Utilisation de DHCP pour contourner routeurs et pare-feux

Olivier Bal-Pétré

5-7 juin, SSTIC 2024

Un client DHCP sur un pare-feu ?

Pare-feux “traditionnels”

- Connectés à des réseaux tiers (WAN…)

Un client DHCP sur un pare-feu ?

Pare-feux “traditionnels”

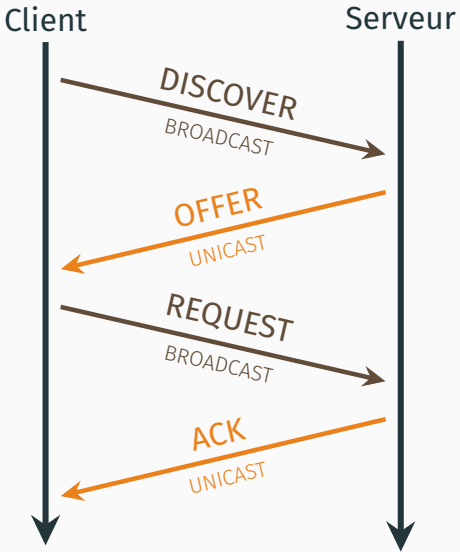
- Connectés à des réseaux tiers (WAN…)

Machines hébergeant containers, VM ou VPN → routeur/pare-feu

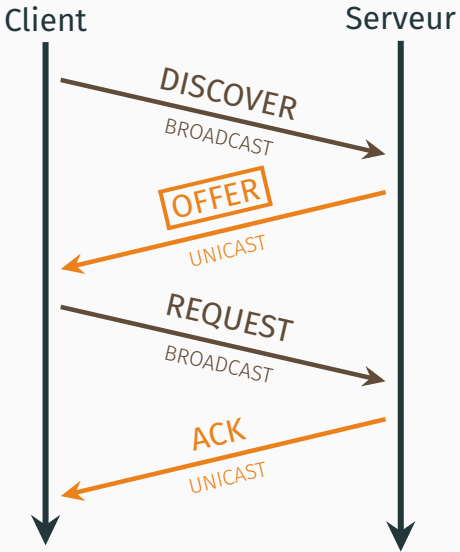
- Postes de travail d’admin, de dev, multi-niveaux…
- Serveurs virtualisés

Influence de DHCP sur le routage

DHCP – Échanges



DHCP – Échanges



DHCP – Format d'un message OFFER

```
Client IP address: 0.0.0.0
Your (client) IP address: 192.168.167.38
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: Dell_de:7a:2f (98:e7:43:de:7a:2f)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
▸ Option: (53) DHCP Message Type (Offer)
▸ Option: (54) DHCP Server Identifier (192.168.167.1)
▸ Option: (51) IP Address Lease Time
▸ Option: (1) Subnet Mask (255.255.255.0)
▸ Option: (3) Router
▸ Option: (6) Domain Name Server
▸ Option: (255) End
Padding: 000000000000000000000000000000000000000000000000000000
```

Adresse IP proposée

Liste d'options

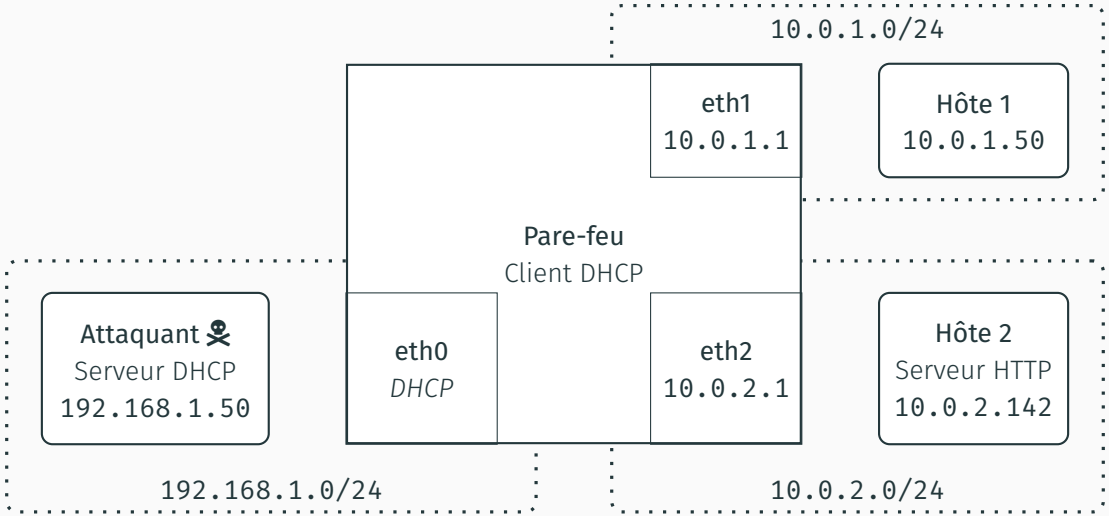
DHCP – Influence sur la table de routage

Options DHCP influant sur la table de routage du système du client

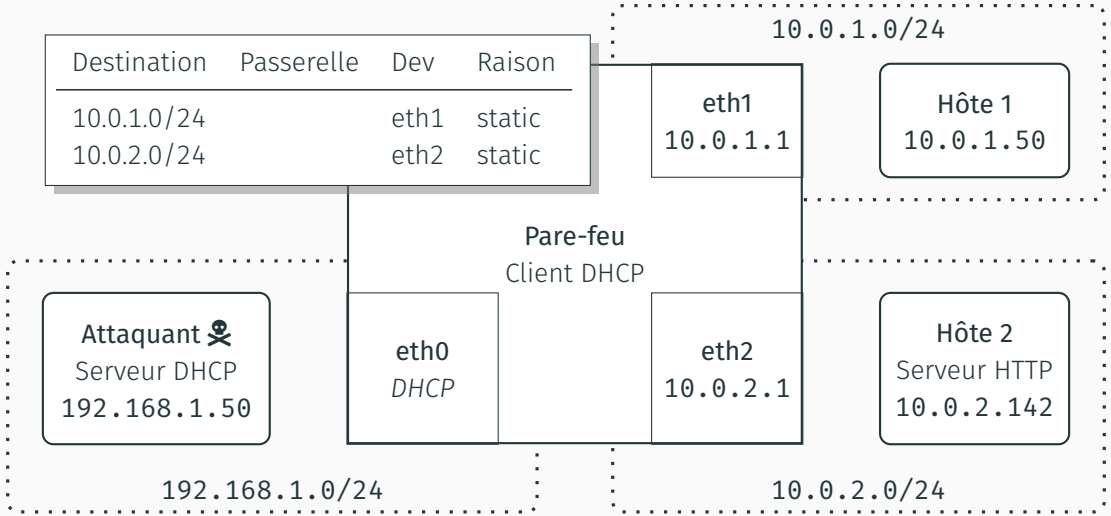
- Adresse IP + Subnet Mask (opt 1)
  - Route pour le sous-réseau : `IP/Mask`
- Classless Static Route Option (opt 121)
  - Liste de routes arbitraires
- Router Option (opt 3)
- Static Route Option (opt 33)
- Microsoft Classless Static Route Option (vendor opt 249)

Attaques

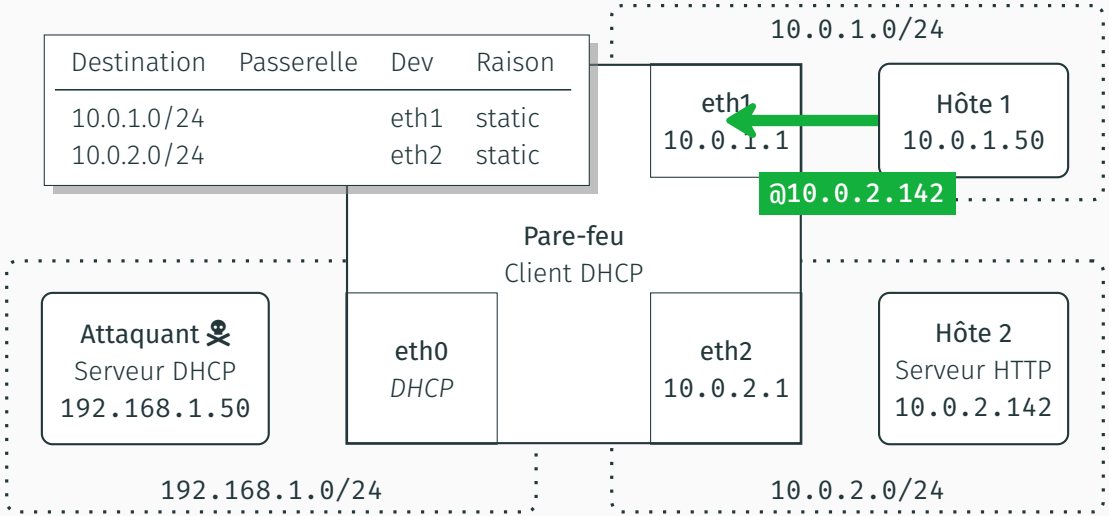
Cas nominal


10.0.1.0/24
eth1
10.0.1.1
Hôte 1
10.0.1.50
Pare-feu
Client DHCP
Attaquant
Serveur DHCP
192.168.1.50
eth0
DHCP
eth2
10.0.2.1
Hôte 2
Serveur HTTP
10.0.2.142
192.168.1.0/24
10.0.2.0/24

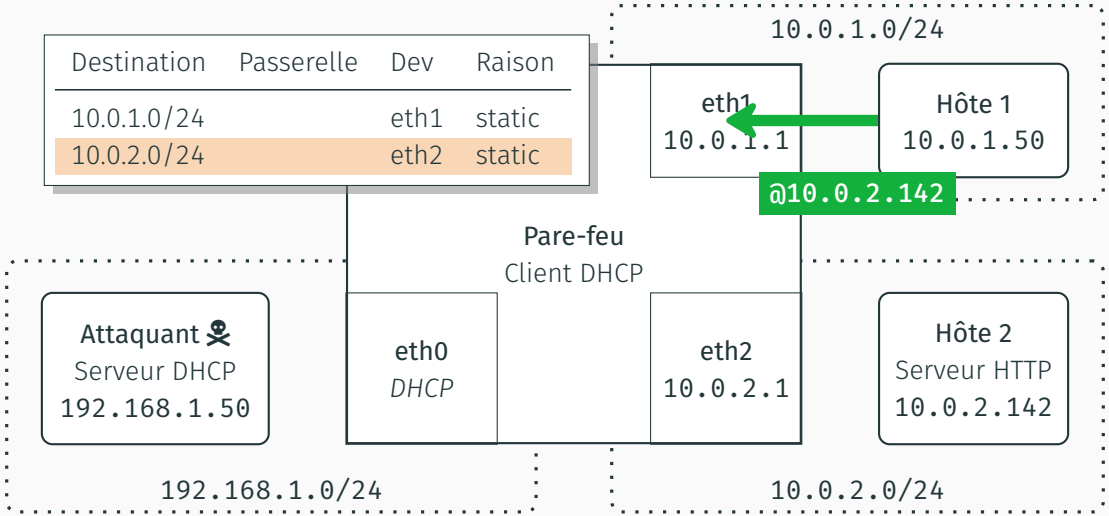
Cas nominal


Destination Passerelle Dev Raison
10.0.1.0/24 eth1 static
10.0.2.0/24 eth2 static
10.0.1.0/24
eth1
10.0.1.1
Hôte 1
10.0.1.50
Pare-feu
Client DHCP
Attaquant
Serveur DHCP
192.168.1.50
eth0
DHCP
eth2
10.0.2.1
Hôte 2
Serveur HTTP
10.0.2.142
192.168.1.0/24
10.0.2.0/24

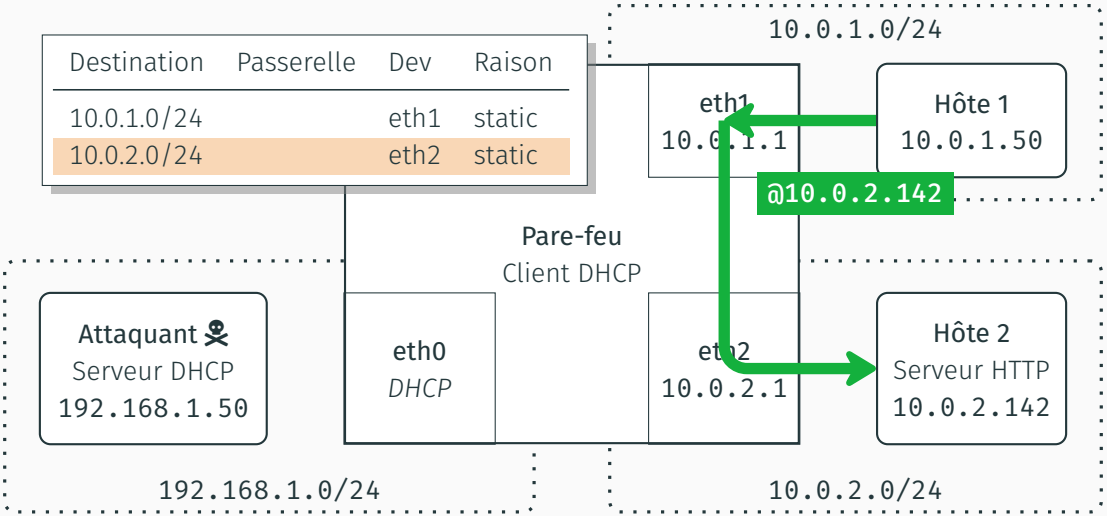
Cas nominal


Destination Passerelle Dev Raison
10.0.1.0/24 eth1 static
10.0.2.0/24 eth2 static
10.0.1.0/24
eth1
10.0.1.1
Hôte 1
10.0.1.50
@10.0.2.142
Pare-feu
Client DHCP
Attaquant
Serveur DHCP
192.168.1.50
eth0
DHCP
eth2
10.0.2.1
Hôte 2
Serveur HTTP
10.0.2.142
192.168.1.0/24
10.0.2.0/24

Cas nominal


Destination Passerelle Dev Raison
10.0.1.0/24 eth1 static
10.0.2.0/24 eth2 static
10.0.1.0/24
eth1
10.0.1.1
Hôte 1
10.0.1.50
@10.0.2.142
Pare-feu
Client DHCP
Attaquant
Serveur DHCP
192.168.1.50
eth0
DHCP
eth2
10.0.2.1
Hôte 2
Serveur HTTP
10.0.2.142
192.168.1.0/24
10.0.2.0/24

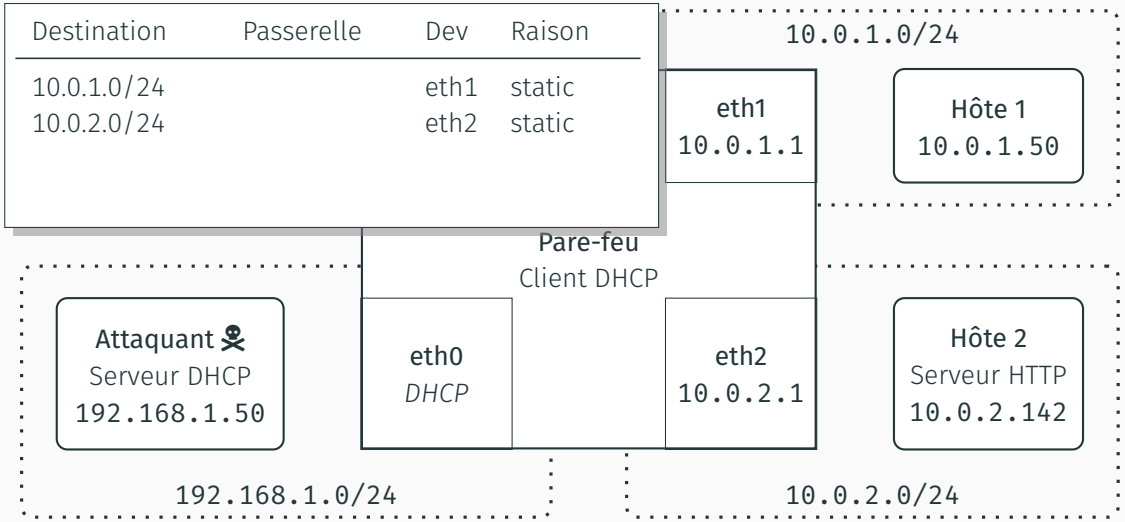
Cas nominal


Destination Passerelle Dev Raison
10.0.1.0/24 eth1 static
10.0.2.0/24 eth2 static
10.0.1.0/24
eth1
10.0.1.1
Hôte 1
10.0.1.50
@10.0.2.142
Pare-feu
Client DHCP
Attaquant
Serveur DHCP
192.168.1.50
eth0
DHCP
eth2
10.0.2.1
Hôte 2
Serveur HTTP
10.0.2.142
192.168.1.0/24
10.0.2.0/24

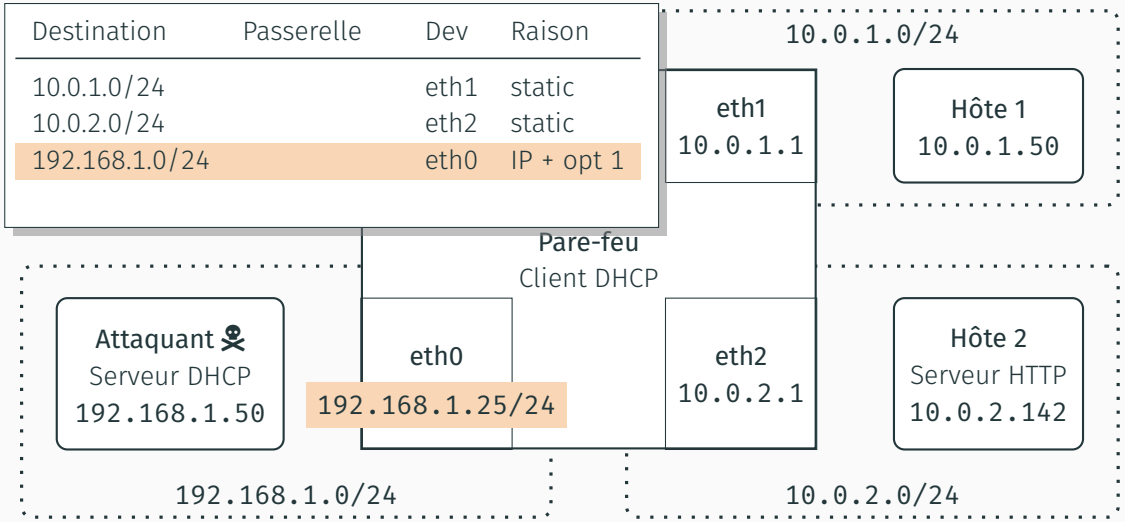
Attaque 1 – Usurpation d’un service

Attaque 1 – Usurpation d'un service

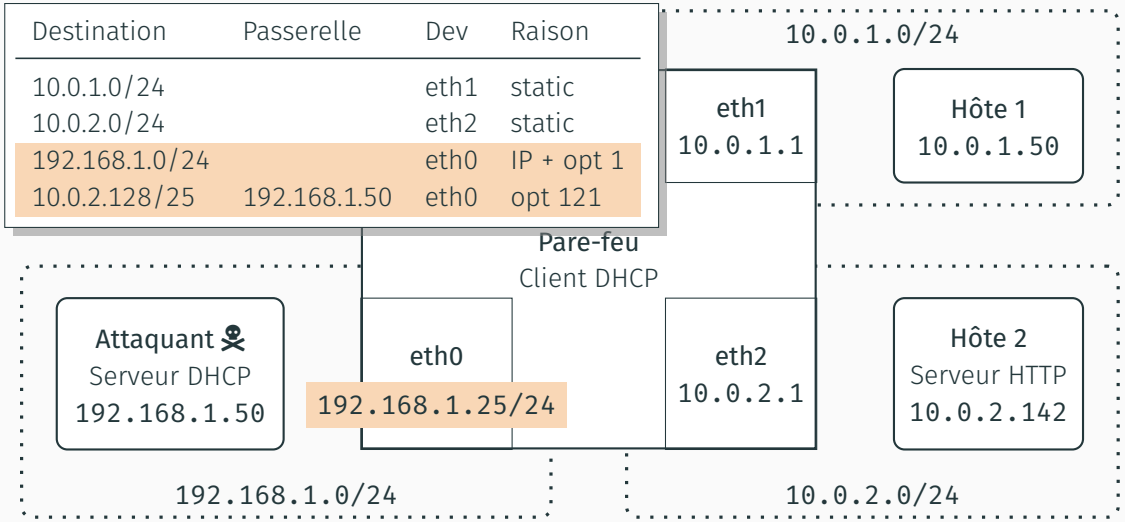
Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static



Attaque 1 – Usurpation d'un service

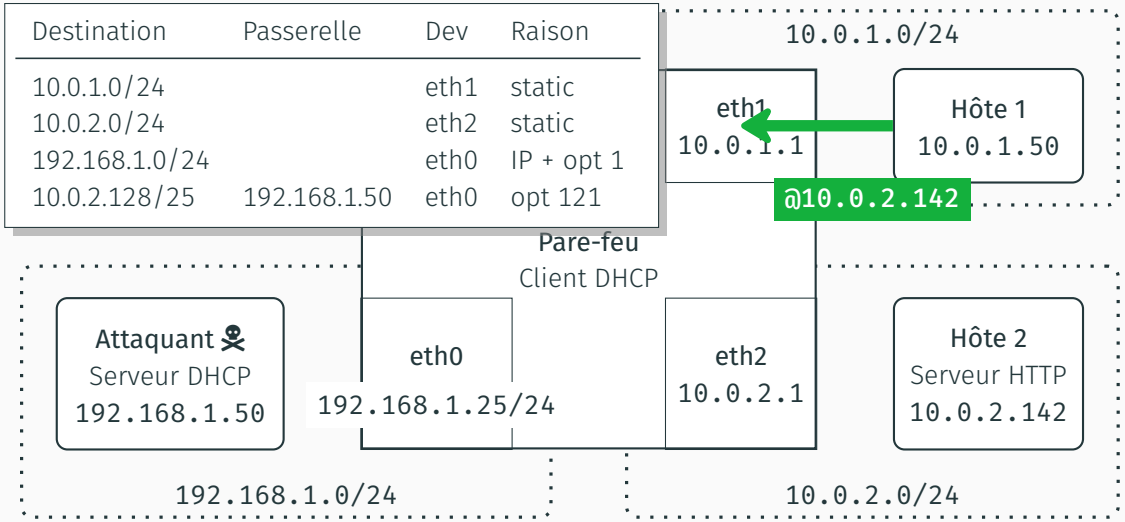


Attaque 1 – Usurpation d'un service



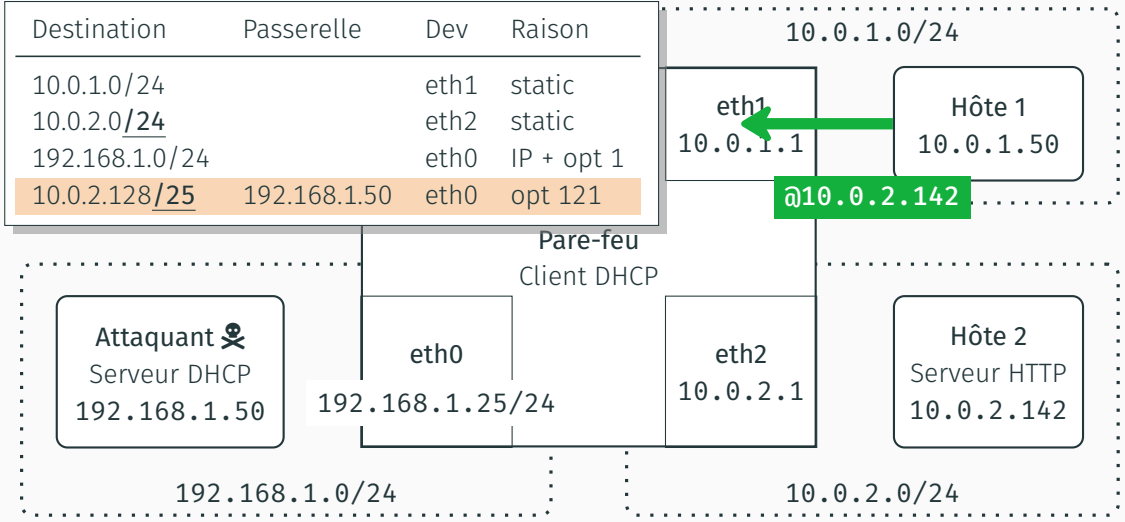
Attaque 1 – Usurpation d'un service

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.2.128/25	192.168.1.50	eth0	opt 121



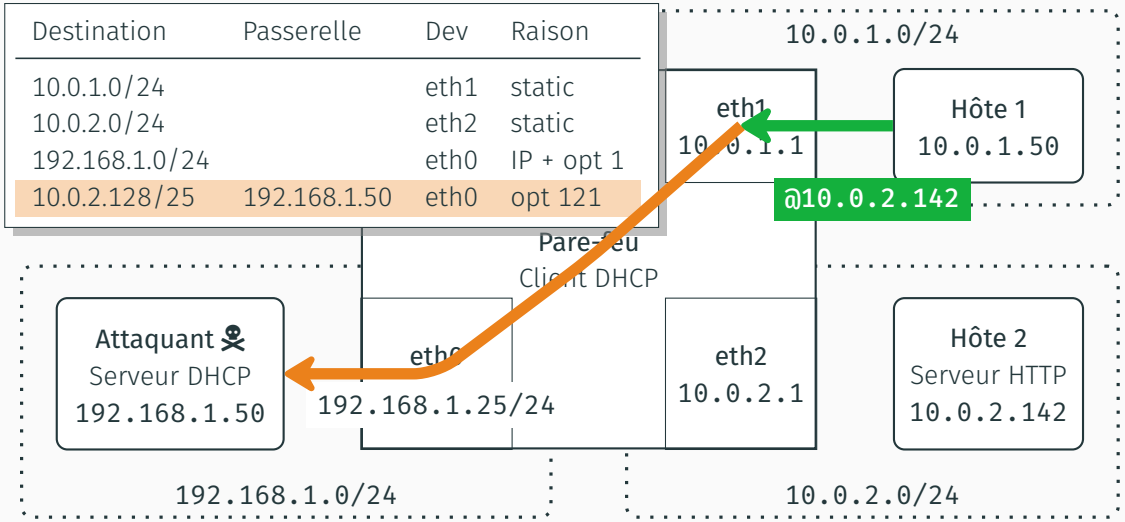
Attaque 1 – Usurpation d'un service

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.2.128/25	192.168.1.50	eth0	opt 121



Attaque 1 – Usurpation d'un service

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.2.128/25	192.168.1.50	eth0	opt 121



Attaque 1 – Cas particulier des VPN

Cas particulier : les VPN

- Interface virtuelle
- Choix des paquets via le routage
- Fuite de données en clair

Attaque 1 – Cas particulier des VPN

Cas particulier : les VPN

- Interface virtuelle
- Choix des paquets via le routage
- Fuite de données en clair

TunnelCrack, Août 2023, USENIX

- *Bypassing Tunnels: Leaking VPN Client Traffic by Abusing Routing Tables*

TunnelVision, Mai 2024

Attaque 1 – Récapitulatif

Hypothèses

- Attaquant en mesure de répondre aux requêtes DHCP du pare-feu

Attaque

- Re-routage des paquets à destination de l'Hôte 2 (serveur)

Impact

- Usurper un service derrière le pare-feu
- Faire fuiter des données

Attaque 1 – Récapitulatif

Hypothèses

- Attaquant en mesure de répondre aux requêtes DHCP du pare-feu
- **Paquets acceptés de l'Hôte 1 vers l'Attaquant (chaine *forward*)**

Attaque

- Re-routage des paquets à destination de l'Hôte 2 (serveur)

Impact

- Usurper un service derrière le pare-feu
- Faire fuiter des données

Comment

...

exécuter l’attaque avec des règles de pare-feu ?

accéder directement aux services de l’Hôte 2 ?

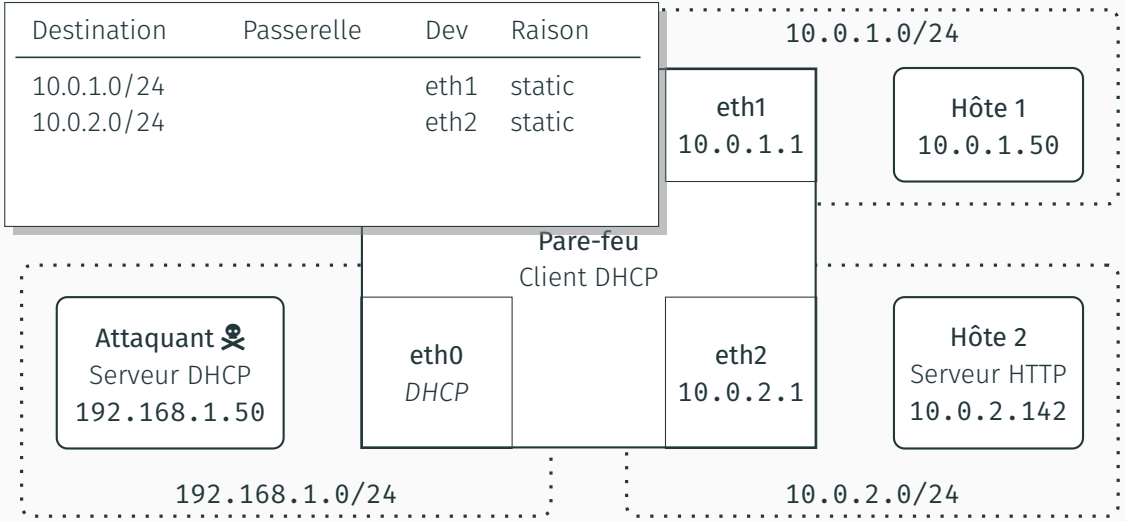
Attaque 2 – Vol de connexion

Attaque 2 – Règles de pare-feu

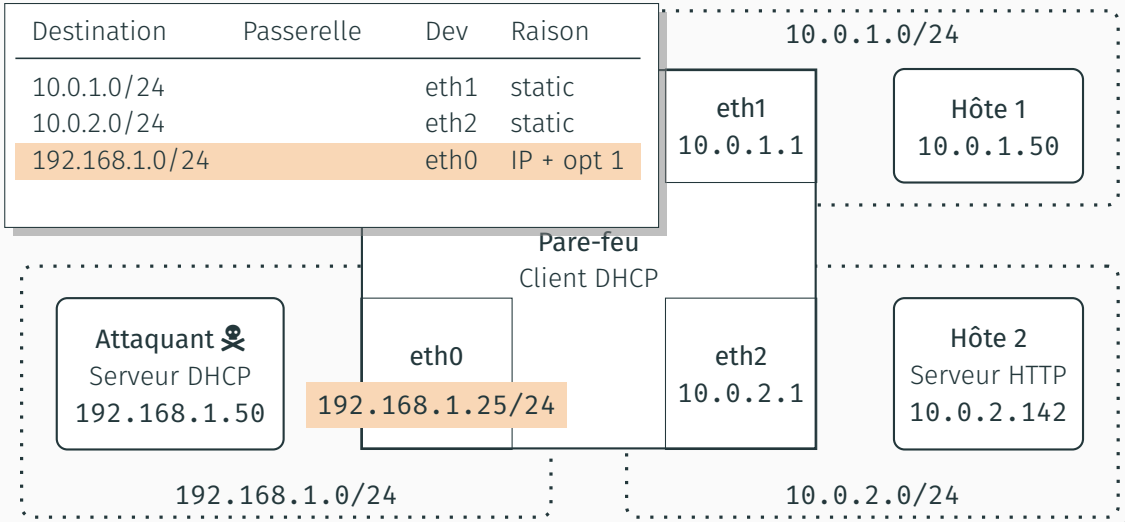
```
chain forward {
    type filter hook forward priority filter; policy drop;

    # Pare-feu "stateful"
    ct state established accept

    # De HOTE 1 vers HOTE 2
    iifname eth1 oifname eth2 \
        ip saddr 10.0.1.50 ip daddr 10.0.2.142 \
        tcp dport 80 accept
}
```

Attaque 2 – Vol de connexion

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static

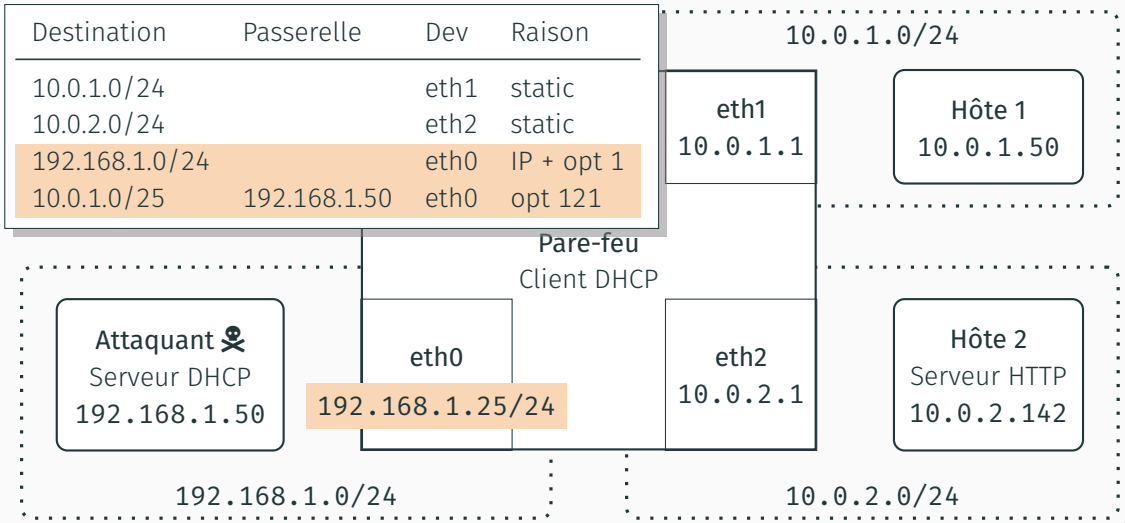


Attaque 2 – Vol de connexion



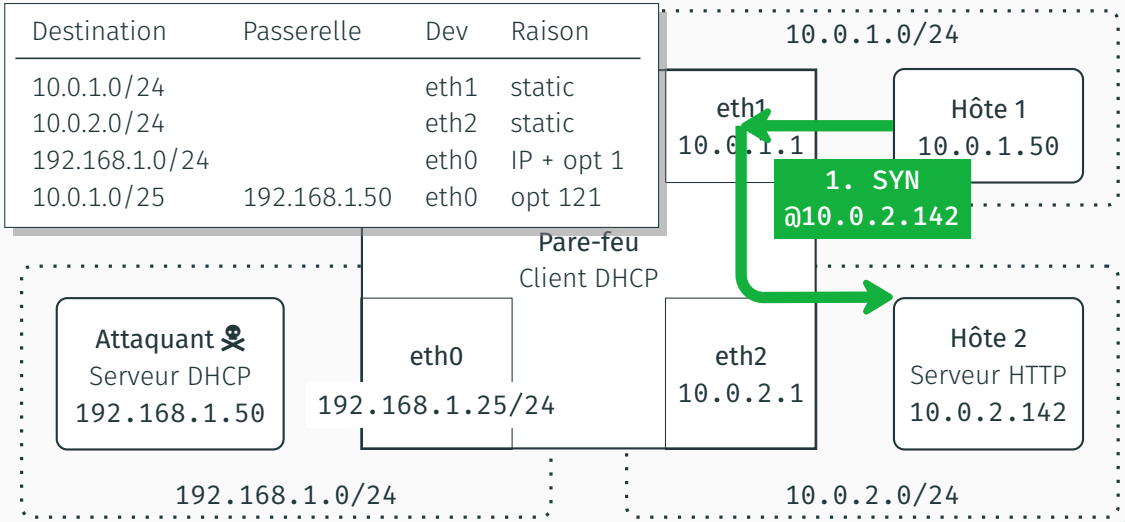
Attaque 2 – Vol de connexion

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.1.0/25	192.168.1.50	eth0	opt 121



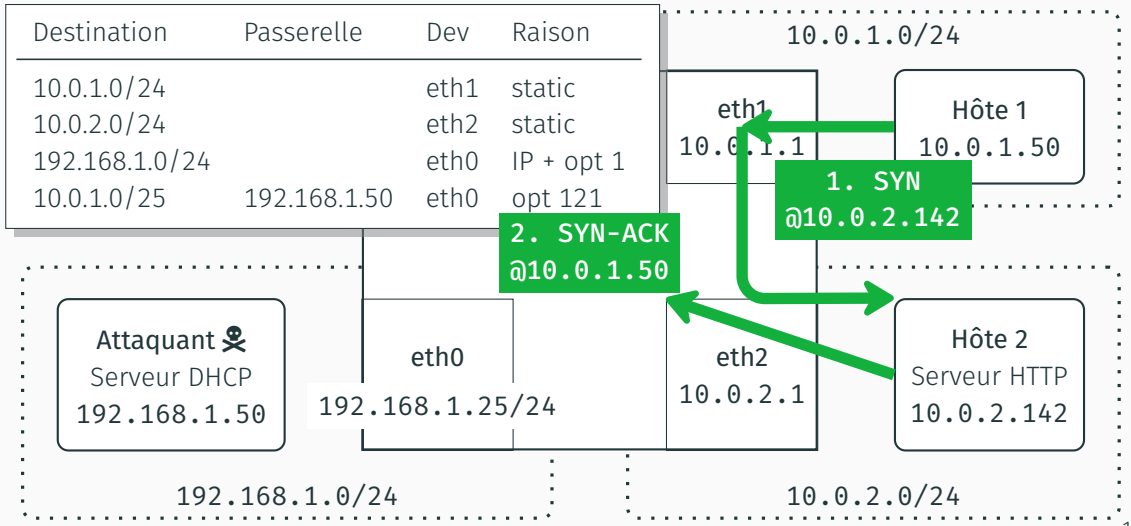
Attaque 2 – Vol de connexion

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.1.0/25	192.168.1.50	eth0	opt 121



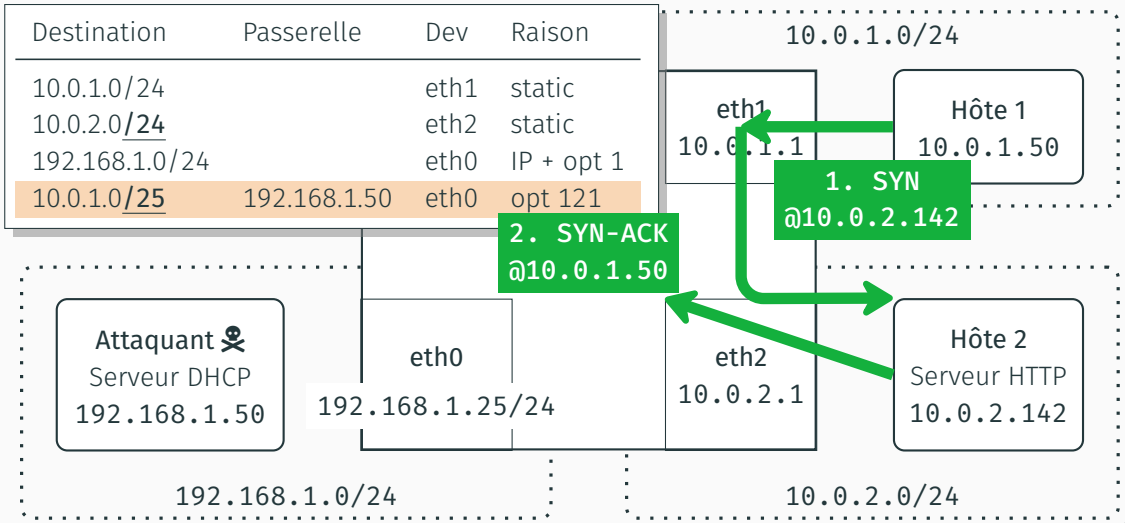
Attaque 2 – Vol de connexion

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.1.0/25	192.168.1.50	eth0	opt 121



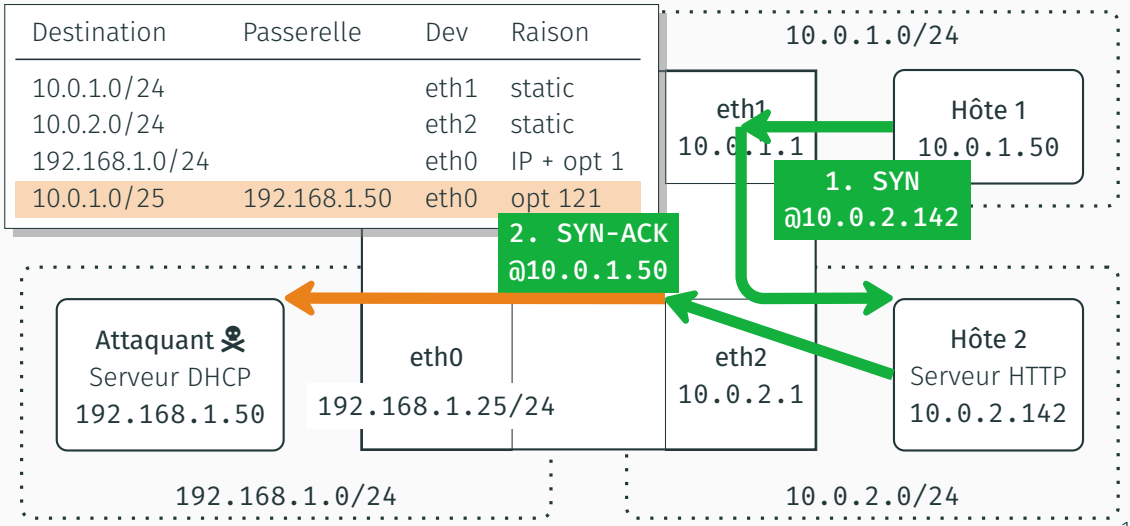
Attaque 2 – Vol de connexion

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.1.0/25	192.168.1.50	eth0	opt 121



Attaque 2 – Vol de connexion

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.1.0/25	192.168.1.50	eth0	opt 121

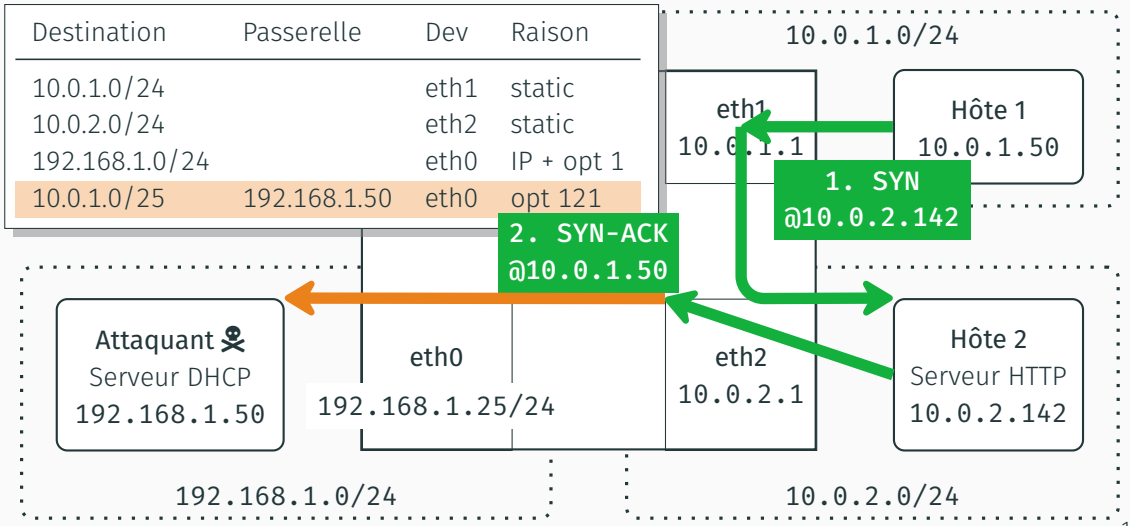


Attaque 2 – Vol de connexion

```
chain forward {
    type filter hook forward priority filter; policy drop;

    ct state established accept

    iifname eth1 oifname eth2 \
        ip saddr 10.0.1.50 ip daddr 10.0.2.142 \
        tcp dport 80 accept
}

# Table de suivi de connexion conntrack
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
```

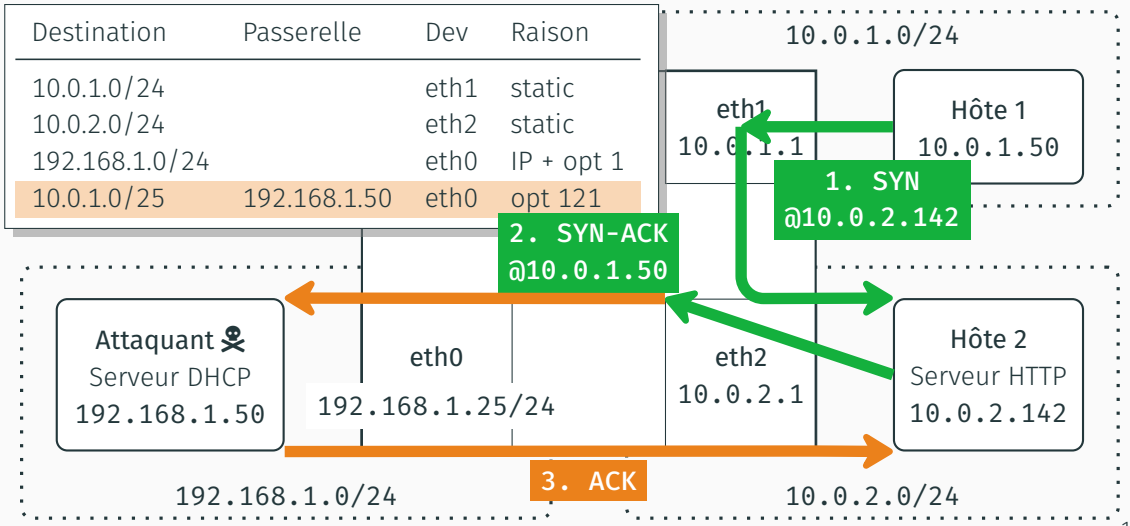
Attaque 2 – Vol de connexion

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.1.0/25	192.168.1.50	eth0	opt 121



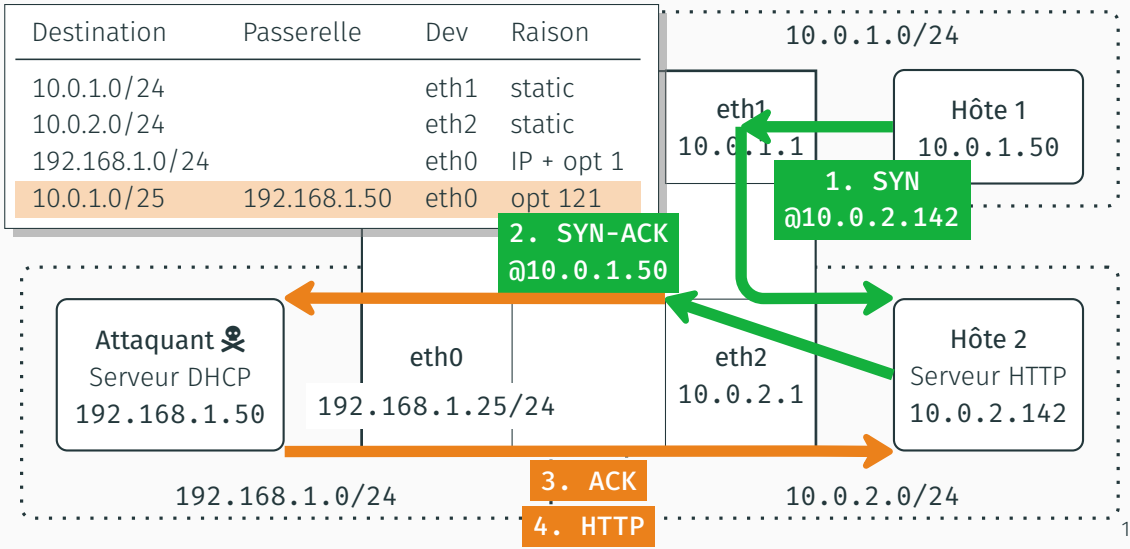
Attaque 2 – Vol de connexion

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.1.0/25	192.168.1.50	eth0	opt 121



Attaque 2 – Vol de connexion

Destination	Passerelle	Dev	Raison
10.0.1.0/24		eth1	static
10.0.2.0/24		eth2	static
192.168.1.0/24		eth0	IP + opt 1
10.0.1.0/25	192.168.1.50	eth0	opt 121



Attaque 2 – Cas particulier des services exposés par le pare-feu

Cas particulier : les services exposés par le pare-feu

- Console d'admin Web, SSH…
- Exposition généralement restreinte

Attaque 2 – Récapitulatif

Hypothèses

- Attaquant en mesure de répondre aux requêtes DHCP du pare-feu
- ~~Paquets acceptés de l'Hôte 1 vers l'Attaquant (chaine *forward*)~~
- Pare-feu *stateful* : `ct state established accept`

Attaque

- Re-routage des paquets "retours" à destination de l'Hôte 1 (client)

Impact

- Vol des connexions traversant le pare-feu

Existe-t-il des pare-feux acceptant tous les paquets des connexions établies ?

Pare-feux du marché

Évaluation des pare-feux du marché

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

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

Évaluation des pare-feux du marché (Contre-mesures)

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

Contre-mesures : Adéquates ☺ Partielles 😐 Inexistantes ☹

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

Contre-mesures

Contre-mesures – Client DHCP et tables de routage

Empêcher toute création de routes

Contre-mesures – Client DHCP et tables de routage

~~Empêcher toute création de routes~~

Ne pas accepter les routes conflictuelles

Contre-mesures – Client DHCP et tables de routage

~~Empêcher toute création de routes~~

~~Ne pas accepter les routes conflictuelles~~

Utiliser les *metrics* des routes

Contre-mesures – Client DHCP et tables de routage

~~Empêcher toute création de routes~~

~~Ne pas accepter les routes conflictuelles~~

~~Utiliser les *metrics* des routes~~

Utiliser du *policy-based routing*

→ Linux : `man ip-rule(8)`

Contre-mesures – Pare-feu Linux/Netfilter

```
chain forward {
    type filter hook forward priority filter; policy drop;

    ct state established accept

    iifname eth1 oifname eth2 … accept




}
```

Contre-mesures – Pare-feu Linux/Netfilter

```
chain forward {
    type filter hook forward priority filter; policy drop;

    ct state established accept

    iifname eth1 oifname eth2 … accept




}
```

Contre-mesures – Pare-feu Linux/Netfilter

```
chain forward {
    type filter hook forward priority filter; policy drop;

    ct state established accept

    iifname eth1 oifname eth2 … accept

    iifname eth1 oifname eth2 … ct state {new,established} acce

    iifname eth2 oifname eth1 … ct state {established} accept
}
```

Contre-mesures – Reverse Path Filtering

Contre l'usurpation d'IP (*antispoofing*)

Basé sur les routes … sous le contrôle de l'attaquant

Complique fortement la tâche de l'attaquant

Conclusion

Conclusion

Mécanismes de suivi de connexion mal utilisés

- Linux $\rightarrow$ netfilter + conntrack
- BSD $\rightarrow$ PF (Plus de détails dans les Actes)

$\Rightarrow$ Vol de connexion

Exploitable via tout moyen de manipuler des routes

- DHCP
- Protocoles de routage (OSPF, ...)

Questions ?