Attaques et sécurisation des couches OSI
DoS : Dénis de Service

Philippe Owezarski
LAAS-CNRS, Toulouse, France
owe@laas.fr

DoS : c'est quoi ?

Objectif

Réduire la disponibilité normale d'un service pour quelqu'un de précis ou de façon générale

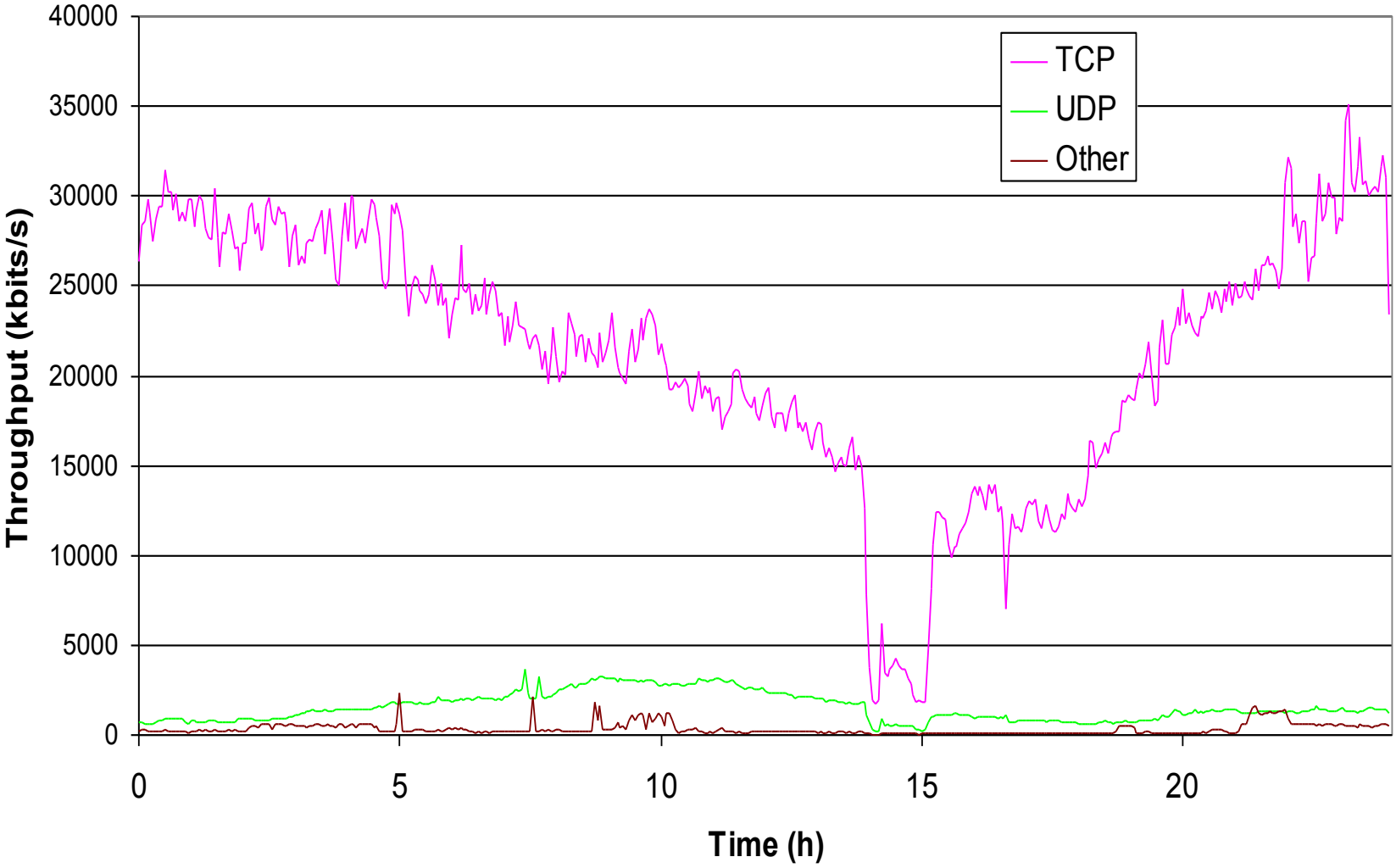
Comment ?

De nombreuses approches sont possibles, on distingue généralement :

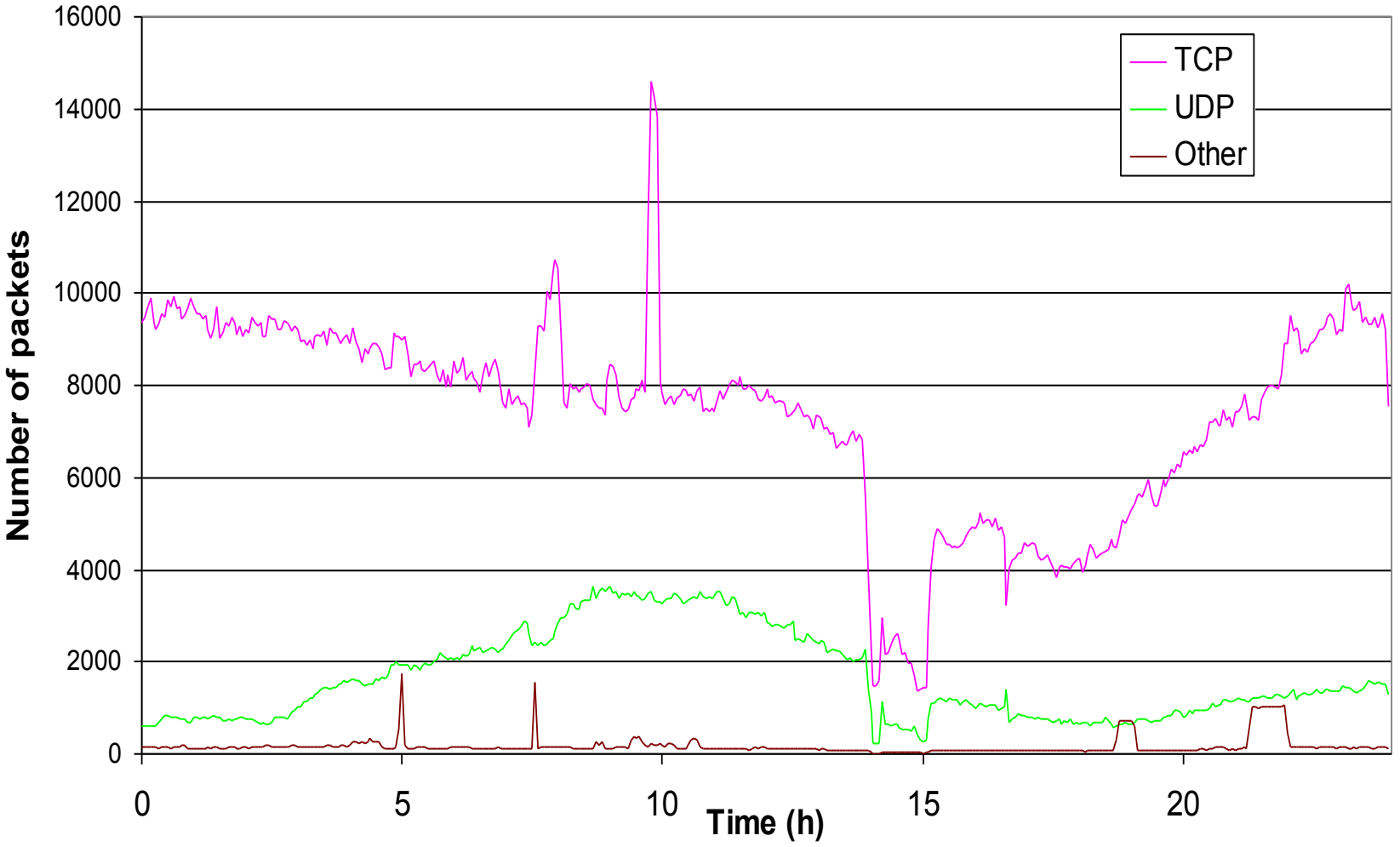
- Le déni provoqué à partir d'un unique nœud
- Le déni provoqué à partir d'un grand ensemble de nœuds (déni de service distribué ou DDoS)

Dans le cas d'un DoS provoqué par un nœud unique, on distingue également si c'est par exploitation d'un bug, ou par inondation de requêtes via un mécanisme d'amplification (flood)

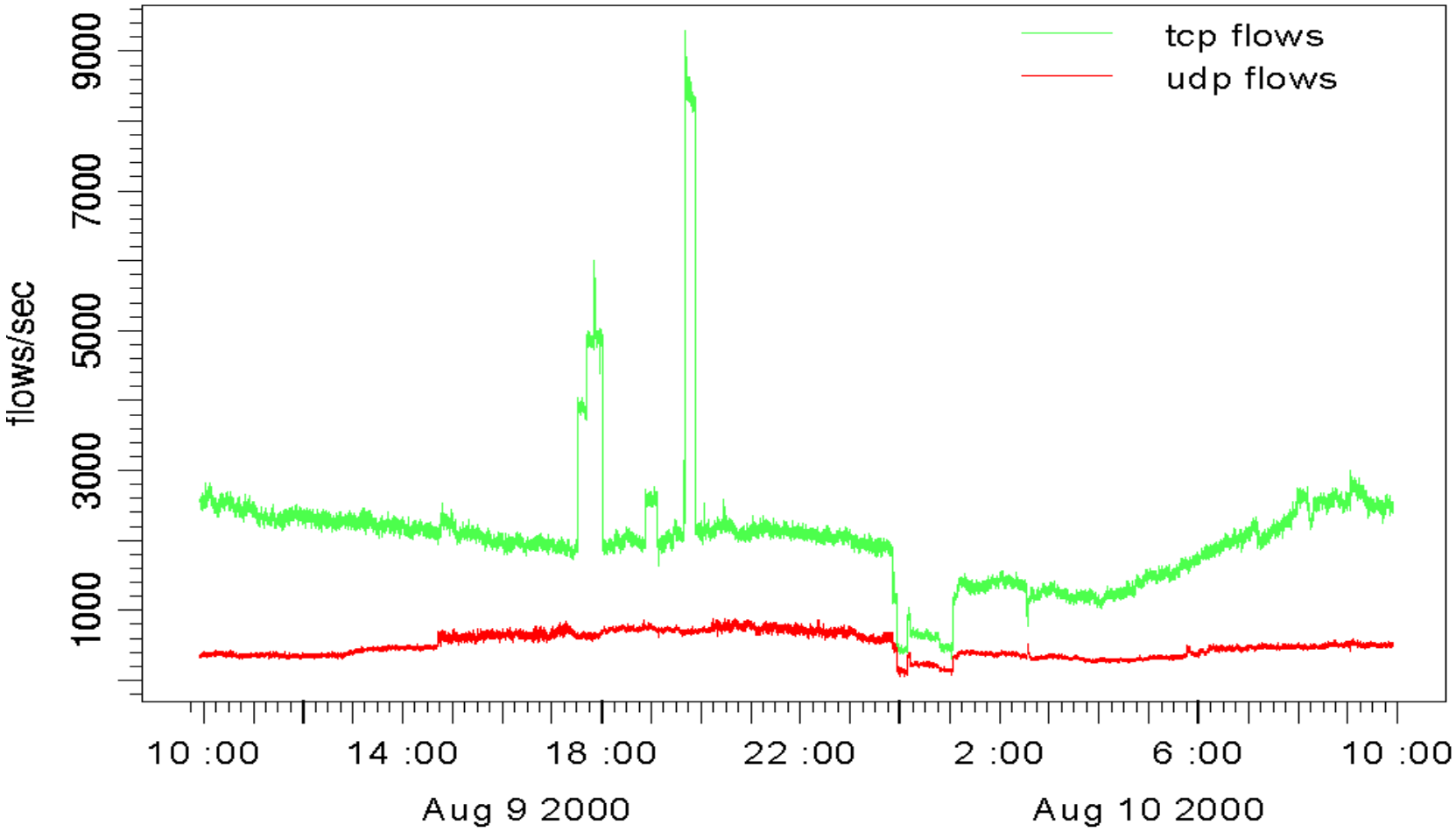
Exemple : Débit sur un lien



Exemple : Débit paquets



Exemple : nombre de flux instantanés



Comment marchent les DoS

- En exploitant des vulnérabilités de sécurité et des faiblesses dans les protocoles et architectures
- En essayant de rendre la performance du réseau plus lente ou aussi inefficace que possible
- En éliminant des nœuds ou en altérant la topologie du réseau
- En dégradant sensiblement la performance des serveurs ou des pairs distants
- En engloutissant la victime sous des calculs fastidieux ou en la poussant à rassembler des fragments de code de sorte qu'elle soit trop occupée pour répondre aux autres requêtes.

Principe (au début du cours)

Attaques par couches

Un DoS peut avoir pour cible une application donnée ou la pile IP d'un ordinateur → Toutes les couches sont concernées

Contre-mesures par couche

Peu à dire sur les contre-mesures génériques (IDS, Firewalls)

Internet est très mal conçu pour contrer les dénis de service

Attaques locales

Principe

Diffuser en continu au maximum de la bande passante, renforcé par :

- L'envoi de messages en broadcast (e.g. des requêtes ARP)
- La saturation des tables CAM (commutateurs→ponts)

Contre-mesure générique

Le réseau risque de se voir bloqué donc il serait difficile de détecter l'origine de l'attaque. Principe de dichotomie (on coupe un bout du réseau et on observe si le problème persiste

Prévention

Configurer les commutateurs pour limiter le trafic broadcast

Pour une attaque en unicast, revenir à une solution générique

Attaques distantes

Différences par rapport au cas local

Plus facile à filtrer si la source est unique, et le broadcast est alors impossible

→ Recherche de phénomènes d'amplification

DDoS

Cas classique : l'attaquant contrôle un botnet (réseau d'ordinateurs corrompus) qui vont agir de concert pour potentialiser l'attaque de déni de service

Très difficile à bloquer sans couper tout le trafic extérieur

→ Ce sont des attaques contrôlées à distance

Attaques sur la couche physique

Dans les communication sans fil

Déauthentification WIFI, brouilleurs de signaux, générateurs de bruits, …

Attaques DHCP

Attaques par famine

Prendre toutes les ressources avant les autres

Limitations / Contre-mesures

Ne bloque pas les machines déjà connectées / ayant une adresse réservée

Similaire : ARP Spoofing

Déni silencieux et polyvalent d'une machine donnée

Attaques ICMP

Ping flooding

Simple déni de service par inondation du lien physique

Version « Smurf » pour amplifier l'attaque

Envoyer un ICMP Echo request à un réseau en remplaçant l'IP source par celle de la victime

→Tous les membres envoient des ICMP Echo Reply à la cible

L'attaque « Fraggle » est identique, mais avec des paquets UDP

Contre-mesures

Filtrage ICMP, en particulier Echo Reply ne répondant pas à un paquet sortant

Smurf Amplifier Registry énumère les réseaux mal configurés pouvant mener à une attaque de type Smurf

Attaques sur IP

Ping of Death (PoD) : une première attaque par fragmentation (1975-1998)

Rien à voir avec ICMP mais classiquement le paquet mal formé était un ping.

Idée : envoyer un ping de plus de 65535 octets pour provoquer l'envoi de nombreux fragments IP de 1500 octets, puis un dernier avec un offset de 65528 = $(2^{13} - 1) * 8$ d'une taille de plus de 7 octets.

→Après reconstruction, le paquet est trop grand (buffer overflow, crash, etc.)

Teardrop

Variante utilisant des fragments se chevauchant (bugs→crashs)

Famine

Certains équipements (firewalls, switches,…) peuvent être saturés en envoyant des paquets fragmentés avec des trous importants entre les données (forte amplification pour la saturation mémoire)

Attaques TCP RST

Principe

Interdire à une machine donnée d'accéder à un service donné

Comment ?

Essayer de deviner un numéro de séquence valide est envoyer un TCP RST

Pourquoi ça marche ?

Naïvement : une chance sur 2^{32}

Il suffit de tomber dans la fenêtre TCP ($1500 < x < 2^{24}$, 65000 par défaut)

Limites

Les connexions sont courtes

Randomisation du port source et du numéro de séquence à l'ouverture de la connexion

TCP SYN flood (1/2)

Version non distribuée

Saturer la file pour les connexions semi ouvertes

La taille limite de cette file peut être très basse

Initialement entre 10-128, aujourd'hui 1024 ou plus

Contre-mesures immédiates

- Augmenter la taille de cette file, réduire le temps d'attente → Gain linéaire pour coût linéaire
- Filtrage IP (contourné par la randomisation de la source)

SYN cookies

- Idée de base : en cas d'attaque ne pas enregistrer les connexions ouvertes avant le ACK final
- Mais comment garder les informations indispensables qu'on enregistre normalement à la réception du SYN (adresses et ports source et destination, MSS) ?
- Principe : utiliser une clé secrète et les données qu'on aurait enregistré pour générer le SN du SYN/ACK

TCP SYN flood (2/2)

SYN Cookies : Comment ça marche ?

- Clé : Clé secrête choisie au hasard au démarrage
- T : Compteur sur 5 bits incrémentés toutes les 64 secondes
- L : MAC_{cle} (Saddr, Sport, Daddr, Dport, SN_c, T) sur 24bits
- SN_s : Concaténation de T, MSS et M
- Que 3 bits pour MSS : 8 valeurs prédéfinies

→Le serveur n'a pas besoin de sauvegarder l'état

→Quand il reçoit un SYN/ACK, il extrait MSS, vérifie L et T et n'alloue de l'espace que si tout marche

TCP Cookie Transactions

- Les SYN Cookies marchent quel que soit le client, mais toutes les options TCP sont perdues
- TCPCT est une extension dédiée évitant ce problème, mais il faut que le client et le serveur l'activent (Linux kernel > 2.6)

DNS amplification

Principe

Envoyer des requêtes DNS UDP à des résolveurs récursifs en remplaçant l'adresse source par celle de la victime

Pourquoi ça marche ?

Estimation 2006 : 580 000 résolveurs récursifs accessibles sur Internet

Requête DNS de 60 octets : Réponse de 512 octets (x8)

EDNS0 et DNSSEC

- En activant DNSSEC on peut avoir des réponses UDP de plus de 512 octets
- Attention : Au delà d'une certaine taille, on passe dur du TCP

Windows 8 : 4000 octets (x60)!!!

Attaques génériques

Coûts déséquilibrés sur le calcul

Connexion TLS (client hello, serveur hello, client enc(key))

Le client chiffre, le serveur déchiffre (x10)

→ Le coût de calcul peut aussi être utilisé pour protéger un réseau, par exemple si la demande d'ouverture d'une connexion nécessite de résoudre un puzzle. Le coût de l'attaque deviendra tel, qu'elle ne pourra jamais être massive

Coûts déséquilibrés sur la bande passante

Demander un gros fichier PDF… ou des réponses DNS en UDP…

Exploitation de bugs

Pour faire tomber un serveur, par ex.

Attaques sur la topologie du réseau

Attaque par « éclipse »

Partitionner le réseau pour interdire aux membres des différents sous réseaux de communiquer entre eux

Comment ?

- Prendre le contrôle de nœuds du réseau sur des chemins (path) stratégiques
- Les nœuds corrompus bloquent les paquets qui leur arrivent

Attaques BGP

Objectif

- Saturer les routeurs (cf. précédemment) – cela peut aller jusqu'à éclipser un AS
- Empoisonnement des tables de routage

Comment ?

- Flooding d'annonces BGP
- Envoi de flux de paquets ayant des caractéristiques temporelles précises

- Le plus souvent, le DoS est provoqué par des erreurs de configuration (ex. Pakistan Telecom et trafic YouTube)

→ Sinkhole (à la fois attaque et système de défense)

Contre-mesures

S-BGP, BGPSEC (exploitent des solutions de crypto)

That’s all folks !