université de BORDEAUX

DNS CACHE POISONNING ATTACK RELOADED

Auteurs :
Hanane BENARAB
Youssef LACHABI
Alexandre ROSE

Encadrant :
Abdou GUERMOUCHE

SOMMAIRE

Objectif du PROJET

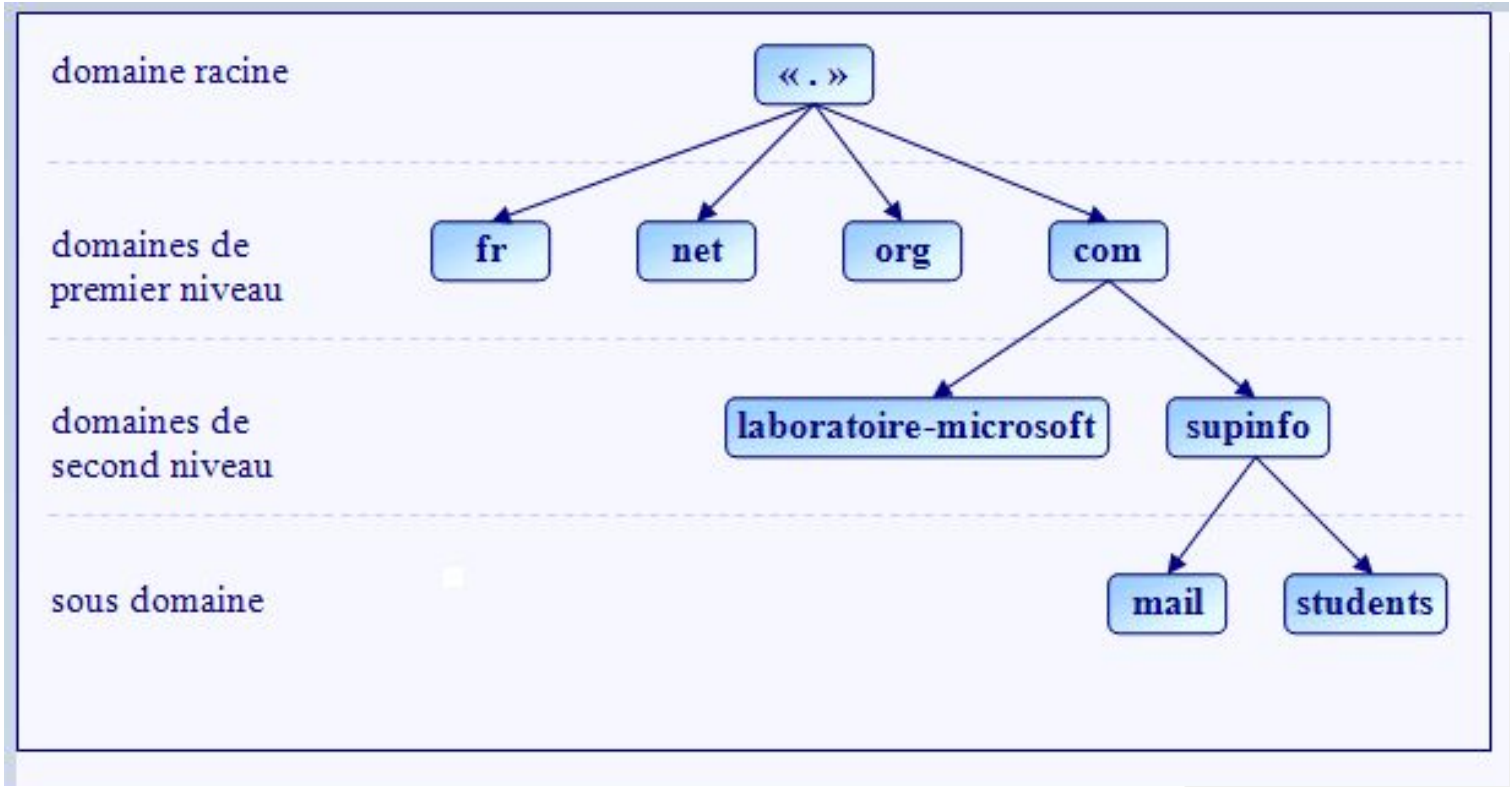
1. Comprendre le fonctionnement de DNS
2. Analyser les techniques d'attaques par empoisonnement du cache DNS
3. Explorer des solutions de prévention
4. Comment renforcer la sécurité des infrastructure DNS

DOMAIN NAME SYSTEM

DNS est responsable de la résolution de nom de domaine en une adresse IP correspondante.

Il fonctionne comme un annuaire Internet qui associe les noms de domaine aux adresses IP des serveurs où se trouvent les sites Web et les autres ressources en ligne.

Hiérarchie de DNS



Type de serveurs DNS

Serveur DNS Local (Resolveur)

Responsable de la résolution des requêtes DNS

Serveur de nom racine (Root)

Fournir des informations sur les serveur DNS TLD

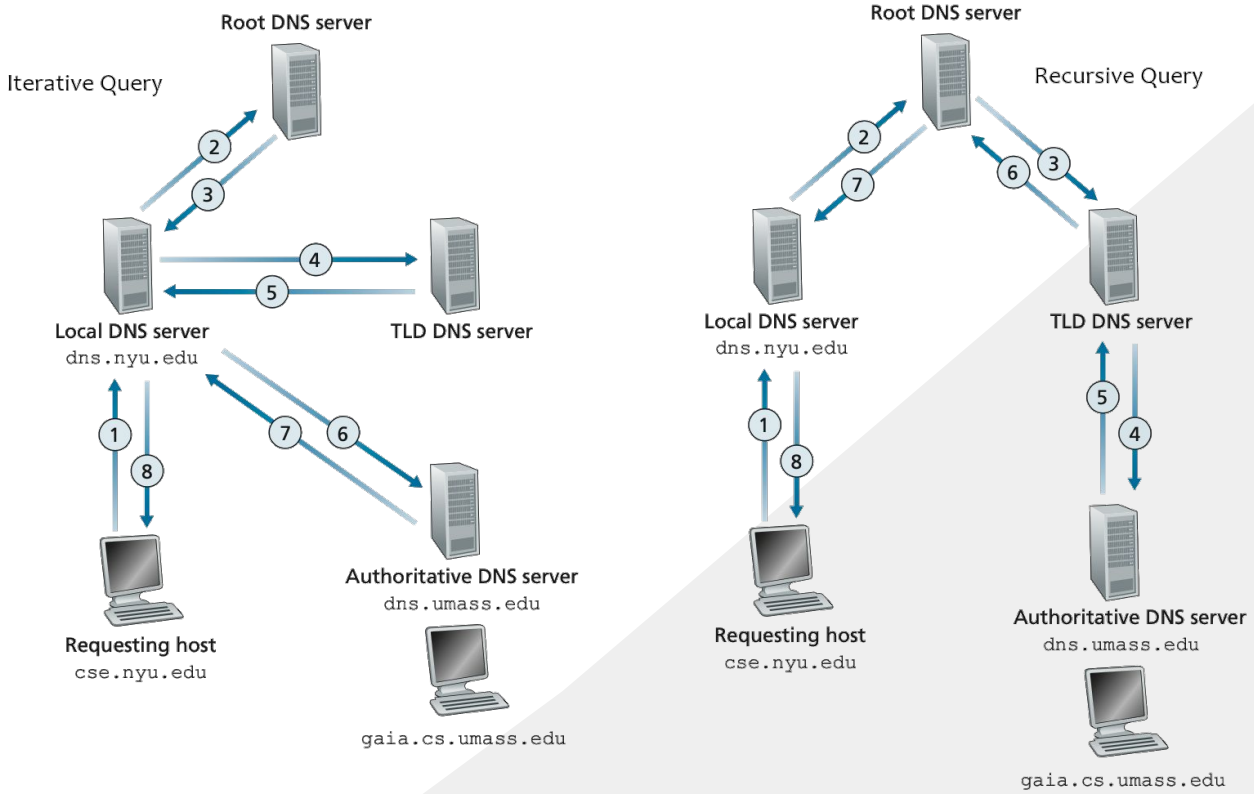
Serveur DNS de premier niveau (TLD)

fournir des informations sur les serveurs DNS autoritaires associés à chaque domaine de premier niveau

Serveur DNS Autoritaire

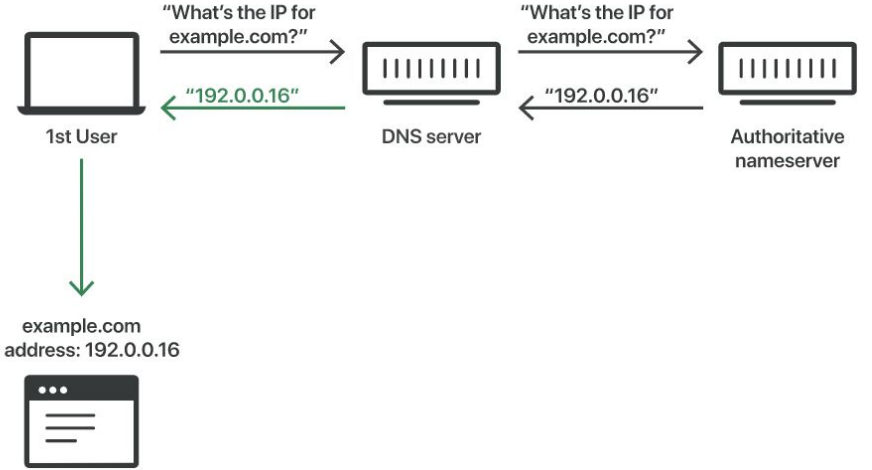
Gérer la gestion des enregistrements DNS pour un domaine spécifique.

Le parcours d'une requête DNS

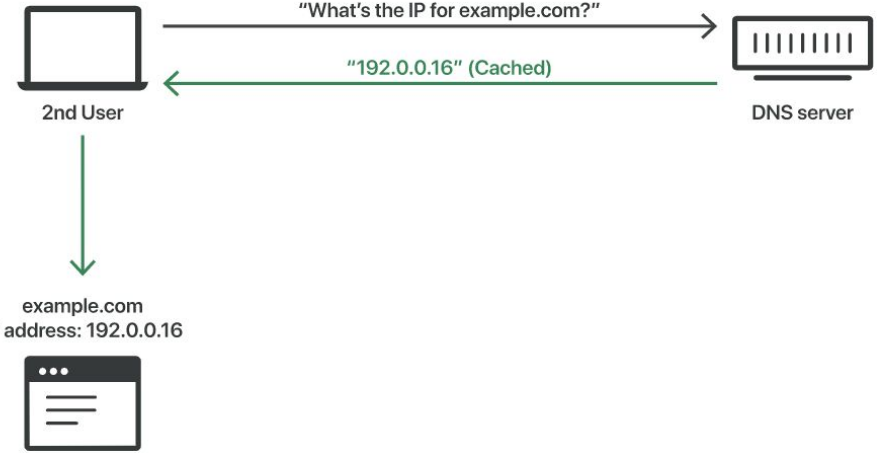


Mise en CACHE DNS

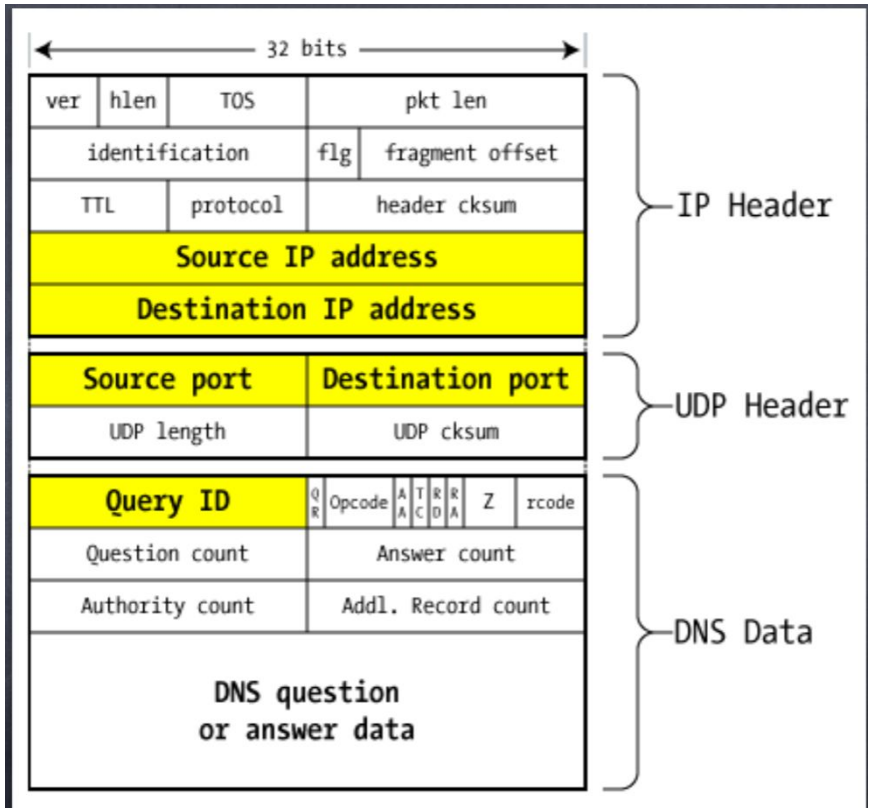
Réponse DNS non mise en cache :



Réponse mise en cache DNS :



Un paquet d'une requête DNS



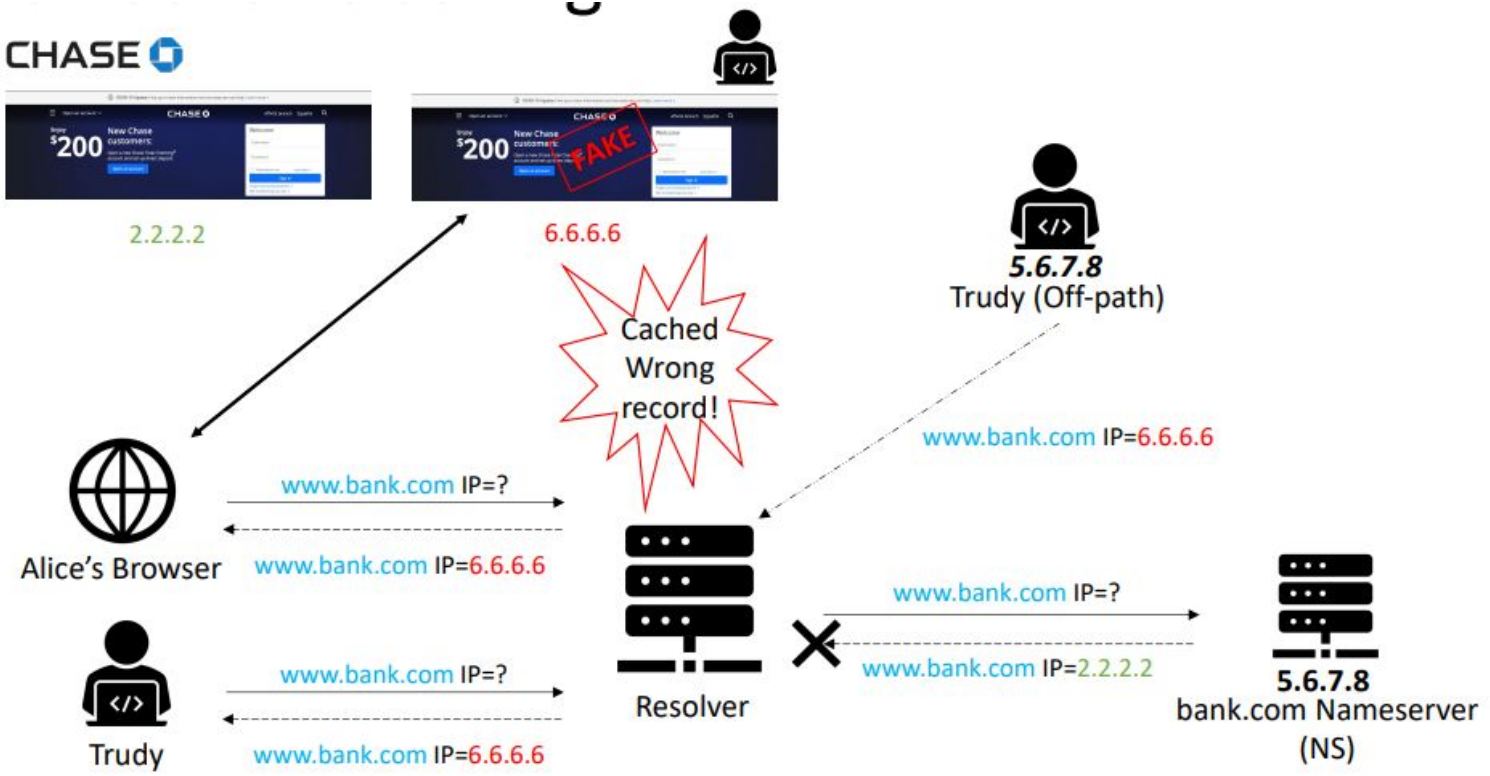
Query ID (ou Transaction ID) : valeur aléatoire de 16 bits choisie par le client.

Lorsqu'il envoie une requête à un serveur DNS, celui-ci enregistre la question et son identifiant.

Destination Port : Les serveurs DNS reçoivent les requêtes sur le port 53/udp.

Sort Port : varie considérablement, parfois, il s'agit également du port 53/udp, parfois d'un port fixe choisi au hasard par le système d'exploitation,

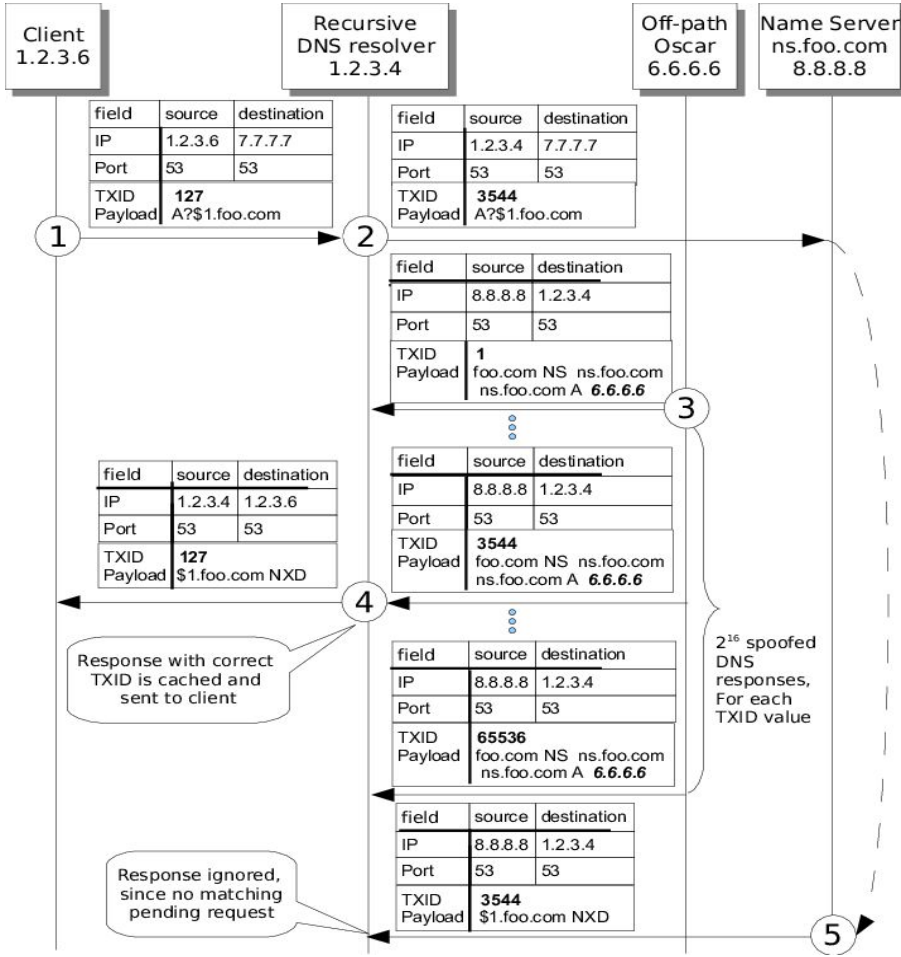
En quoi consiste le CACHE POISONING ?



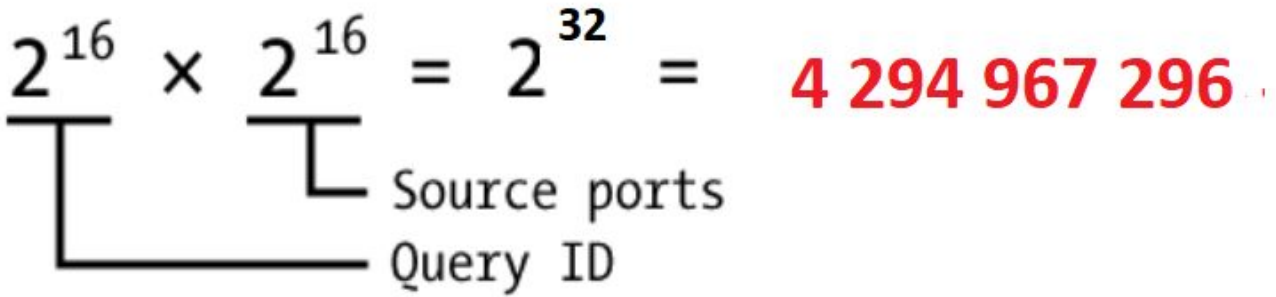
Attaque de Kaminsky

DNS Cache Poisoning

IP Layer	**Src:** 5.6.7.8	
	Dst: *(resolver)*	
UDP Layer	**Src Port:** 53	**Dst Port:**
DNS Layer	**TxID:**	
	Question: www.bank.com A ?	
	Answer: www.bank.com A 6.6.6.6, TTL=**99999**	



Quel est le correctif ?



Randomization du port source

L’attaque de Kaminsky devenue impossible ?

IP Layer	**Src:** 5.6.7.8	
	Dst: *(resolver)*	
UDP Layer	**Src Port:** 53	**Dst Port:** ? (16 bit)
DNS Layer	**TxID:** ? (16 bit)	
	Question: www.bank.com A ?	
	Answer: www.bank.com A 6.6.6.6, TTL=**99999**	

Deux inconnus :

1. Le port source généré par le résolveur de façon aléatoire.
2. La transaction-ID.

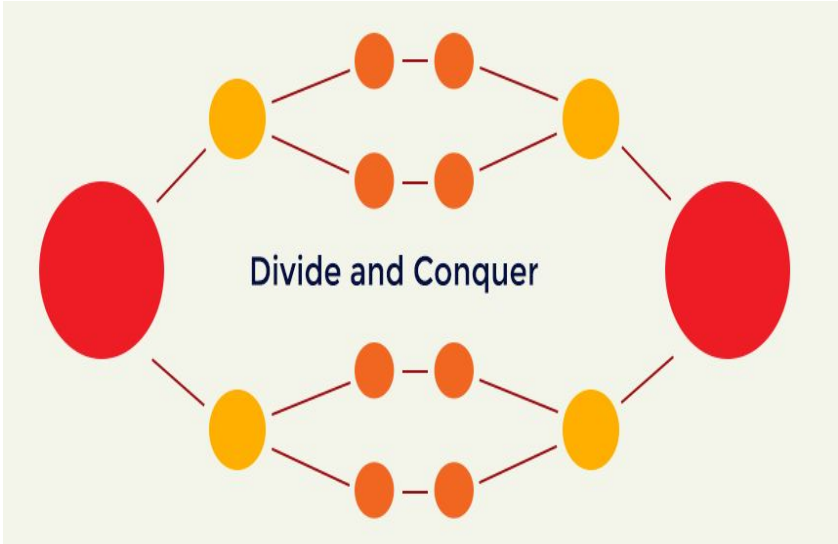
L’univers des possibilités est donc de $2^{16} \times 2^{16} = 2^{32}$.

Diviser pour Régner

Diviser l'espace de recherche :

1. Identification du port source
2. Identification du transaction ID

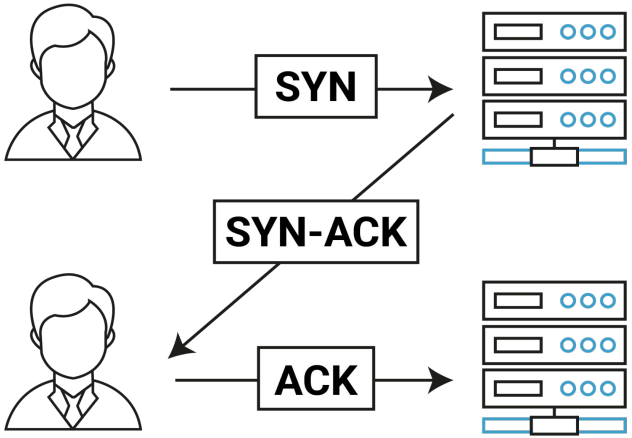
En procédant ainsi, nous passons d'un nombre impracticable, 2^{32} à $2^{16} + 2^{16} \approx 2^{16}$

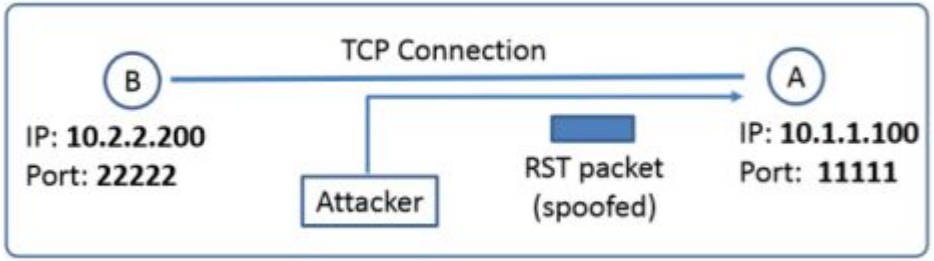


Comment identifier le port source ?

1. Problème sur le protocole TCP et l'ajout du ACK Challenge
2. Exploit TCP permettant de deviner des informations sensibles
3. Le protocole ICMP et le taux limite
4. La stratégie de scan de ports s'inspirant du TCP exploit

Le protocole TCP vulnérable ?

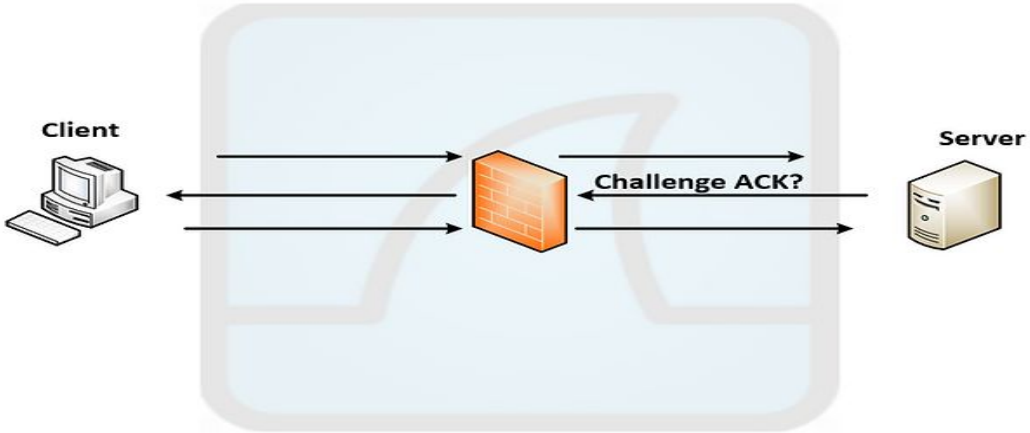


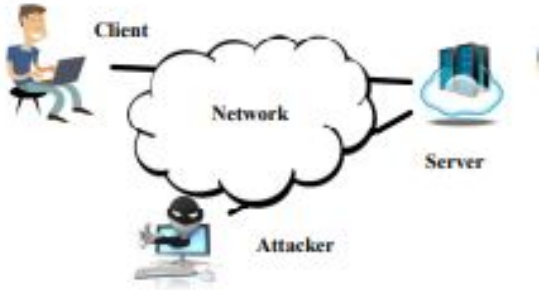


Interruption de connexion TCP entre deux entités par un attaquant à l'aide de l'injection d'un paquet SYN ou RST.

L'attaquant connaît un quadruplé d'une connection active
L'attaquant n'a pas accès aux informations entre A et B

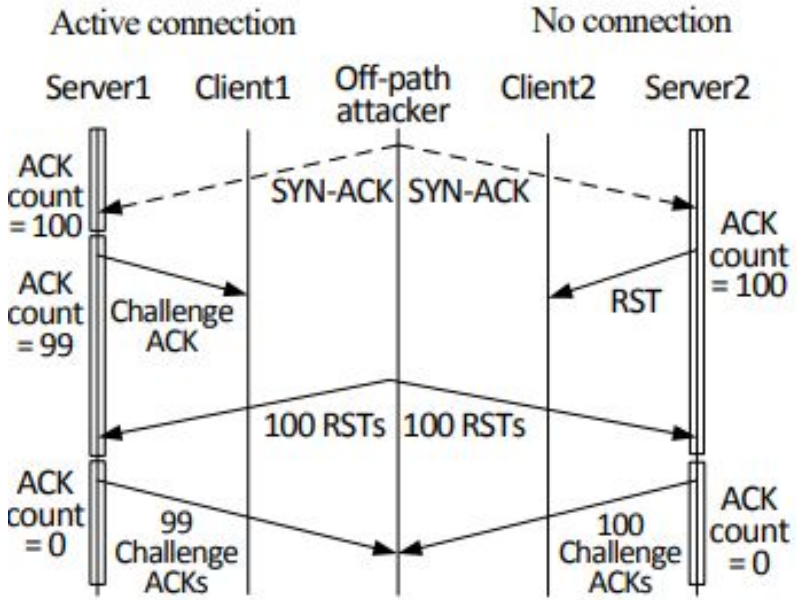
ACK Challenge





1. Le serveur envoie un paquet appelé challenge ACK suite à un RST ou SYN.
2. Confirmer que la requête d'interruption de la connexion en cours était légitime.
3. Ajout d'un taux limite contrôlant le nombre de challenge ACK généré par seconde.
4. Numéro ACK généré aléatoirement.

Exploitation du taux limite d' ACK Challenge

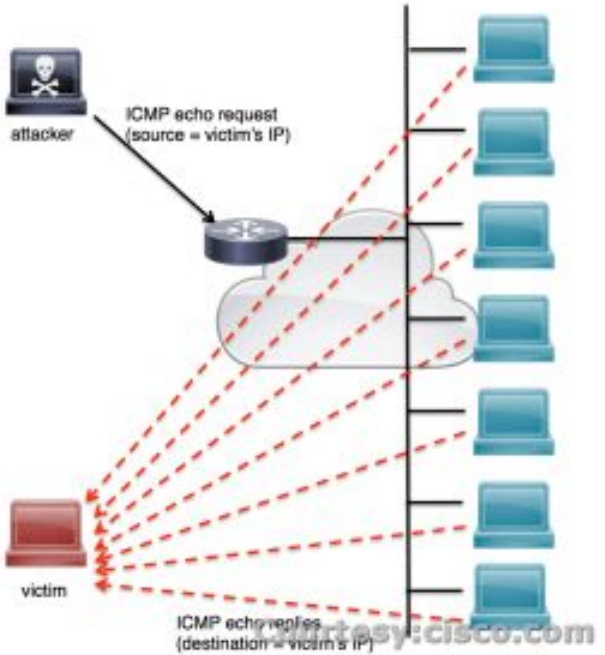


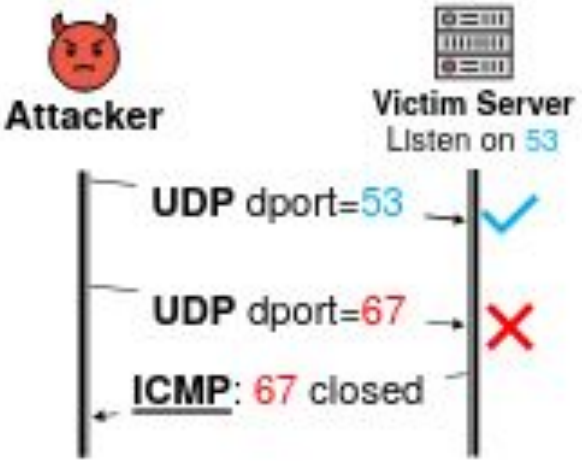
Le taux est par défaut de 100 par seconde sous Linux.

Utilisé pour :

- **savoir si un quadruplé correspond a une connection active**
- **deviner le numéro de séquence ou d'ACK.**

ICMP comme vecteur d'attaque





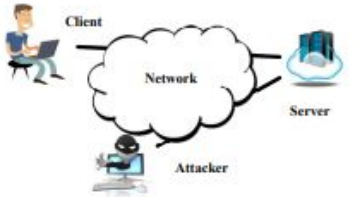
- **Possibilité de faire du déni de service en usurpant l'adresse IP de la victime.**
- **Un serveur peut aussi passer son temps à envoyer des messages ICMP par exemple si on le soumet à un grand nombre de paquets sur un port fermé. (Port Unreachable)**

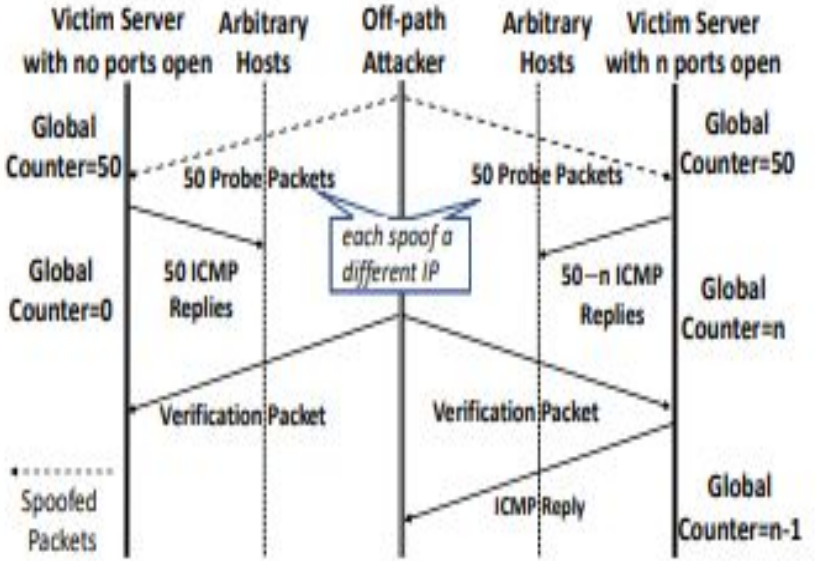
ICMP rate limit

- Ajout d’un compteur global du nombre de messages ICMP pouvant être générés (par défaut sous Linux 50 par 50ms).
- Ajout d’un compteur aussi propre à chaque IP (1 par seconde par défaut).

Si au moins un des deux compteurs n’est pas satisfait, la réponse ICMP ne sera pas générée.

Scan rapide de ports

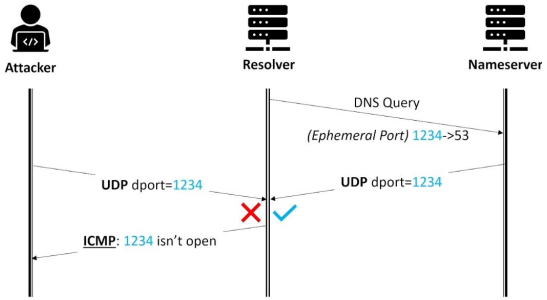




- Utilisation d'adresses spoofées
- Rafale de 50 paquets sur 50 ports différents
- Paquet de vérification

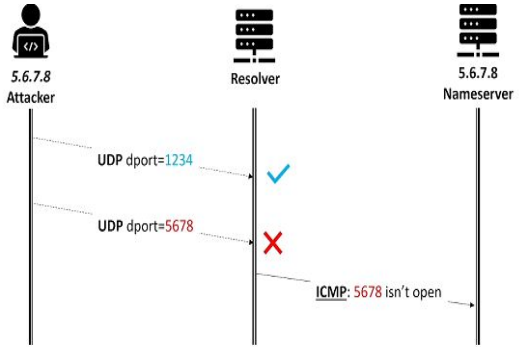
Vérification de 1000 ports par seconde soit environ 60 secondes pour tester les 2^{16} possibilités.

Scan rapide de ports même privés



Utilisation de connect() pour rendre le port privé entre deux entités.

L'algorithme de Linux vérifie d'abord le taux globale avant celui par IP.

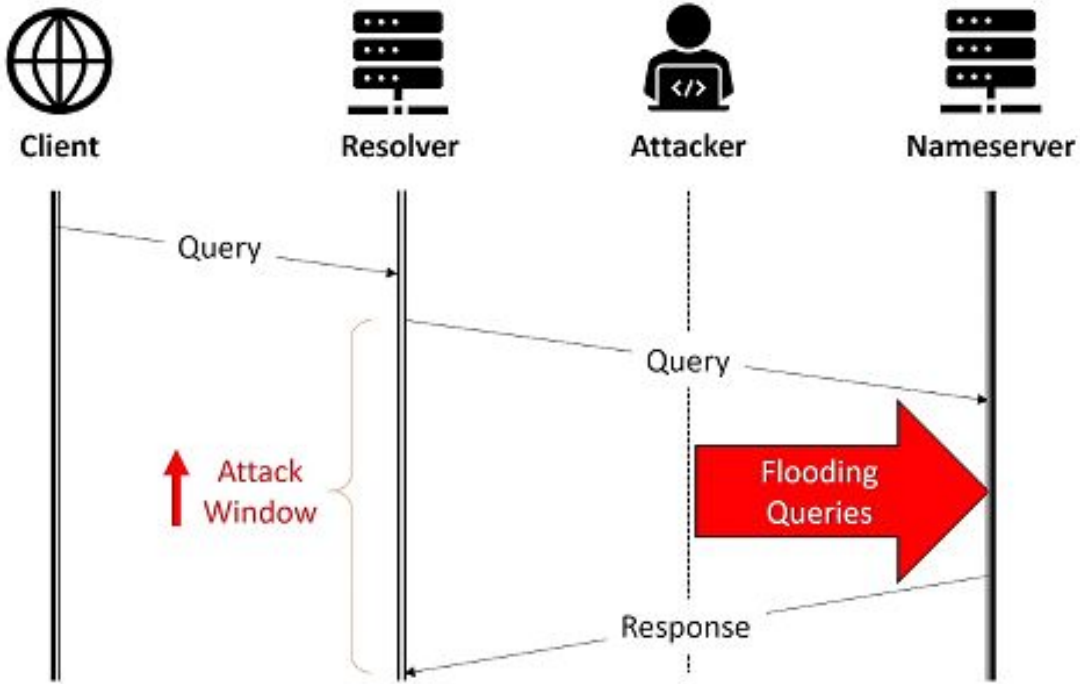


Possibilité d'usurper l'adresse IP du serveur de nom.

L’attaque de Kaminsky redevient faisable

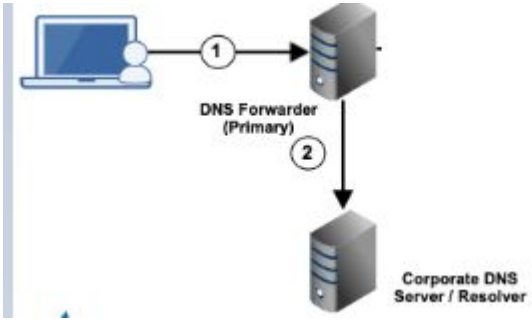
1. L’attaquant demande la résolution d’un nom de domaine victime.
2. L’attaquant utilise le scan de port pour identifier le port source.
3. On se retrouve dans le même contexte que l’attaque de Kaminsky.
4. Réponse avant le serveur, l’enregistrement falsifié est mis en cache.

Amélioration de l'attaque (1)



Augmenter la fenêtre d'attaque

Amélioration de l'attaque (2)



Answer	www.attacker.com	CNAME	www.victim.com
	www.victim.com	A	1.2.3.4

Attaque sur le serveur de redirection

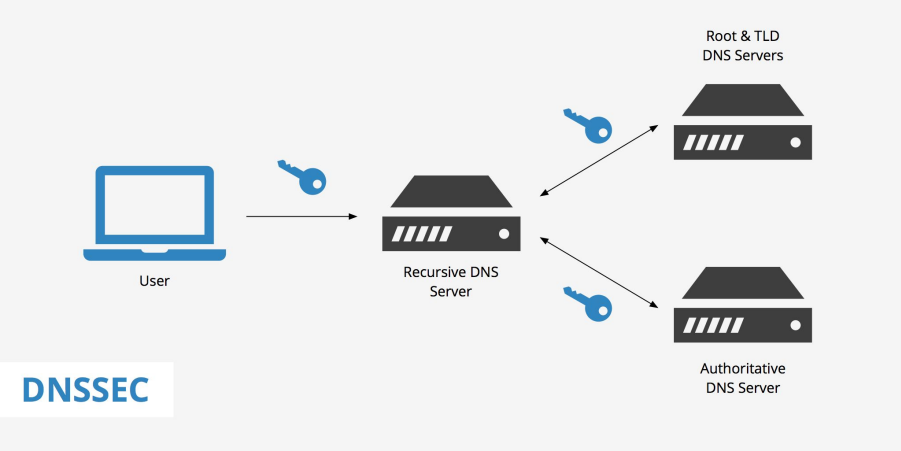
Field	Value
Question	{nonce}.www.victim.com
Answer	
Authoritative	www.victim.com NS ns.attacker.com
Additional	

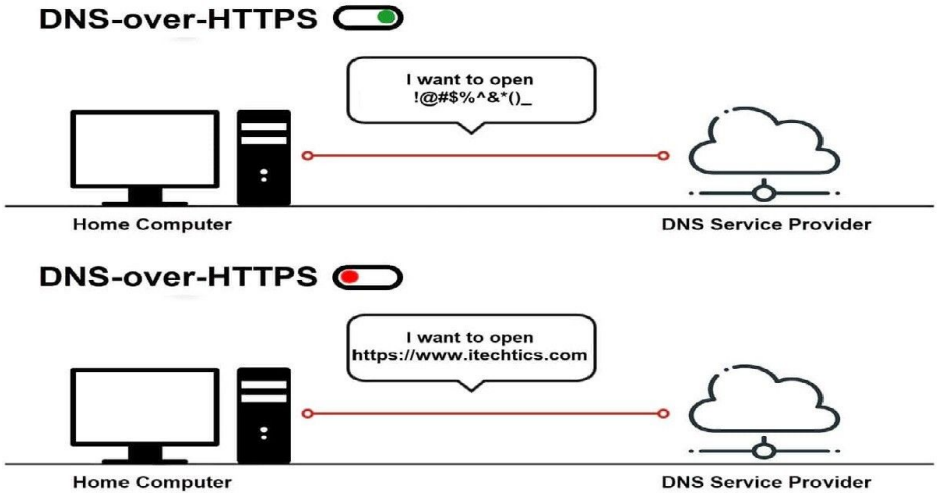
Contournement du TTL du cache

Les contre-mesures (1)

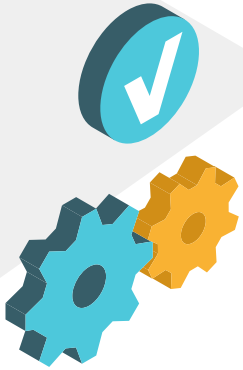
1. Ne plus émettre de paquet Port Unreachable
2. Randomiser le taux limite ICMP

Les contre-mesures (2)





CONCLUSION



MERCI DE VOTRE ATTENTION