Ah, il y a une différence entre nom de domaine et nom de machine ?

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 24 novembre 2015

https://www.bortzmeyer.org/host-vs-domain.html

Un cas intéressant cité par Phil Mayers sur la liste des utilisateurs de BIND va me permettre, cher lecteur, de t'instruire sur la différence entre **nom de domaine** et **nom de machine**.

La question originale était de savoir pourquoi on pouvait faire une résolution DNS (par exemple avec dig) de `nexistesurementpas.um.outlook.com` alors que les commandes comme ping ne pouvaient pas utiliser ce nom :

```
% dig +short nexistesurementpas.um.outlook.com
*.um.outlook.com.glbdns2.microsoft.com.
wildcard-emeasouth.um.outlook.com.
157.55.9.252
```

On récupère bien une adresse IP (c'est pareil avec d'autres outils DNS comme `host`) mais :

```
% ping nexistesurementpas.um.outlook.com
ping: unknown host nexistesurementpas.um.outlook.com
```

Mais, alors, pourquoi est-ce que ping prétend que ce nom n'existe pas ? (Le problème n'est pas spécifique à ping, d'autres commandes comme telnet font le même diagnostic.)

L'explication est qu'il existe une différence entre les noms de domaine ("*domain names*") et les noms de machines ("*host names*"). Les premiers permettent à peu près tous les caractères possibles (cf. RFC 2181 [1], section 11, et regardez le nom `&-funny-%-syntax-$.bortzmeyer.org` pour s'en convaincre). Les seconds obéissent à une syntaxe bien plus restrictive, documentée dans le RFC 1123, section 2.1. En gros, un nom de machine est restreint à LDH ("*Letters, Digits and Hyphen*"). C'est pour cela que je peux résoudre le nom rigolo indiqué plus haut :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc2181.txt`

```
% dig +short +nodnssec '&-funny-%-syntax-$.bortzmeyer.org'
www.bortzmeyer.org.
204.62.14.153
```

Mais que je ne peux pas l'utiliser :

```
% ping '&-funny-%-syntax-$.bortzmeyer.org'
ping: unknown host &-funny-%-syntax-$.bortzmeyer.org
```

(Notez qu'il a fallu l'encadrer d'apostrophes car certains caractères sont spéciaux pour le shell Unix.)

Arrivé à ce stade, mes lecteurs et lectrices, qui sont très malins, se grattent la tête « OK dans l'exemple avec `&-funny-%-syntax-$.bortzmeyer.org` mais le nom au début, `nexistesurementpas.um.outlook.com` est parfaitement conforme au RFC 1123, lui ! » Il faut examiner de plus près le processus de résolution DNS :

```
% dig  nexistesurementpas.um.outlook.com
...
;; ANSWER SECTION:
nexistesurementpas.um.outlook.com. 0 IN CNAME *.um.outlook.com.glbdns2.microsoft.com.
*.um.outlook.com.glbdns2.microsoft.com. 0 IN CNAME wildcard-emeasouth.um.outlook.com.
wildcard-emeasouth.um.outlook.com. 300 IN A 157.55.9.252
```

Le nom est en fait un pointeur (enregistrement DNS de type `CNAME`) vers un nom canonique, `*.um.outlook.com` (qui, à son tour, pointe vers un autre nom canonique, `wildcard-emeasouth.um.outlook.com`, mais ce n'est pas important ici).

Or, le sous-programme `getaddrinfo()`, qu'appellent les applications pour résoudre un nom en adresse IP, teste la validité de ces noms. Sur les systèmes utilisant la GNU libc (ce qui est le cas de l'Ubuntu sur laquelle j'ai fait les tests), `getaddrinfo` est encore plus violent, il teste **tous** les noms de la chaîne. Arrivant au `*.um.outlook.com.glbdns2.microsoft.com`, il le rejette car contenant un caractère invalide, l'astérisque. Ainsi, la résolution échoue et le nom est considéré comme inconnu. (Les fanas du DNS noteront que, dans un fichier de zone, l'astérisque a un sens spécial, faisant du nom un **joker**. Mais ça n'a pas d'importance ici : c'est un caractère illégal, point.)

À noter que le résultat dépend du système d'exploitation utilisé car toutes les mises en œuvre de `getaddrinfo` ne sont pas aussi sévères. Apparemment, FreeBSD refuse ces noms, mais pas Mac OS ou Windows. Ainsi, sur Mac OS, même le nom le plus bizarre est accepté :

```
$ ping -c 1 '&-funny-%-syntax-$.bortzmeyer.org'
PING www.bortzmeyer.org (204.62.14.153): 56 data bytes
64 bytes from 204.62.14.153: icmp_seq=0 ttl=51 time=73.471 ms

--- www.bortzmeyer.org ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 73.471/73.471/73.471/0.000 ms
```

Windows, lui, accepte le premier nom (la machine ne répond pas aux requêtes ICMP mais peu importe) :

```
E:\Users\moi>ping -n 1 nexistesurementpas.um.outlook.com

Envoi d'une requête 'ping' sur wildcard-emeacenter.um.outlook.com [157.55.9.252] avec 32 octets de données :
Délai d'attente de la demande dépassé.

Statistiques Ping pour 157.55.9.252:
    Paquets : envoyés = 1, reçus = 0, perdus = 1 (perte 100%),
```

Les programmeurs n'ont qu'à jeter un œil au code source de la GNU libc, dans `resolv/res_comp.c` :

```
#define alphachar(c) (((c) >= 0x41 && (c) <= 0x5a) \
   || ((c) >= 0x61 && (c) <= 0x7a))
#define digitchar(c) ((c) >= 0x30 && (c) <= 0x39)

#define borderchar(c) (alphachar(c) || digitchar(c))
#define middlechar(c) (borderchar(c) || hyphenchar(c) || underscorechar(c))
#define domainchar(c) ((c) > 0x20 && (c) < 0x7f)

int
res_hnok(const char *dn) {
int pch = PERIOD, ch = *dn++;

while (ch != '\0') {
int nch = *dn++;

if (periodchar(ch)) {
(void)NULL;
} else if (periodchar(pch)) {
if (!borderchar(ch))
return (0);
} else if (periodchar(nch) || nch == '\0') {
if (!borderchar(ch))
return (0);
} else {
if (!middlechar(ch))
return (0);
}
pch = ch, ch = nch;
}
return (1);
}
```

Ce code teste que le nom de machine suit l'expression rationnelle `[a-z0-9\.\-]+` (pas tout à fait : il autorise aussi le trait bas à certains endroits, ce qui est une erreur). Le code équivalent pour FreeBSD est `gethostbydns.c`.

À noter qu'on a testé que le cas où le nom illégal est au milieu de la chaîne des pointeurs. Il serait intéressant de refaire le test avec tous ces systèmes d'exploitation pour les cas où le nom illégal est au début ou à la fin de cette chaîne.

Un post-scriptum pour mes lecteurs juristes : les définitions de domaine et de machine utilisées ici sont évidemment celles des RFC. Mais le gouvernement se moque bien de la terminologie existante et, dans sa hâte de contrôler l'Internet, il invente un vocabulaire bizarre. C'est ainsi que le décret

---

https://www.bortzmeyer.org/host-vs-domain.html

français n° 2015-125 du 5 février 2015 <http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000030195477&dateTexte=&categorieLien=id>, qui institue la censure administrative en France <https://www.bortzmeyer.org/censure-francaise.html> a cette définition pittoresque « Les adresses électroniques figurant sur la liste comportent soit un nom de domaine (DNS), soit un nom d'hôte caractérisé par un nom de domaine précédé d'un nom de serveur. »

Merci à Kim-Minh Kaplan pour la discussion et à Vincent Archer et Pascal Courtois pour avoir prêté des machines à la science.