Utilisation de DHCP pour contourner routeurs et pare-feux

Olivier Bal-Pétré
olivier.bal-petre@ssi.gouv.fr

Résumé. Le protocole DHCP permet de configurer automatiquement les machines d'un réseau et de leur fournir une adresse IP. Il permet également d'ajouter des routes dans leurs tables de routage. Cet article abuse de cette dernière fonctionnalité pour détourner du trafic à son avantage. Combinée à des erreurs dans la logique de génération des règles de filtrage de certains pare-feux, un attaquant peut accéder à des ressources qui lui sont en théorie inaccessibles. Cette attaque a été testée avec succès sur différents pare-feux du marché.

1 Introduction

DHCP (Dynamic Host Configuration Protocol) est un protocole construit sur un modèle client–serveur où, dynamiquement, le serveur alloue les adresses IP et fournit des configurations aux clients [1]. Il est massivement utilisé dans les réseaux résidentiels et publics et par les opérateurs de télécommunications. La grande majorité des fournisseurs de Cloud l'ont également adopté pour le déploiement des machines virtuelles.

Dans un premier temps, nous présenterons les différentes méthodes que le protocole offre pour modifier les tables de routage des clients DHCP. Dans un second temps, nous étudierons comment cette technique peut être utilisée pour abuser des pare-feux et plus largement toute machine effectuant du routage. Enfin, nous proposerons un ensemble de recommandations et de contre-mesures.

2 Ajout de routes dynamiques avec DHCP

L'offre de bail (*lease*) DHCP envoyée au client inclut une adresse IP et une liste d'options [2]. Parmi celles-ci, les suivantes permettent au serveur de créer des routes sur le système du client :

— **Adresse IP + Subnet Mask (opt 1)** – Automatiquement, le client crée une route pour le sous-réseau de l'adresse IP offerte.
— **Classless Static Route Option (opt 121) [3]** – Liste de routes arbitraires que le client rajoute à sa table de routage.[1]

[1] L'option 33 et l'option vendeur 249 peuvent être utilisées de manière similaire quand elles sont supportées par le client.

— **Router Option (opt 3)** – Le client ajoute une route par défaut vers les routeurs, mais parfois aussi une route dédiée (`/32`).

Le choix d'une entrée dans la table de routage se fait en priorité sur la longueur du masque de sous-réseau : l'entrée avec le plus long est préférée. Autrement dit, la route la plus spécifique est choisie. Ainsi, dans l'exemple du Tableau 1 où trois routes ont été ajoutées via DHCP aux deux routes statiques :

— Un paquet à destination de `10.0.2.33` correspond uniquement à la route n°2 et est routé sur l'interface `eth2`.

— Un paquet à destination de `10.0.2.142` correspond aussi à la route n°2, mais également à la route n°4 ajoutée via DHCP. Cette dernière sera choisie car son masque de sous-réseau est plus long (`/25` contre `/24`). Le paquet est routé via la passerelle renseignée dans la route : `192.168.1.50 (eth0)`.

N°	Destination	Passerelle	Protocole	Dev	Raison de l'ajout
1	10.0.1.0/24		static	eth1	
2	10.0.2.0/24		static	eth2	
3	192.168.1.0/24		dhcp	eth0	IP + opt 1
4	10.0.2.128/25	192.168.1.50	dhcp	eth0	opt 121 (attaque 1)
5	10.0.1.0/25	192.168.1.50	dhcp	eth0	opt 121 (attaque 2)

Tableau 1. Table de routage

Ainsi, un serveur DHCP malveillant est en mesure de modifier la table de routage de ses clients, et, par conséquent, de rediriger tout le trafic qu'ils routent.

3 Attaques sur les routeurs et pare-feux

Nous prenons pour exemple la topologie réseau de la Figure 1 où l'attaquant est dans le réseau de gauche. Il est admis qu'il est en capacité de répondre aux requêtes DHCP du pare-feu. Pour cela, il peut être le serveur DHCP, l'avoir compromis, ou encore l'usurper sur le réseau.

Dans le cas `A. Nominal`, l'hôte 1 envoie un paquet à destination de l'hôte 2 que le pare-feu reçoit sur l'interface `eth1`. Celui-ci consulte sa table de routage (Tableau 1), qui dans le cas nominal ne contient aucun conflit (les routes n°4 et n°5 seront ajoutées respectivement dans les sections 3.1 et 3.2). Le pare-feu route le paquet sur l'interface `eth2`. Le paquet retour correspond à la route n°1 et il suit le même chemin en sens inverse.

3.1 Attaque 1 : Usurpation d'un service

L'attaquant cherche à usurper le service de l'hôte 2 afin de pouvoir présenter un site web malveillant à l'hôte 1. Pour cela, il fournit au client DHCP du pare-feu :

— une adresse IP et un masque : `192.168.1.25/24` ;
— une route malveillante, ici via l'option 121 : `10.0.2.128/25 via 192.168.1.50`. La plage IP inclut **l'adresse IP destination** à usurper. La passerelle est l'adresse IP de l'attaquant.

D'après la table de routage (Tableau 1), avec l'ajout de la route malveillante (route n°4), les paquets à destination de `10.0.2.142` sont maintenant routés à l'attaquant via l'interface `eth0`. Celui-ci répond à l'hôte 1 et usurpe ainsi le service. La route n°1 est utilisée pour router la réponse de l'attaquant.

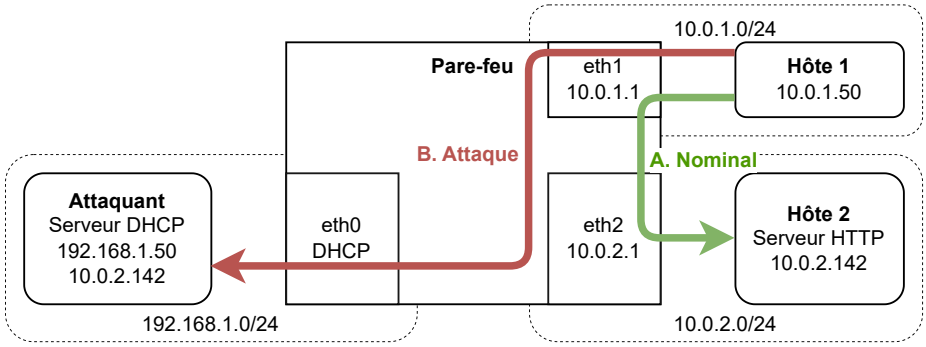

Fig. 1. Usurpation d'un service à travers le pare-feu

L'attaque nécessite que les règles de filtrage autorisent le flux à être transmis de `eth1` à `eth0` (*forward*).

3.2 Attaque 2 : Vol de connexion (Linux/Netfilter)

Pour ce second scénario, l'attaquant cherche à requêter lui-même le service de l'hôte 2. Cependant, le pare-feu est configuré pour n'autoriser que le trafic du cas nominal. Aucune connexion entre le réseau de gauche et ceux de droite n'est donc en théorie possible (Listing 1).

Listing 1: Règles de filtrage (nftables)

```
chain forward {
    type filter hook forward priority filter; policy drop;
    // Pare-feu "stateful" pour autoriser les paquets retour
    ct state established accept
    // Cas nominal : l'hôte-1 accède au service de l'hôte 2
    iifname eth1 oifname eth2 ip saddr 10.0.1.50 ip daddr
    ↪  10.0.2.142 tcp dport 80 accept
}
```

Cette fois-ci, l'attaquant fournit au client DHCP du pare-feu :

— une adresse IP et un masque : `192.168.1.25/24` ;

— une route malveillante, ici via l'option 121 : `10.0.1.0/25 via 192.168.1.50`. La plage IP inclut **l'adresse IP source** de la connexion à voler. La passerelle est l'adresse IP de l'attaquant.

L'exemple de l'établissement d'une connexion TCP (*3-way handshake*) est utilisé ci-dessous, mais l'attaque fonctionne de manière similaire pour UDP. Se référer à la Figure 2 pour suivre les étapes.

1. L'hôte 1 initie une connexion vers l'hôte 2 et émet un paquet SYN. Le pare-feu le reçoit sur l'interface `eth1` et le route à l'hôte 2 via l'interface `eth2` (route n°2). Afin de pouvoir identifier les futurs paquets de cette connexion, le mécanisme de suivi de connexion, *conntrack*, ajoute une entrée dans sa table de suivi de connexion :

```
tcp src=10.0.1.50 dst=10.0.2.142 sport=xxx dport=80
↪  src=10.0.2.142 dst=10.0.1.50 sport=80 dport=xxx
```

2. L'hôte 2 reçoit le SYN et répond un SYN-ACK que le pare-feu route à l'attaquant (route n°5). L'état `established` est attribué au paquet car il correspond à l'entrée *conntrack* créée par le SYN. En effet, *conntrack* ne prend pas en compte les interfaces et se base uniquement sur les informations des couches 3 et 4 du modèle OSI (IP src/dst, ports src/dst…). Le paquet est donc accepté par la première règle de pare-feu de la chaine `forward`.

3. L'attaquant répond [2] un ACK qui est routé à l'hôte 2 (route n°2). Ce paquet correspond à la même entrée *conntrack* que précédemment, et de la même manière, il est accepté par le pare-feu. L'attaquant vient de compléter le *TCP 3-way handshake* avec le serveur.

[2] L'attaquant reçoit le SYN-ACK sans avoir envoyé de SYN. Son système ne saura qu'en faire et le rejettera. Il doit donc écouter sur une *raw socket* et forger les paquets.

4. L'attaquant requête maintenant librement le service de l'hôte 2 en utilisant la session TCP établie.

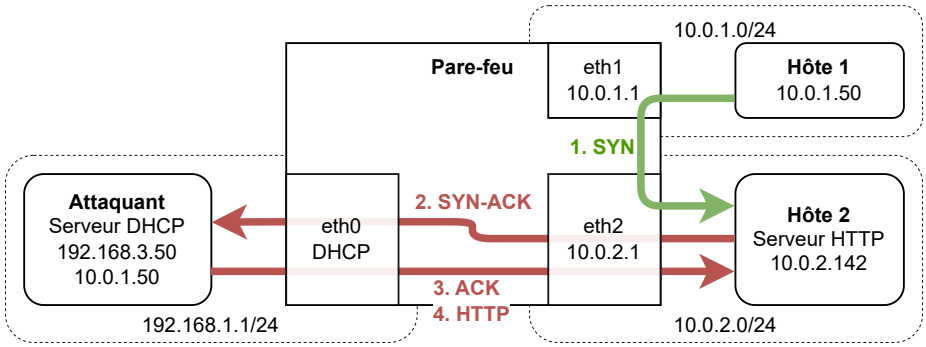

Fig. 2. Vol d'une connexion TCP traversant le pare-feu

Un attaquant peut donc, via l'ajout d'une route malveillante, rediriger les paquets retour vers lui-même et ainsi voler une connexion. Le pare-feu, à cause d'une règle *a priori* inoffensive, autorise le flux. Comme nous le verrons dans la section 4, cette règle est présente sur des produits de pare-feu reconnus et n'est généralement ni modifiable, ni supprimable.

3.3 Attaque 2 : Vol de connexion (BSD/PF)

Les systèmes BSD utilisant PF (Packet Filter) sont également vulnérables. Contrairement à Netfilter, les paquets aller et retour sont acceptés en une seule règle avec l'option `keep state`.

```
pass in quick on eth1 inet proto tcp from 10.0.1.50 to 10.0.2.142
↪  port = http flags S/SA keep state
```

Cependant, là encore, par défaut, PF accepte les paquets des connexions établies sans vérifier l'interface. Dans la table de suivi de connexions ci-dessous, le mot-clef `all` désigne l'interface.

```
all tcp 10.0.2.142:80 <- 10.0.1.50:xxx
all tcp 10.0.1.50:xxx -> 10.0.2.142:80
```

3.4 Attaque 2 : Vol de connexion (interfaces d'admin)

Un cas particulier de cette attaque est de voler une connexion à destination d'un service exposé par le pare-feu. Les interfaces d'administration

(web, SSH...) sont notamment une cible de choix. Celles-ci ne sont normalement exposées que sur une seule interface réseau en raison de leur sensibilité et des nombreuses vulnérabilités auxquelles elles sont sujettes.

4 Évaluation des pare-feux du marché

Les attaques présentées ont été testées sur différents pare-feux du marché configurés de manière similaire aux exemples de la section 3.

La majorité d'entre eux (Tableau 2) sont vulnérables aux deux attaques. Certains permettent de se protéger via une configuration avancée, mais celle-ci est à la charge de l'administrateur.

Pare-feu	OS Sous-jacent	Code Source	Opt 121	Attaques 1 & 2	Interface d'admin
CheckPoint USFW	Linux	closed		✓	✓
FortiGate	Linux	closed			
OpenWrt	Linux	open	✓	✓	✓
OPNsense	FreeBSD	open	✓	✓	✓
PfSense	FreeBSD	open	✓	✓	✓
Stormshield SNS	FreeBSD	closed		✓[3]	
Ubiquiti ER-X	Linux	closed		✓	✓

Tableau 2. Pare-feux évalués

Les vendeurs impactés ont été contactés, et deux d'entre eux ont choisi d'appliquer des contre-mesures pour aider à se prémunir de l'attaque 2. L'attaque 1 se base sur des règles de filtrage laxistes et n'est pas considérée comme une vulnérabilité. De son côté, le fait qu'un serveur DHCP envoie des routes malveillantes est un comportement prévu et décrit dans les RFC 2131 [1] et 3442 [3]. DHCP est donc comparable aux protocoles de routage tels que RIP, OSPF, ou BGP, et il convient de prendre les mêmes précautions quant à son emploi.

5 Cas d'usage impactés

Les pare-feux du marché sont étudiés dans cet article, mais toute machine effectuant du routage et ayant un client DHCP est concernée. Notamment, il est commun que les serveurs virtualisés obtiennent leur IP

[3] Nécessite de désactiver l'IPS pour la règle de pare-feu concernée (activé par défaut).

par DHCP. Si ceux-ci hébergent des VM et containers, ou possèdent des tunnels VPN, alors ils effectuent probablement des actions de routage et sont impactés. Pour la même raison, les machines d'administration ou de développement peuvent faire du routage et être vulnérables à ces attaques.

Enfin, la majorité des VPN utilisent le routage pour déterminer quels paquets envoyer dans le tunnel. Une route malveillante peut donc porter atteinte à la confidentialité du trafic. L'attaque 1 a d'ailleurs déjà été testée en 2023 [4] sur 67 clients VPN et 64.6% d'entre eux étaient vulnérables.

6 Contre-mesures et recommandations

6.1 Pour le client DHCP

L'option 121 [3] n'est pas la seule manière d'ajouter des routes avec DHCP, mais elle est la plus puissante pour un attaquant. Il est donc recommandé de configurer son client pour ne pas l'accepter.

Certains clients DHCP proposent une option[4,5] indiquant dans quelle table de routage ajouter les routes obtenues. Via le mécanisme de *policy-based routing* il est possible de configurer cette table comme moins prioritaire que celles hébergeant les routes statiques et d'ainsi éviter toute redirection via l'ajout de route malveillante.

6.2 Pour les règles de pare-feu (Linux/Netfilter)

Il ne faut pas accepter les paquets d'une connexion établie sans vérifier les interfaces d'entrée et de sortie. À la place, il est recommandé de créer deux règles par flux : une pour l'aller, et une pour le retour.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    iifname eth1 oifname eth2 ip saddr 10.0.1.50 ip daddr
    ↪  10.0.2.142 tcp dport 80 ct state {new,established} accept
    oifname eth1 iifname eth2 ip daddr 10.0.1.50 ip saddr
    ↪  10.0.2.142 tcp sport 80 ct state {established} accept
}
```

6.3 Pour les règles de pare-feu (BSD/PF)

Par défaut, PF suit les connexions pour l'ensemble des interfaces. Pour qu'il les associe à une unique interface, il faut configurer l'option `state-policy` ou surcharger chaque règle :

[4] `man systemd.network(5)` section `DHCPv4`, option `RouteTable=`
[5] `man NetworkManager.conf(5)`, option `ipv4.route-table`

```
// Option globale appliquée à l'ensemble des règles
set state-policy if-bound

// Option surchargée pour une règle
pass in quick on eth1 inet proto tcp from 10.0.1.50 to 10.0.2.142
↪  port = http flags S/SA keep state (if-bound)
```

On peut vérifier que chaque connexion est associée à une interface :

```
eth1 tcp 10.0.2.142:80 <- 10.0.1.50:xxx
eth2 tcp 10.0.1.50:xxx -> 10.0.2.142:80
```

6.4 Reverse Path Filtering (RPF)

Le RPF bloque les paquets émis par l'hôte 1 lors de l'attaque 2, empêchant notamment la création de nouvelles connexions (paquet SYN). Cependant, l'attaquant contrôle les routes du pare-feu. Il peut donc laisser l'hôte 1 établir une connexion, puis changer les routes et voler la connexion établie. Les clients DHCP acceptent des baux de quelques secondes, rendant cette technique efficace. L'attaque devient tout de même significativement plus compliquée avec le RPF, et il donc est conseillé de l'activer.

7 Conclusions

La mauvaise compréhension des systèmes de suivi de connexion de Linux et BSD amène les éditeurs et nombre d'administrateurs à écrire des règles de pare-feu vulnérables. Un mécanisme peu connu de DHCP nous permet ici de les exploiter, mais toute autre technique permettant de manipuler les tables de routage d'une machine fonctionne également. Suite aux remontées faites dans le cadre de ce travail, certains éditeurs ont appliqué des contre-mesures adéquates. D'autres produits et cas d'usage sont cependant aussi impactés.

Références

1. Ralph Droms. Dynamic Host Configuration Protocol. RFC 2131, March 1997. https://www.rfc-editor.org/info/rfc2131.
2. Ralph Droms and Steve Alexander. DHCP Options and BOOTP Vendor Extensions. RFC 2132, March 1997. https://www.rfc-editor.org/info/rfc2132.
3. Ted Lemon, Stuart Cheshire, and Bernie Volz. The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) version 4. RFC 3442, Dec. 2002. https://www.rfc-editor.org/info/rfc3442.
4. Nian Xue, Yashaswi Malla, Zihang Xia, Christina Pöpper, and Mathy Vanhoef. Bypassing tunnels : Leaking VPN client traffic by abusing routing tables. Aug. 2023.