Quelques exemples pratiques de l'utilisation de la commandes NMAP

Nmap est un scanner de ports libre.
Il est conçu pour détecter les ports ouverts, identifier les services hébergés et obtenir des informations sur le système d'exploitation d'un ordinateur distant.
Ce logiciel est devenu une référence pour les administrateurs réseaux, car l'audit des résultats de Nmap fournit des indications sur la sécurité d'un réseau.
Il est disponible sous Windows, Mac OS X, Linux, BSD et Solaris.

Le code source de Nmap est disponible sous la licence GNU GPL.

Dans cet article, nous allons couvrir quelques exemples pratiques de la commande NMAP sous Linux.
Les principales utilisations de nmap sont:

- Déterminer les ports et les services ouverts en cours d'exécution dans un hôte
- Déterminer le système d'exploitation exécuté sur un hôte
- Usurpé l'adresse IP source (consistera à utiliser l'option -S)

Installation de NMAP

Sur Debian/Ubuntu :

```
aptitude install nmap
```

Sur CentOS/RedHat :

```
yum install nmap
```

découverte d'hôtes sur un sous réseau

Cette commande est généralement un simple `scan ping`.
Parfois vous voulez juste savoir quels sont les hôtes actifs d'un réseau.
Nmap peut le faire pour vous en envoyant des paquets d'écho ICMP à chaque adresse IP du réseau spécifié.
Les hôtes qui répondent sont actifs.
Malheureusement, certains sites bloquent les paquets d'écho.
Toutefois nmap peut aussi envoyer un paquet TCP ACK au port 80 (par défaut).
Si vous recevez un RST en retour, la machine est active.
Une troisième technique consiste à envoyer un paquet SYN et d'attendre un RST ou un SYN/ACK.

N'utilisez cette option que si vous voulez faire un balayage de ping sans faire d'analyse de ports.

```
nmap -sP 192.168.1.0/24
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 20:52 CEST
Nmap scan report for 192.168.1.14
```

```
Host is up (0.0036s latency).
MAC Address:
Nmap scan report for 192.168.1.254
Host is up (0.00031s latency).
MAC Address:
Nmap scan report for 192.168.1.250
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 7.76 seconds
```

Scanner les ports ouverts

Cette commande est l'utilisation par défaut de NMAP.
Il faudra du temps pour que NMAP vous donne la réponse, il va tenter une connexion TCP SYN sur 1000 ports, les plus communs ainsi qu'une demande d'écho ICMP pour déterminer si un hôte est actif. nmap va également effectuer une recherche DNS inversée sur les IP identifiées, cela peut parfois être des informations utiles.

```
nmap 192.168.1.0/24
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 20:53 CEST
Nmap scan report for 192.168.1.14
Host is up (0.0060s latency).
All 1000 scanned ports on 192.168.1.14 are filtered
MAC Address:

Nmap scan report for 192.168.1.254
Host is up (0.00046s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
139/tcp  open  netbios-ssn
631/tcp  open  ipp
8200/tcp open  trivnet1
MAC Address:

Nmap scan report for 192.168.1.250
Host is up (0.000012s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind
443/tcp open  https

Nmap done: 256 IP addresses (3 hosts up) scanned in 205.61 seconds
```

Identifier le système d’exploitation d'un hôte

Pour identifier le système d'exploitation d'un hôte à l'aide de nmap, vous pouvez le faire avec l'option -O.
Mais l'analyse du système d'exploitation exige les privilèges root.

```
nmap -O 192.168.1.254
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 21:01 CEST
Nmap scan report for 192.168.1.254
Host is up (0.00047s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
139/tcp  open  netbios-ssn
631/tcp  open  ipp
8200/tcp open  trivnet1
MAC Address:
Device type: switch|media device|general purpose
Running: HP embedded, Philips embedded, Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: HP Brocade 4Gb SAN switch or, Linux 2.6.15 - 2.6.26 (likely
embedded)
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 189.64 seconds
```

Identifier les noms d'hôtes

NMAP vous permet de trouver des noms d'hôte pour tous les IP dans un sous-réseau sans avoir envoyer un paquet à l'individu.

L'option -SL de NMAP permet de faire une requête DNS simple pour l'IP spécifiée.
Cette analyse ne nécessite pas les privilèges root.

```
nmap -sL 172.16.0.0/24
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 21:01 CEST
Nmap scan report for 172.16.0.0
Nmap scan report for router.local (172.16.0.1)
Nmap scan report for myhost.local (172.16.0.2)
Nmap scan report for another.myhost.local (172.16.0.3)
```

Scanne TCP et UDP port

Cette commande nmap -sS -sU -PN vérifiera environ 2000 ports TCP et UDP commun pour voir s'ils répondent.
Lorsque vous utilisez l'option -Pn, NMAP va considérer tous les hôtes comme étant connectés, saute l'étape de découverte des hôtes.

Cela peut être utile pour vérifier si un pare-feu empêche réponses ICMP.
Cette analyse nécessite un des privilèges root.

```
nmap -sS -sU -PN 192.168.1.100
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 21:31 CEST
Nmap scan report for 192.168.1.100
Host is up (0.00029s latency).
Not shown: 1494 closed ports, 496 filtered ports
PORT STATE SERVICE
88/tcp open kerberos-sec
139/tcp open netbios-ssn
445/tcp open microsoft-ds
631/tcp open ipp
88/udp open|filtered kerberos-sec
123/udp open ntp
137/udp open netbios-ns
138/udp open|filtered netbios-dgm
631/udp open|filtered ipp
5353/udp open zeroconf
```

Scanne tous les ports TCP et UDP

Cette commande nécessite les privilèges root et il est le même que ci-dessus mais en spécifiant la plage complète des ports de 1 à 65535 NMAP va scanner pour voir si l'hôte est à l'écoute sur tous les ports disponibles.

```
nmap -sS -sU -PN -p 1-65535 192.168.1.250
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 22:02 CEST
Nmap scan report for 192.168.1.250
Host is up (0.00029s latency).
Not shown: 131052 closed ports
PORT STATE SERVICE
88/tcp open kerberos-sec
139/tcp open netbios-ssn
445/tcp open microsoft-ds
631/tcp open ipp
17500/tcp open unknown
88/udp open|filtered kerberos-sec
123/udp open ntp
137/udp open netbios-ns
138/udp open|filtered netbios-dgm
631/udp open|filtered ipp
5353/udp open zeroconf
17500/udp open|filtered unknown
51657/udp open|filtered unknown
54658/udp open|filtered unknown
56128/udp open|filtered unknown
57798/udp open|filtered unknown
58488/udp open|filtered unknown
```

```
60027/udp open|filtered unknown
```

Scanné les connexions TCP

Cette commande va demander à l'OS d'établir une connexion TCP vers les 1000 ports communs.

```
nmap -sT 192.168.1.110
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 22:46 CEST
Nmap scan report for 192.168.1.110
Host is up (0.0014s latency).
Not shown: 964 closed ports, 32 filtered ports
PORT STATE SERVICE
88/tcp open kerberos-sec
139/tcp open netbios-ssn
445/tcp open microsoft-ds
631/tcp open ipp
```

Scanne rapide

Vous pouvez utiliser ce scanne pour vérifier les 100 ports les plus courants.

```
nmap -T4 -F 192.168.1.99
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 23:00 CEST
Nmap scan report for 192.168.1.99
Host is up (0.00047s latency).
Not shown: 96 closed ports
PORT STATE SERVICE
88/tcp open kerberos-sec
139/tcp open netbios-ssn
445/tcp open microsoft-ds
631/tcp open ipp
```

Scanne agressif et envahissant de numérisation

Pas comme les commandes antérieures cette analyse est très agressif et très envahissant.
L'option -A va effectuer une vérification du système d'exploitation et une vérification de version ainsi que l'utilisation de traceroute.
Le -T4 est pour le modèle de vitesse ainsi que le nom de la machine cible.

```
nmap -A -T4 192.168.1.0/24
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 23:40 CEST
Nmap scan report for 192.168.1.250
Host is up (0.000010s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 6.7p1 Debian 5 (protocol 2.0)
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
```

```
80/tcp  open  http     Apache httpd
|_http-title: Apache2 Debian Default Page: It works
111/tcp open  rpcbind  2-4 (RPC #100000)
| rpcinfo:
|   program version   port/proto  service
|   100000  2,3,4        111/tcp  rpcbind
|   100000  2,3,4        111/udp  rpcbind
|   100024  1          41789/udp  status
|_  100024  1          59267/tcp  status
443/tcp open  ssl/http Apache httpd
|_http-title: Apache2 Debian Default Page: It works
| ssl-cert: Subject: commonName=note-home
| Not valid before: 2015-04-22T18:45:08+00:00
|_Not valid after:  2025-04-19T18:45:08+00:00
|_ssl-date: 1987-01-04T01:09:35+00:00; -28y174d20h34m03s from local time.
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.7 - 3.15
Network Distance: 0 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Verbeux

Quand vous ajoutez `verbose` à la ligne de commande, vous obtiendrez une meilleure information.

```
nmap -A -T4 -v 192.168.1.250
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-27 23:51 CEST
NSE: Loaded 118 scripts for scanning.
NSE: Script Pre-scanning.
Initiating Parallel DNS resolution of 1 host. at 23:51
Completed Parallel DNS resolution of 1 host. at 23:51, 0.00s elapsed
Initiating SYN Stealth Scan at 23:51
Scanning 192.168.1.250 [1000 ports]
Discovered open port 80/tcp on 192.168.1.250
Discovered open port 111/tcp on 192.168.1.250
Discovered open port 443/tcp on 192.168.1.250
Discovered open port 22/tcp on 192.168.1.250
Completed SYN Stealth Scan at 23:51, 2.02s elapsed (1000 total ports)
Initiating Service scan at 23:51
Scanning 4 services on 192.168.1.250
Completed Service scan at 23:51, 12.15s elapsed (4 services on 1 host)
Initiating OS detection (try #1) against 192.168.1.250
NSE: Script scanning 192.168.1.250.
Initiating NSE at 23:51
NSOCK ERROR [17.3740s] mksock_bind_addr(): Bind to 0.0.0.0:53 failed (IOD
#19): Address already in use (98)
Completed NSE at 23:51, 1.57s elapsed
Nmap scan report for 192.168.1.250
Host is up (0.000014s latency).
```

```
Not shown: 996 closed ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 6.7p1 Debian 5 (protocol 2.0)
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
80/tcp  open  http     Apache httpd
|_http-methods: GET HEAD POST OPTIONS
|_http-title: Apache2 Debian Default Page: It works
111/tcp open  rpcbind  2-4 (RPC #100000)
| rpcinfo:
|   program version   port/proto  service
|   100000  2,3,4        111/tcp  rpcbind
|   100000  2,3,4        111/udp  rpcbind
|   100024  1          41789/udp  status
|_  100024  1          59267/tcp  status
443/tcp open  ssl/http Apache httpd
|_http-methods: GET HEAD POST OPTIONS
|_http-title: Apache2 Debian Default Page: It works
| ssl-cert: Subject: commonName=note-home
| Issuer: commonName=note-home
| Public Key type: rsa
| Public Key bits: 2048
| Not valid before: 2015-04-22T18:45:08+00:00
| Not valid after:  2025-04-19T18:45:08+00:00
| MD5:   f091 c3ca 70dd 72ea be91 f2ea 39ca 59cc
|_SHA-1: c21d 4ebf 85f7 e610 fbee c75c 9093 1b03 6aa3 52d4
|_ssl-date: ERROR: Script execution failed (use -d to debug)
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.7 - 3.15
Uptime guess: 0.365 days (since Sat Jun 27 15:05:26 2015)
Network Distance: 0 hops
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.79 seconds
           Raw packets sent: 1078 (50.184KB) | Rcvd: 2170 (95.364KB)
```

Usurpation d’adresse IP source

Pour faire cela, nous allons utiliser l'option `-S` suivit `<spoofed_IP_addr>`:

```
nmap -e eth0 -Pn -S 192.9.204.1 192.9.204.111 -p80
Nmap scan report for toto.test.loc (192.9.204.111)
Host is up (0.00046s latency).
```

```
PORT   STATE SERVICE
80/tcp open  http
MAC Address:

Nmap done: 1 IP address (1 host up) scanned in 0.63 seconds
```

Usurpation d'addresse MAC

Pour faire cela, nous utiliserons l'option –`spoof-mac`

```
nmap --spoof-mac 01:02:03:04:05:06 127.0.0.1
nmap --spoof-mac Cisco 127.0.0.1
Starting Nmap 6.47 ( http://nmap.org ) at 2015-06-29 09:08 CEST
Spoofing MAC address 00:00:0C:F7:EB:11 (Cisco Systems)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000090s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
111/tcp  open  rpcbind
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 2.52 seconds
```

Choisir un fichier de sortie pour y écrire les résultats d'un scanne

```
nmap -oN resultat 127.0.0.1
nmap -oX resultat.xml 127.0.0.1
```

Trace les paquets et les données envoyés et reçus.

Pratique pour vérifier qu'une usurpation fonctionne :

```
nmap --packet-trace -S 10.0.0.0 -eth0 127.0.0.1
```

Option intéressante

Presque toutes les options de scanne disponibles dans la commande NMAP on été vu ensemble.
Mais quelques options non pas été traité :

- **–ttl <val>** : Ce paramètre permet de modifier la valeur TTL (Time To Live, en français durée de vie) des paquets de test envoyés par Nmap.
- **–mtu <val>** : L'option –mtu permet de spécifier la taille du fragment offset. Cette option est très souvent associé avec **-f** pour forcer la fragmentation des paquets
- **-D <decoy1, [,decoy2][,ME] …>** : Cette option permet d'obscurci le scanne avec des leurre,

en dissimulant notre scanne parmi des scannes fictifs d'hôtes dans la liste decoy1, decoy2, etc.

Effectivement, des paquets simulant un scan de 5 à 10 ports sont envoyés vers l'hôte scanné afin de noyer notre IP.
Exemple :

```
nmap -n -e eth0 172.10.1.1 -D 172.10.1.12,172.10.1.13 -p 80
```

Par ailleurs, l'option NE permet de positionner sa véritable adresse dans la liste des decoys. Si NE est positionné en sixième position ou au delà, la probabilité pour qu'un IDS ne repère pas notre IP est plus élevée.

- **-g/–source-port <srcport>** : Utilise le numéro de port comme source, certains par-feu mal paramétrés peuvent baser leur confiance sur le port source des paquets.
  Ainsi par exemple, si un par-feu ne laisse écouter le port 21 qu'en provenance du port 20, il est possible de modifier le scanne en utilisant le port source 20.

Exemple :

```
nmap -sS -g 20 -p 21 172.10.1.18
```

Si vous souhaitez inclure une option dans cette liste, faite le savoir.

sources: nmap.org, delafond.org

From:
https://www.ksh-linux.info/ - **Know Sharing**

Permanent link:
https://www.ksh-linux.info/systeme/quelques-exemples-pratiques-d-utilisation-de-la-commande-nmap

Last update: **12/11/2016 20:12**