TLS-SEC **2019-2020**

Bureau d'Étude réseau

Internet Protocol version 6 - vulnérabilités

29 août 2019

Ce TP à pour objectif d'expérimenter les mécanismes liés aux attaques par dépassement de tampon qui ont été exposés en cours. Pour ce faire, de premiers exemples simples sont étudiés. Par la suite, il sera exposé comment utiliser les dépassements de tapons pour contrôler le flux d'exécution d'un programme. Enfin, il est exposé quels sont les problèmes et concepts auxquels l'attaquant est confronté afin de développer une charge utile malveillante.

Ce TP apporte aussi quelques notions autour des interfaces binaire applicatives et noyau, de rétro ingénierie et d'utilisation d'outils de débogage de logiciels.

Matériel de travail pratique sur machines virtuelles Téléchargez et décompressez la machine virtuelle dédiée au travail pratique. Configurez votré émulateur de terminal dans `start.sh` (`gnome-terminal` ou `sakura` supportés).

```
$ wget -c 'http://flash.enseeiht.fr/bemorgan/debian-9.5.0-ipv6.tar'
$ tar xvf debian-9.5.0-ipv6.tar
$ cd debian-9.5.0-ipv6
# ./start.sh
```

Six fenêtres QEMU correpondant au 5 machines et routeurs du réseau vont s'ouvrir et vous proposer une interfaction graphique avec la console virtuelle.

Matériel de travail pratique sur machines physiques Munissez-vous d'une rangée de 4 machines. Brassez les interfaces réseaux des machines sur les ports du switch de votre baie en respectant la configuration des `vlans` préparés par l'enseignant.

> ☞ Si vous travaillez sur machine physique, vous commencerez par configurer le réseau interne, puis vous la basculerez sur le réseau Internet et la DMZ lorsque le TP traitera des attaques à distance.

1 Infrastructure réseau

Nous considérons dans ce bureau d'étude un réseau volontairement simplifié de manière à pouvoir se concentrer sur des aspects très précis du protocole IPv6 (figure 1).

Le réseau du Fournisseur d'Accès à Internet (FAI) est connecté au pare-feu de la société via le sous réseau `2002::/64`. En ce qui concerne le réseau de l'entreprise, la DMZ est connectée à l'aide du réseau `2000::/64` et sera configurée de façon statique. Enfin, les hôte du réseau interne doit supporter une flexibilité de topologie sur le réseau `2001::/64`. Leurs adresses IP et routeurs accessibles seront donc configurées automatiquement (à l'exeption du pare-feu qui sera configuré sur l'adresse `::1`).

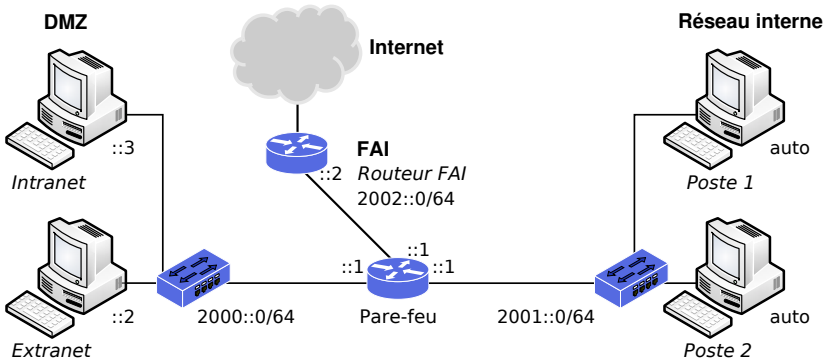

FIGURE 1 – Réseau considéré pour le TP

1.1 Mise en place de la topologie physique

Démarrez les machines virtuelles ou configurez physiquement le réseau comme indiqué dans le préambule du sujet. Il est possible d'identifier les interfaces physiques de vos machines à l'aide de la commande `ethtool`.

```
# ethtool --identify INTERFACE
```

Une fois les machines interconnectées, il est temps de configurer les interfaces réseau.

1.2 Configuration des hôtes statiques et du routeur

La commande UNIX `ip` est une boite à outil qui essaie de centraliser toutes les fonctionnalités de configuration du réseau. Elle remplace par exemple les commandes `ifconfig` par `ip {address | link}` ; `route` par `ip route` et `arp` par `ip neighbour`. Notez que la commande `ip` supporte les raccourcis d'arguments : `ip neighbour` peut être par exemple invoqué par `ip n`.

Activez les interfaces réseau du pare-feu, de la machine DME et du routeur FAI, à l'aide de la commande `ip`.

```
Listez les interfaces
# ip a
Activez le lien
```

```
# ip link set up dev INTERFACE
Observez la différence
# ip a
```

Constatez la configuration automatique de l'adresse *link local*. Utilisez la commande ping pour tester la connectivité sur les même domaines de diffusion.

```
$ ping LINK-LOCAL-ADDRESS
```

Que ce passe-t-il ? Pourquoi est-ce que cela ne fonctionne pas ? Consultez la table de routage du la machine pare-feu.

```
$ ip -6 route
```

Spécifiez l'interface de sortie à l'aide de la notation `%INTERFACE`.

```
$ ping LINK-LOCAL-ADDRESS%INTERFACE
```

Associez les adresses IPv6 aux machines et routeurs du réseau configurés statiquement.

```
# ip a a ADDRESS/MASK dev INTERFACE
```

Configurez les routes statiques pour les hôtes des réseaux FAI, DMZ et pour le pare-feu. N'utilisez pas de routes par défaut. N'oubliez pas l'accessibilité du réseau interne.

```
# ip route a NETWORK-ADDRESS/MASK via ADDRESS [dev INTERFACE]
```

Activez le relayage IPv6 (mode routeur) de la machine pare-feu. Testez la connectivité avec la commande ping.

```
firewall# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
ou
firewall$ echo 1 | sudo tee /proc/sys/net/ipv6/conf/all/forwarding
```

1.3 Configuration automatique du réseau interne

Dans cet exercice, nous allons configurer les adresses ip des machines internes et leurs routes par défaut à l'aide du protocole *Neighbour Discovery* (ND) et plus spécifiquement des messages ICMPv6 Router Advertisement (RA).

Les messages RA sont les seul à pouvoir pourvoir une route par défaut dans le cadre de la configuration automatique ou dynamique à l'aide de DHCPv6.

La configuration automatique des adresses IP est réalisée à l'aide de l'envoi de *router advertisements* par le(s) router(s) du réseau. Ces paquets contiennent un liste de préfixes IP dont certains sont annoncé comme a utiliser pour l'auto configuration.

Les préfixes sont annoncés comme accessibles par le routeur annonçant et sont donc éventuellement ajoutés à la table de routage.

Vérifiifiez que les RA pour la configuration automatique sont bien acceptés par la pile IP des machines internes. Activez la si nécessaire.

```
firewall$ cat /proc/sys/net/conf/ipv6/all/ra_accept
firewall$ echo 1 | sudo tee /proc/sys/net/conf/ipv6/all/ra_accept
```

La documentation des fichiers de configuration systèmes qui concernent spécifiquement la configuration automatique, mais plus généralement IPv6 est disponible à cette adresse :
`https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt`.

Sous les distributions GNU/linux Un *daemon* est responsable de l'envoi des RA. Il s'agit de `radvd` sur votre machine pare-feu Debian. Son fichier de configuration est situé au chemin `/etc/radvd.conf` et possède la forme suivante :

```
interface INTERFACE {
  AdvSendAdvert on;
  MinRtrAdvInterval 3;
  MaxRtrAdvInterval 10;
  prefix NETWORK/MASK {
    AdvOnLink {on|off};
    AdvAutonomous {on|off};
    AdvRouterAddr {on|off};
  };
};
```

La section prefix est le cœur de la semantique du RA et peut être répétée autant que nécessaire. Le drapeau `AdvOnLink` signifie que le préfixe est acessible sur le domaine de diffusion local. Le drapeau `AdvAutonomous` signifie que l'hôte doit générer une ip préfixe et la configurer sur l'interface de réception. Le drapeau `AdvRouterAddr` signifie que ce préfixe est routable.

Configurer `/etc/radvd.conf` pour le réseau interne et démarrez le service `radvd.service`.

```
firewall# systemctl start radvd.service
```

Activez les interfaces des machines internes 1 et 2 et constatez les modification de configuration des interfaces et de la table de routage.

```
$ ip a
$ ip -6 r
```

Quelles sont les différences notoires entre ces adresses automatiquement configurées et celles du pare-feu ?

Testez la connectivité des machines internes entre la DMZ et le routeur du FAI avec la commande `ping`.

Observez le réseau sur le routeur à l'aide de la commande `wireshark{|-gtk}`. Si vous être sur machine virtuelle, observez le bridge `br1`. Observez le traffic RA à l'aide des commandes `radvdump` et `ip monitor`.

```
# radvdump
$ ip monitor
```

Redémarrez le service radvd sur le pare-feu et observez les messages RA générés à l'aide de wireshark et leur effet sur les hôtes internes à l'aide de la commande `ip monitor`.

```
firewall# systemctl start radvd.service # Plusieurs fois
```

2 Attaquant interne

Considérons le modèle d'attaque décrit pas la figure (figure 2). Une machine interne à été totalement corrompue, c'est à dire qu'elle peut émettre à volonté du trafic réseau sur son interface. Le but de cet exercice est de démontrer pourquoi les concepteurs d'IPv6 ont considéré le même modèle d'attaquand que pour la version 4 du même protocole : c'est à dire que les hôte internes sont de confiance.

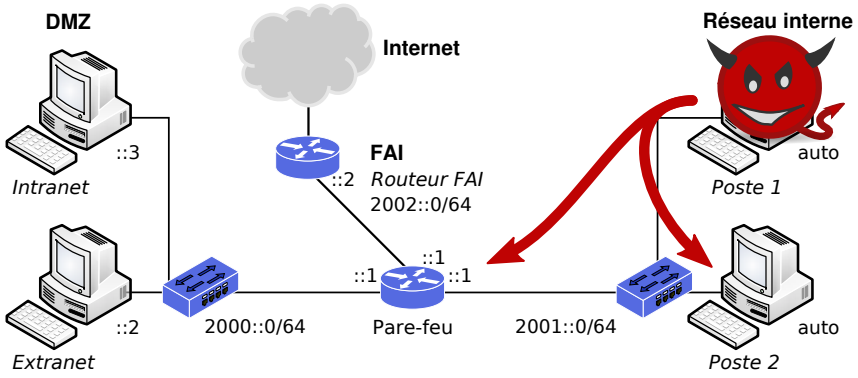

FIGURE 2 – Attaquant interne

2.1 Dénis de service par *Router Advertisement*

Comme expliqué dans la section précédente, l'autoconfiguration est basée sur des messages de *Router Advertisement* (RA) envoyés par le routeur. Nous avons aussi vu que ces paquets impliquent la **création d'entrées dans la table de routage** des hôtes auto configurés.

— Proposez une méthode d'attaque de la disponibilité du réseau du Poste 2 basée sur l'envoi de RA malveillants depuis le Poste 1. Sélectionnez les paramètres intéressant afin de mener à bien l'attaque. L'objectif de cette attaque est de supprimer la route par défaut du Poste 2, depuis le Poste 1.

— Quels sont les paramètres temporels de cette attaque ?

Une fois la stratégie d'attaque, sélectionnez les champs des messages RA susceptibles ce mener à bien l'attaque.

À l'aide de l'outil *Scapy* (commande `scapy`) et forts de votre dernière observation, générez depuis la machine corrompue interne 1, des RA malveillants pour perturber la disponibilité du trafic réseau du Poste 1.

Vous vérifierez le succès de votre attaque à l'aide de la commande `ip -6 r` pour afficher la table de routage IPv6.

Proposez une contremesure simple pour empêcher Poste 1 de se déguiser en routeur.

2.2 Dénis de service par *Duplicate Address Detection*

Lorsqu'un hôte reçoit le premier *Router Advertisement* concernant sa configuration automatique, il va dérouler le protocole *Duplicate Address Detection* (DAD), afin de déterminer s'il est l'unique usager de cette adresse IP sur le domaine de diffusion local concerné. En effet, plusieurs stratégies peuvent être déroulée afin de générer une adresse IP, mais aucune n'offre la certitude de ne pas avoir sélectionné la même adresse IP qu'un hôte du même domaine de diffusion. Ce constat est vrai même avec dérivation depuis l'adresse MAC (déguisement MAC possible).

Pour ce faire, après avoir généré l'adresse IP, l'hôte envoie une requête de *Neighbour Solicitation* (NS) sur une adresse *multicast* dérivée de l'adresse cible. Un hôte utilisant déjà cette adresse IP va répondre a cette requête avec un *Neighbou Advertisement* (NA). Si l'hôte qui se configure reçoit une réponse au NS, un conflit d'adresse est détecté, il doit sélectionner une autre adresse IP et rejouer le même protocole.

— Proposez une méthode pour attaquer la disponibilité du réseau de l'hôte qui se configure, en utilisant le protocole DAD. L'attaquant est toujours placé sur le Poste 1.

— Quelles sont les contraintes temporelles de cette attaque ?

De nombre outils de tests d'intrusion IPv6 existent actuellement, tels que `thc-ipv6` :
`https://github.com/vanhauser-thc/thc-ipv6`
`https://tools.kali.org/information-gathering/thc-ipv6`

- Lisez la documentation de l'outil `dos-new-ip`.
- Désactivez l'interface réseau du Poste 2 afin de libérer son adresse IP.
- Utilisez l'outil depuis la Poste 1, afin d'empêcher le Poste 2 d'accéder au réseau.
- Activez l'interface réseau du Poste 2.
- Constatez les fonctionnement à l'aide de la commande `ip a` et de `wireshark`

2.3 Empoisonnement du cache *Neighbours* par *Neighbour Advertisement*

IPv6 nécessite toujours une traduction des adresses IP vers adresses de liaison de données. Dans notre cas il s'agit bien sûr d'Ehternet et d'adresses MAC.

La traduction s'effetue à l'aide des messages ICMPv6 de requête de traduction et de réponse de traduction, respectivement mis en œuvre avec les paquets ICMPv6 NS et NA.

- Proposez une stratégie d'attaque de ce protocole de traduction de façon a affecter la vision du réseau du Poste 2 en attaquant depuis le Poste 1. L'objectif de l'attaquant est de voler l'adresse IP du routeur (`2000::1`). Quelles sont les contraintes temporelles de cette attaque ?
- À l'aide de l'outil scapy, générez un flot conséquent de paquet ICMPv6 bien choisit pour vous faire passer pour la machine Poste 2 du point de vue du pare-feu.

Vérifiez l'efficacité de votre attaque à l'aide de `wireshark` et de `ip neigh`. Commencez par observer l'état du cache *neighbours* sur le poste 2 victime. Ensuite générez le flot de *Neighbour Adversitement* adéquat depuis le poste 1 malveillant. Générez des *echo request* ICMPv6 depuis la machine victime et observez l'adresse MAC utilisée.

Est-ce que le Poste 2 reçoit les réponses *echo-reply* ? Pourquoi ?

Lors de l'empoisonnement du cache *neighbour*, vous indiquez à la victime de vous transmettre des trames Ethernet dont l'adresse IP de destination ne vous appartient pas.

Comment modifier la configuration du Poste malveillant 1 pour ques les paquets que vous avez reçu soient tout de même relayés au routeur de façon à ce que le Poste 1 reçoive une réponse.

L'attaquant sera ainsi placé en homme dans le milieu dans le sens de communication Poste 2 vers pare-feu.

— Observez les traces réseau générés après modification de votre configuration du Poste 1.

— En plus du relayage et de la réponse à la requête ICMPv6 *echo*, quels type de message ICMPv6 la machine Poste 1 génère en plus ? En quoi est-ce que ces messages vous dévoilent en tant qu'attaquant ?

— Enfin, quel autre champ du paquet modifié après le relayage par le Poste 1 trahis aussi votre présence en homme dans le milieu ? Auprès de qui ?

2.4 Homme dans le milieu par empoisonnement du cache *Neighbours* par *Neighour Advertisement*

Lors du précédent exercice nous avons réussi à nous placer partiellement en homme dans le milieu entre deux machines.

— Proposez une extension de votre stratégie d'attaque précédente de façon à vous placer en homme dans le milieu complet.

— Sans mettre en œuvre l'attaque avec `sacpy` proposez rapidement les ajouts à effectuer pour avoir une attaque réussie.

L'outil `fake_advertise6` de la suite `thc-ipv6` d'empoisonner facilement le cache *neighbour* dans le milieu dans un domaine de diffusion local.

— Documentez-vous sur l'outil `fake_advertise6`

— Exécutez l'attaque pour placer le Poste 1 en homme dans le milieu entre le Poste 2 et le routeur.

— Observez les paquets échangés à l'aide de `wireshark`.

3 Attaquant externe

Dans cette partie du TP, nous allons nous intéresser à des attaquand externes. C'est à dire en dehors du réseau administré par une même entité et plus spécifiquement placé à bonne distance des domaines de diffusion ciblés par l'attaquant.

3.1 Déguisement IP et *routing header* de type 0

La DMZ héberge deux serveur. L'extranet, accessible depuis le réseau internet et l'intranet qui doit être accessible uniquement depuis le réseau interne.

> ☞ Si vous travaillez sur machine physique, configurez les 4 machines : pare-feu ; internet ; intranet et extranet.

Lisez les slides 11 à 14 de la présentation suivant par Philippe biondi : `http://morgan.perso.enseeiht.fr/supports/philippe-biondi-ipv6.pdf`.

L'administrateur réseau de l'entreprise à configuré les règles de pare-feu suivantes avec le pare-feu iptables (xtables) :

```
# iptables -t filter -P FORWARD DROP
# iptables -t tilfer -A FORWARD -s 2002::0/64 -d 2000::2 -j ACCEPT
# iptables -t tilfer -A FORWARD -s 2000::0/64 -d 2002::/64 -j ACCEPT
```

Installez ces règles de pare-feu et constatez leur fonctionnement avec ICMP.

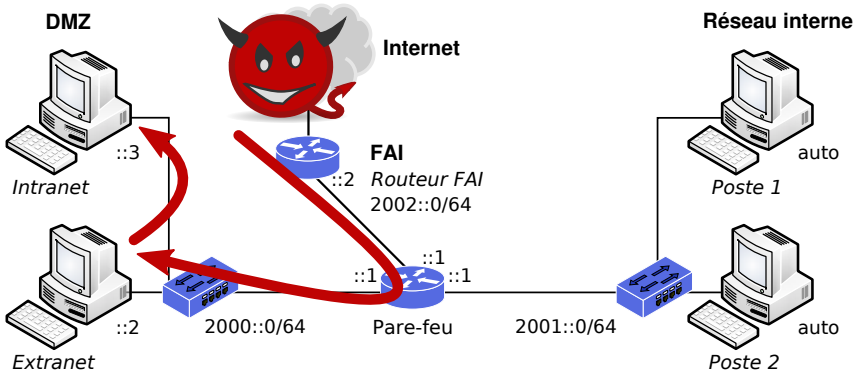

FIGURE 3 – Attaquant externe : *routing header* de type 0

Proposez une méthode générique d'attaque permettant de contourner le filtrage réseau effectué par le pare-feu, en utilisant des paquets IPv6 étendus avec le *routing header* de type 0 (inspirez-vous de la figure 3).

Dans notre cas, l'attaquant est placé sur la machine internet et veut envoyer une requête ICMPv6 *echo* à la machine intranet. Le réseau DMZ ne dispose pas de routeur supplémentaire derrière le pare-feu.

— Pour mener à bien cet exercice, activez le relayage IPv6 et sur la machine extranet, de façon à la transformer temporairement en routeur. — Utilisez l'outil `scapy` pour générer l'attaque.

Observez les échanges réseau à l'aide de *wireshark* pour confirmer que l'attaque est fonctionnelle.

Selon-vous est-ce que ce type de règle est suffisant pour filtrer précisément des paquets d'une même connexion TCP ?

Notons que les *extension headers* de type 0 ont été dépréciés par RFC en 2007 pour des raisons évidentes de sécurité. Au même titre que les extension de *source routing* IPv4.