le cache des voisins. Tout d'abord, les adresses IP et MAC de la source sont cherchées dans la base de données des voisins que se constitue Ndmmon. Si les 2 sont trouvées, la fonction *watch_last_time* regarde si la source a été active récemment, auquel cas pour cette adresse source, le champ de *time* de la base de données est mis à jour avec la date de réception du paquet. En revanche si aucune des deux adresses n'est trouvée, Ndmmon considère le paquet comme venant d'un nouveau voisin qu'il ajoute à la liste.

Si l'adresse MAC de la source est inconnue alors que l'adresse IP a déjà été recensée, c'est un cas de changement d'adresse ethernet. Les derniers changements d'adresse ethernet sont mémorisés par Ndmmon et sont utilisés pour détecter les changements de type flip flop et de réutilisation d'adresse.

Messages RS/RA : Lorsqu'un RA est reçu, les fonctions *watch_ra_mac* et *watch_ra_ip* vérifient que les adresses MAC et IP de la source ont été définies dans le fichier de configuration comme celles d'un routeur officiel. La fonction *watch_ra_prefix* vérifie ensuite la validité du préfixe annoncé en option du RA.

Messages NS/NA : Lorsqu'un NS est envoyé par un noeud n'ayant pas encore d'adresse IP, l'adresse demandée est mémorisée par Ndmon. Lors de la réception d'un NA, la fonction *watch_dad_dos* vérifie si le message répond au NS précédent. Si l'adresse demandée est annoncée par une nouvelle adresse MAC ou si la correspondance n'est pas confirmée par la base de données, Ndmon repère l'attaque. Il est aussi vérifié à la réception d'un NA par *watch_R_flag* que la source ne s'annonce pas comme étant un routeur par un flag spécial.

Messages Rd : La seule fonction appelée lors de la réception d'un message de redirection *watch_rd_src* vérifie que la source du message est bien un routeur officiel.

IV.4 Tests

Chaque fonctionnalité de Ndmon a fait l'objet d'une étape de validation. Le but de ces tests, dont certains sont présentés en annexe A, était de recréer les conditions de chaque activité observée par Ndmon afin de confirmer que le programme réagissait comme souhaité ou au contraire s'il nécessitait quelques modifications.

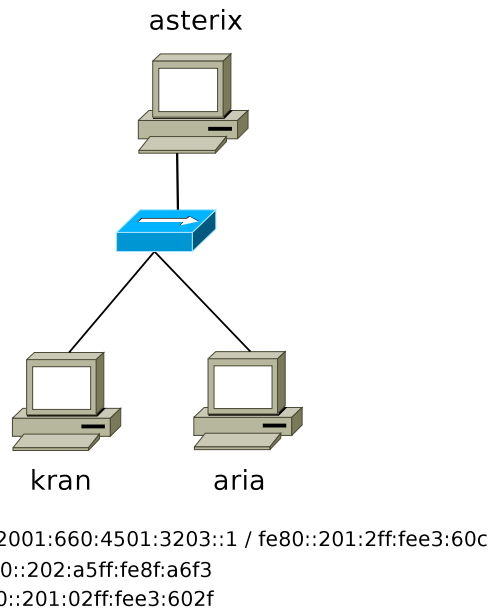
La majorité des tests a été réalisée grâce à l'outil *IPv6 hacking tool* développé spécifiquement par le groupe *The Hacker's Choice* pour manipuler facilement les paquets de configuration utilisés par IPv6.

Les tests ont pu être effectués sur un simple réseau de 3 machines comme le montre le schéma 5. Ndmon était exécuté sur l'une des machines alors qu'une autre modifiait sa configuration ou envoyait de faux paquets sur le lien.

IV.5 Difficultés

Le développement de Ndmon n'a pas posé beaucoup de difficultés même si certains points ont freiné le développement. Tout d'abord, concernant la *libpcap*, les modalités de configuration ont été déconcertantes. En effet, avant la compilation, la librairie doit être configurée avec l'option *-enable-ipv6* qui a été très difficile à trouver car cette librairie n'est pas documentée sur ce sujet. Alors qu'elle semblait bien fonctionner et que le manuel présentait la possibilité d'utiliser des filtres "ip6" ou "icmp6", ceux-ci refusaient d'opérer sans cette configuration préalable.

Un comportement étrange de l'API DOM de la *libxml2* a aussi été source de problèmes. En effet, cette librairie permet de retourner sous forme d'arbre correspondant au fichier XML le résultat d'une requête. Cependant, parcourir cet arbre DOM semblait donner des résultats incohérents. Il s'avère que l'arbre DOM contient des noeuds de type <texte> dont le contenu est vide, qui correspondent aux retours chariot servant à mettre en forme le fichier XML.



F . 5 – Réseau de test

L'information recherchée était donc souvent le fils de ce noeud vide. Pour un langage sensé accorder une grande importance à la sémantique, cette organisation a été déroutante.

Enfin, bien que peu utilisés, certains paquets IPv6 peuvent contenir des en-têtes optionnelles entre l'en-tête IP et l'en-tête ICMP qui est utilisée par Ndmom. Il a donc fallu prendre cette propriété en considération de manière à trouver les informations souhaitées en sautant les en-têtes optionnelles, ce qui était assez laborieux.

Conclusion

Au terme de ce stage, l'application Ndmmon qui m'a été confiée de réaliser est fonctionnelle. Les tests effectués permettent de penser que le programme est stable et fiable quant aux activités détectées, même si l'absence totale de bug est impossible à affirmer. Ndmmon peut d'ores et déjà être mis à disposition et testé sur d'autres réseaux extérieurs au Loria. Les remarques des utilisateurs permettront d'améliorer ensuite l'application. L'outil Ndmmon a été conçu de manière à évoluer facilement si son succès amène de nouveaux besoins. De nouveaux algorithmes pourront y être implantés afin de détecter d'autres problèmes de configuration ou des attaques plus élaborées.

Ce stage a été très enrichissant à de nombreux égards. Il m'a tout d'abord permis d'acquérir de nouvelles connaissances sur divers sujets comme le protocole IPv6, particulièrement sur les fonctionnalités de Neighbor Discovery, ou encore d'approfondir les problématiques liées à la sécurité des réseaux. D'un point de vue technique, il m'a donné l'occasion d'apprendre à utiliser le langage XML et de parfaire ma pratique du langage C par l'utilisation de nouvelles bibliothèques et d'outils d'optimisation. Enfin, il fut très intéressant de découvrir le fonctionnement d'un laboratoire de recherche, le type de travail qui y est réalisé et les différentes thématiques traitées. J'ai également eu la chance d'assister à des conférences au Loria et des réunions de l'équipe Madynes.

J'ai beaucoup apprécié cette mission car le travail demandé était très intéressant et correspond à l'une des branches de l'informatique que j'affectionne particulièrement, la sécurité informatique. De plus, le Loria propose un cadre et des conditions de travail très agréables qui ont facilité le bon déroulement de ce stage. Enfin, travailler au sein de l'équipe Madynes a été très plaisant compte tenu des qualités à la fois professionnelles et humaines de ses membres et de l'ambiance très sympathique qui y règne.

Glossaire

adresse MAC/ethernet : Adresse spécifique du périphérique reliant le noeud à la couche physique.

DAD : Duplicate Address Detection, ce mécanisme de Neighbor Discovery est utilisé quand un noeud se connecte au réseau, il demande préalablement si l'adresse ip qu'il désire est libre pour ne pas créer de conflit.

DOS : Deny Of Service, attaque visant à interrompre le bon fonctionnement d'une machine ou d'un réseau.

IETF : Internet Engineering Task Force est un groupe informel, international, ouvert à tout individu, qui participe à l'élaboration de standards pour Internet.

MTU : Maximum transmission unit, paramètre d'un réseau IP définissant la taille maximale (en octets) du paquet pouvant être transmis en une seule fois.

noeud : Périphérique implantant le protocole IP.

RFC : Request For Comment, littéralement demande de commentaires, sont une série de documents et normes concernant l'Internet rédigées sur l'initiative d'experts techniques.

routeur : Noeud transmettant les paquets IPv6 qui ne lui sont pas explicitement adressés, faisant le lien entre différents réseaux.

Références

- [1] Jari Arkko, Tuomas Aura, James Kempf, Vesa-Matti Mantyla, Pekka Nikander, Michael Roe, *Securing IPv6 Neighbor and Router Discovery*
- [2] P. Nikander, J. Kempf, E. Nordmark, *IPv6 Neighbor Discovery Trust Models and Threats*, May 2004
- [3] Tim Carstens, *Programming with pcap* <http://www.tcpdump.org/pcap.htm>
- [4] Packet Capture With libpcap and other Low Level Network Tricks <http://www.cet.nau.edu/~mc8/Socket/Tutorials/section1.html>
- [5] Libxml : The XML C parser and toolkit of Gnome <http://www.xmlsoft.org>
- [6] Yves Mettier, *Briques en C : libxml2 et petits fichiers XML* http://ymettier.free.fr/articles_lmag/lmag51_briques_en_C17/lmag51_briques_en_C17.html
- [7] IPv6 : <http://www.ietf.org/rfc/rfc2460.txt>
- [8] ICMPv6 : <http://www.ietf.org/rfc/rfc2463.txt>
- [9] Neighbor Discovery : <http://www.ietf.org/rfc/rfc2461.txt>
- [10] Peter Bieringer, HOWTO IPv6 Linux (fr) <http://livre.point6.net/index.php/Accueil>
- [11] Gisèle Cizault, *IPv6 théorie et pratique*, O'Reilly, 2005 <http://livre.point6.net/index.php/Accueil>
- [12] James F.Kurose, Keith W. Ross, *Computer Networking*, Addison Wesley, 2001

A Tests réalisés

Cette annexe présente une partie des tests réalisés durant le développement pour valider le fonctionnement du programme.

Test d'un faux RA : L'envoi d'un faux RA peut être généré très simplement en utilisant une commande de l'outil de *IPv6HT*¹². Il est aussi possible de réaliser cette attaque en utilisant l'outil RADVD, servant à émettre des messages RA pour configurer un réseau, avec un fichier de configuration modifié.

```
1 root@kran:~/thc-ipv6-0.6
2 %fake_router6 eth0 fe80:0:0:0:202:a5ff:fe8f:a6f3 2001:660:4501:3203:0:0:0/64 1500

1 length of this packet: 118
2 Recieved at: Mon Aug 21 14:51:39 2006
3 Source mac address: 0:2:a5:8f:a6:f3
4 Destination mac address: 33:33:0:0:0:1
5 Ethernet type hex:86dd dec: it's an IPv6 packet
6 Source ipv6 address: fe80:0:0:0:202:a5ff:fe8f:a6f3
7 Destination ipv6 address: ff02:0:0:0:0:0:1
8 Next header type: 58
9 IP type: 58, it's an ICMPv6 packet
10 ND type: 134
11 Option type: 5
12 Option length: 1
13 Option type: 3
14 Option length: 4
15 Option type: 1
16 Option length: 1
17 Router Advertisement:
18 Router Lifetime: 64000
19 Reachable Time: 64000
20 Retransmission timer: 7680
21 Neighbor cache updated.
22 Sending mail alert ...
23 Warning: wrong router mac 0:2:a5:8f:a6:f3 fe80:0:0:0:202:a5ff:fe8f:a6f3
24 Sending mail alert ...
25 Warning: wrong router ip 0:2:a5:8f:a6:f3 fe80:0:0:0:202:a5ff:fe8f:a6f3
26 Prefix: 2001:660:4501:3205
27 Sending mail alert ...
28 Warning: wrong prefix 2001:660:4501:3205 0:2:a5:8f:a6:f3 fe80:0:0:0:202:a5ff:fe8f:a6f3
```

On remarque dans la trace générée par Ndmmon suite à la réception de ce paquet que l'outil détecte la mauvaise provenance du message et le mauvais préfixe annoncé par rapport à la configuration normale (cf figure 5).

Test d'ethernet/ip broadcast : Le changement d'adresse MAC d'un poste peut être réalisée par les commandes linux suivantes :

```
1 ifconfig eth0 down
2 ifconfig eth0 hw ether xx:xx:xx:xx:xx
3 ifconfig eth0 up
```

On peut ainsi attribuer une adresse de broadcast à l'interface ethernet de l'ordinateur ce qui est observé par Ndmmon :

```
1 length of this packet: 78
2 Recieved at: Mon Aug 21 17:26:26 2006
3 Source mac address: ff:ff:ff:ff:ff:ff
4 Destination mac address: 33:33:ff:ff:ff:ff
```

¹²IPv6 Hacking Tools

```

5 Ethernet type hex:86dd dec: it's an IPv6 packet
6 Source ipv6 address: 0:0:0:0:0:0:0
7 Destination ipv6 address: ff02:0:0:0:0:1:ffff:ffff
8 Next header type: 58
9 IP type: 58, it's an ICMPv6 packet
10 ND type: 135
11 Warning: ethernet broadcast ff:ff:ff:ff:ff:ff 0:0:0:0:0:0:0
12 Neighbor Solicitation:
13 Target Address: 2001:660:4501:3203:fdff:ffff:feff:ffff

```

Pour annoncer qu'un noeud utilise une adresse IP particulière, l'outil de hacking permet de générer des NA. Ici un NA annonçant une adresse IP de multicast :

```

1 root@kran:~/thc-ipv6-0.6
2 % fake_advertise6 eth0 ff02:0:0:0:0:0:0:1
3 Starting advertisement of ff02:0:0:0:0:0:0:1

1 length of this packet: 86
2 Recieved at: Tue Aug 22 14:27:31 2006
3 Source mac address: 0:2:a5:8f:a6:f3
4 Destination mac address: 33:33:0:0:0:1
5 Ethernet type hex:86dd dec: it's an IPv6 packet
6 Source ipv6 address: ff02:0:0:0:0:0:0:1
7 Destination ipv6 address: ff02:0:0:0:0:0:0:1
8 Next header type: 58
9 IP type: 58, it's an ICMPv6 packet
10 ND type: 136
11 Option type: 2
12 Option length: 1
13 Warning: ip broadcast 0:2:a5:8f:a6:f3 ff02:0:0:0:0:0:0:1
14 Neighbor Advertisement:
15 Target Address: ff02:0:0:0:0:0:0:1
16 Neighbor cache updated.
17 NA flag router: 0

```

Test d'une attaque DAD DOS : Encore une fois, la suite IPv6HT permet de réaliser simplement cette attaque par la commande suivante :

```

1 %dos-new-ip6 eth0
2 Started ICMP6 DAD Denial-of-Service (Press Control-C to end) ...
3 Spoofed packet for existing ip6 as fe80:0000:0000:0000:0201:01ff:fee3:602f
4 Spoofed packet for existing ip6 as 2001:0660:4501:3203:0201:01ff:fee3:602f
5 Spoofed packet for existing ip6 as 2001:0660:4501:3203:0201:01ff:fee3:602f

1 length of this packet: 78
2 Recieved at: Tue Aug 22 17:43:20 2006
3 Source mac address: 0:1:1:e3:60:2f
4 Destination mac address: 33:33:ff:e3:60:2f
5 Ethernet type hex:86dd dec: it's an IPv6 packet
6 Source ipv6 address: 0:0:0:0:0:0:0
7 Destination ipv6 address: ff02:0:0:0:0:1:ffe3:602f
8 Next header type: 58
9 IP type: 58, it's an ICMPv6 packet
10 ND type: 135
11 Neighbor Solicitation:
12 Target Address: fe80:0:0:0:201:1ff:fee3:602f
13 -----
14
15 length of this packet: 86
16 Recieved at: Tue Aug 22 17:43:20 2006
17 Source mac address: 0:2:84:1f:8e:5
18 Destination mac address: 33:33:0:0:0:1
19 Ethernet type hex:86dd dec: it's an IPv6 packet
20 Source ipv6 address: fe80:0:0:0:201:1ff:fee3:602f

```

```

21 Destination ipv6 address: ff02:0:0:0:0:0:1
22 Next header type: 58
23 IP type: 58, it's an ICMPv6 packet
24 ND type: 136
25 Option type: 2
26 Option length: 1
27 Neighbor Advertisement:
28 Target Address: fe80:0:0:0:201:1ff:fee3:602f
29 Warning: new station 0:2:84:1f:8e:5 fe80:0:0:0:201:1ff:fee3:602f
30 nb change 0
31 Warning: changed ethernet address 0:2:12:f0:2e:6d to 0:2:84:1f:8e:5
32 fe80:0:0:0:201:1ff:fee3:602f
33 NA flag router: 0
34 Warning: dad dos 0:2:84:1f:8e:5 fe80:0:0:0:201:1ff:fee3:602f
35 Sending mail alert ...

```

L'attaque est donc bien détectée. La première trace montre un message NS demandant l'obtention d'une adresse, on le voit à l'adresse ip source qui est indéfinie. Le NA répondant affirme détenir l'adresse souhaitée mais comme l'adresse MAC source de ce message ne correspond pas à la base de donnée, l'alerte *changed_ethernet_address* est d'abord déclenchée puis celle concernant le DOS est annoncée à l'administrateur.

Test d'un flip flop : Pour finir ce paragraphe présente un petit programme simulant des changements d'adresse de type flip flop. Ce programme utilise la librairie fournie avec les utilitaires d'IPv6HT.

```

1 #include ...
2 #include <pcap.h>
3 #include "thc-ipv6-lib.c"
4
5 int main(int argc, char **argv){
6
7     unsigned char* srcip = "fe80:0:0:0:202:a5ff:fe8f:a6f3";
8     unsigned char* mac1 = "61:31:3a:62:32:3a";
9     unsigned char* mac2 = "61:61:3a:62:62:3a";
10
11     /*thc_neighboradv6(interface, src, dst, srcmac, dstmac, flags, target);*/
12
13     thc_neighboradv6("eth0", NULL, NULL, mac1, NULL, NULL, NULL);
14     sleep(2);
15     thc_neighboradv6("eth0", NULL, NULL, mac2, NULL, NULL, NULL);
16     sleep(2);
17     thc_neighboradv6("eth0", NULL, NULL, mac1, NULL, NULL, NULL);
18     sleep(2);
19
20 }

```

L'alternance entre les deux adresse MAC est bien repérée par Ndmmon :

```

1 length of this packet: 86
2 Recieved at: Fri Aug 25 10:54:48 2006
3 Source mac address: 61:31:3a:62:32:3a
4 Destination mac address: 33:33:0:0:0:1
5 Ethernet type hex:86dd dec: it's an IPv6 packet
6 Source ipv6 address: 2001:660:4501:3203:202:a5ff:fe8f:a6f3
7 Destination ipv6 address: ff02:0:0:0:0:0:1
8 Next header type: 58
9 IP type: 58, it's an ICMPv6 packet
10 ND type: 136
11 Option type: 2
12 Option length: 1
13 Neighbor Advertisement:
14 Target Address: 2001:660:4501:3203:202:a5ff:fe8f:a6f3
15 Warning: new station 61:31:3a:62:32:3a 2001:660:4501:3203:202:a5ff:fe8f:a6f3
16 Warning: changed ethernet address 61:61:3a:62:62:3a to 61:31:3a:62:32:3a

```

```

17 2001:660:4501:3203:202:a5ff:fe8f:a6f3
18 NA flag router: 0
19 -----
20
21 length of this packet: 86
22 Recieved at: Fri Aug 25 10:54:50 2006
23 Source mac address: 61:61:3a:62:62:3a
24 Destination mac address: 33:33:0:0:0:1
25 Ethernet type hex:86dd dec: it's an IPv6 packet
26 Source ipv6 address: 2001:660:4501:3203:202:a5ff:fe8f:a6f3
27 Destination ipv6 address: ff02:0:0:0:0:0:1
28 Next header type: 58
29 IP type: 58, it's an ICMPv6 packet
30 ND type: 136
31 Option type: 2
32 Option length: 1
33 Neighbor Advertisement:
34 Target Address: 2001:660:4501:3203:202:a5ff:fe8f:a6f3
35 Warning: new station 61:61:3a:62:62:3a 2001:660:4501:3203:202:a5ff:fe8f:a6f3
36 Warning: flip flop between 61:31:3a:62:32:3a and 61:61:3a:62:62:3a
37 2001:660:4501:3203:202:a5ff:fe8f:a6f3
38 Sending mail alert ...

```

B Manuel de Ndmon

NDMON(8)

NDMON(8)

NAME

ndmon - a monitoring software for ipv6 Neighbor Discovery

SYNOPSIS

```
ndmon [ -i interfacename ] [ -f configfile ] [ -F filter ]
      [ -n number ] [ -L ]
```

DESCRIPTION

Ndmon is a monitoring software for ipv6 Neighbor Discovery. It syslogs activity and reports by email malicious ND message. Ndmon uses libpcap to listen for icmp6 packets and libxml2 to use configuration and neighbor cache files.

The `-i` flag is used to change the default interface `eth0`.

The `-f` flag is used to change the path of the configuration file. The default is `config_ndmon.xml`.

The `-n` flag uses libpcap to specify a limited number of packet to capture.

The `-F` flag allows to change the default icmp6 filter.

The `-L` flag is used to disable syslog and mail reports. This is used to

do a learning phase and constitute the neighbor cache.

Note that an empty neighbor_cache.xml file must be created before the first time you run ndmon.

Ndmon must be run with root rights to work.

REPORT MESSAGES

Here is the list of the report messages generated by ndmon:

wrong router mac

The ethernet address of the RA message is not specified in the configuration file.

wrong router ip

The ip address of the RA message is not specified in the configuration file.

wrong prefix

The prefix announced in the RA message is not specified in the configuration file.

wrong router redirect

The RD message doesn't come from a router specified in the configuration file.

NA router flag

The NA specifies a router but isn't one according to the configuration file.

DAD DOS

The NA answer to NS to avoid it to get an ip address.

changed ethernet address

The host switched to a new ethernet address. flip flop
The ethernet address has changed from the most recently seen address to the second most recently seen address.

reused old ethernet address

The ethernet address has changed from the most recently seen address to the third (or greater) least recently seen address.

SYSLOG MESSAGES

Here are some of the syslog messages; note that messages that are reported are also syslogged.

new activity

This ethernet/ip6 address pair has been announced for last time two months or more.

new station

The ethernet address has not been seen before on the link.

ethernet broadcast

The mac ethernet address of the host is a broadcast address.

ip broadcast

The ip address of the host is a broadcast address.

bogon The source ip address is not local to the local subnet.

ethernet mismatch

The source mac ethernet address didn't match the address announced in option of the ND message.

FILES

`config_ndmon.xml` - contains settings which must be fill by the administrator

`neighbor_list.xml` - neighbor cache: all neighbors known to be on the link

SEE ALSO

`arpwatch(8)` `ipv6(7)`, `pcap(3)`, `libxml(3)`.

AUTHOR

Thibault Cholez for MADYNES, Loria, Fr.

BUGS

Please send bug reports to thibault.cholez@esial.uhp-nancy.fr

20 August 2006

NDMON(8)

C Résumé

Ndmon introduction :

Ndmon, Neighbor Discovery Monitor, is a tool working with icmpv6 packets. Ndmon watches on the local network if nodes using neighbor discovery messages behave properly. When it detects a suspicious ND message, it notifies the administrator by writing in the syslog and in some cases by sending an email report. Ndmon is very similar to arpwatch concerning reported activities and erroneous configurations, but it also provides new kinds of ND discovery specific features.

Ndmon can detect about 14 different kinds of activities listed below, further description of each report is available in the man file.

reported activities :

wrong router mac, wrong router ip, wrong prefix, wrong router redirect, NA router flag : ndmon is carefull about nodes sending router advertisements - only nodes specified to be official routers in the configuration file can send one.

DAD DOS, flip flop, reused old ethernet address : other kinds of malicious behaviors.

syslogged activities :

new station, new activity, ethernet broadcast, ip broadcast, bogon, ethernet mismatch, changed ethernet address : activities which are unusual but less serious are syslogged.

Ndmon can also be launch with an option disabling reports. This learning phase allows to build the neighbor database during the first execution without raising unappropriate warnings.

How does Ndmon work ?

Ndmon is all written in C language. It uses libpcap to get and filter neighbor discovery packets ; after that Ndmon does different tests.

Ndmon works with two xml files using libxml2. The first file contains configuration settings like official routers settings or the email address of the admin. The second file (that behaves like a cache) contains the list of all neighbors seen by ndmon on the local network. This cache keeps the ip address, mac address, and the last time of activity for each node. This list is updated automaticaly during the execution and saved on disk.

Conclusion :

Ndmon is a first step to have a better control over networks using ipv6 protocol. As future works, it can be improved by adding new detection rules.