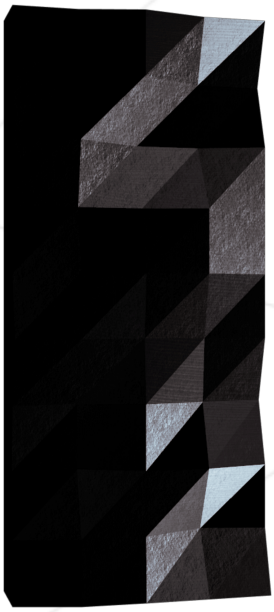

Project UNIX

ft_nmap

42 Staff pedago@staff.42.fr

Résumé: Ce projet consiste à recoder une partie du scanner de port nmap.

Table des matières

Chapitre I

Préambule

Enrico Fermi (29 septembre 1901 à Rome - 28 novembre 1954 à Chicago) est un physicien italo-américain. Ses recherches serviront de socle à l'exploitation de l'énergie nucléaire.

Il est lauréat du prix Nobel de physique de 1938 « pour sa démonstration de l'existence de nouveaux éléments radioactifs produits par bombardements de neutrons, et pour sa découverte des réactions nucléaires créées par les neutrons lents1 ». Il fut également lauréat de la médaille Hughes en 1942, de la médaille Franklin en 1947 et du prix Rumford en 1953.

Enrico Fermi naît le 29 septembre 1901 à Rome. Fils d'Alberto Fermi, inspecteur-chef au ministère des Communications, et d'Ida de Gattis, enseignante d'école élémentaire, Enrico est le dernier d'une fratrie de trois (sa sœur Marie et son frère Giulio, âgés respectivement de deux et un an de plus que lui). Très jeune, Enrico Fermi fait preuve d'une mémoire exceptionnelle et d'une grande intelligence, qui lui permettent d'exceller dans les études. Durant son enfance, il est inséparable de son frère Giulio. Mais en 1915, Giulio meurt au cours d'une opération chirurgicale visant à lui ôter un abcès de la gorge. Enrico, profondément marqué, se jette alors dans l'étude de la physique pour surmonter sa douleur. Bon élève, il se passionne très vite pour la physique et les mathématiques et commence à étudier divers ouvrages qu'il achète et qui traitent de mécanique, d'optique, d'astronomie et d'acoustique. Un ami de son père, l'ingénieur Adolfo Amidei, qui prend conscience des qualités hors du commun du jeune Fermi, lui prête divers ouvrages traitant de mathématiques. Ainsi, à 17 ans, Enrico Fermi maîtrise la géométrie analytique, la géométrie projective, le calcul infinitésimal, le calcul intégral et la mécanique rationnelle.

Á partir d'octobre 1918, Fermi étudie à l'Université de Pise au sein de l'École normale supérieure de Pise avec Franco Rasetti. Comme à son habitude, il étudie seul divers problémes de physique mathématique et consulte des ouvrages de Poincaré, de Poisson ou d'Appell. À partir de 1919, il s'intéresse aux nouvelles théories comme la relativité ou la physique atomique, acquiert une grande connaissance de théories telles que la relativité restreinte, la théorie du corps noir ou encore le modèle de l'hydrogène de Bohr. Ainsi Enrico Fermi, le seul à l'université au fait de ces théories, en arrive, sur l'insistance de ses professeurs, à donner des conférences où il expose aux professeurs et aux assistants les dernières découvertes de physique atomique.

Source.

Chapitre II

Introduction

Nmap est un scanner de ports libre créé par Fyodor et distribué par Insecure.org. 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.

Chapitre III

Objectifs

Le but du sujet est de vous faire recoder une partie du programme nmap et ainsi découvrir une nouvelle bibliothèque très puissante.
Vous allez aussi devoir utiliser les threads dans le but de diminuer drastiquement le temps pour scanner les ports choisis.

```
$> man nmap
```

Ce projet implique d'utiliser surtout les biblothèque PCAP (-lpcap) et THREAD(-lpthread).

Chapitre IV

Partie obligatoire

Usage :

```
$> ft_nmap [--help] [--ports [NOMBRE/PLAGE]] --ip ADRESSE IP [--speedup [NOMBRE]] [--scan [TYPE]]
```

ou

```
$> ft_nmap [--help] [--ports [NOMBRE/PLAGE]] --file FICHIER [--speedup [NOMBRE]] [--scan [TYPE]]
```

- L'exéctuable devra se nommer ft_nmap.
- Un menu d'aide doit être disponible.
- Vous devez gérer uniquement la version IPv4 (adresse/hostname), comme paramètre du programme, pour vos scans.
- Vous devez gérer le FQDN sans pour autant effectuer la résolution DNS.
- Il doit être possible de choisir le nombre de threads (default :0 max :250), dans le but de rendre le scan plus rapide.
- Il doit être possible de lire une liste d'IPs depuis un fichier (format libre).
- Votre programme doit pouvoir lancer ces types de scan :
  - SYN, NULL, ACK, FIN, XMAS, UDP

  Si le type du scan n'est pas précisé alors les 6 types seront utilisés.
- On doit pouvoir lancer chaque type de scan individuellement, et plusieurs types de scan simultanément.
- Les ports à scanner peuvent être passés sous forme de plage de port ou individuellement. Dans le cas où aucun port n'est précisé le scan doit se lancer sur la rangée 1-1024.

- La limite maximale du nombre de ports scannés ne peut dépasser 1024.
- La résolution des types des services sera demandée (pas la version, mais uniquement le TYPE).
- L'affichage du résultat du/des scans doit être le plus propre et clair possible. Le temps doit être clairement lisible.

Vous avez le droit d'utiliser les fonctions de la famille printf ainsi qu'une globale.

Pour les malins (ou pas)... Bien sur vous ne pouvez pas appeler le vrai nmap.

- Voici un exemple d'affichage d'aide possible :

```
 ./ft_nmap --help
Help Screen
ft_nmap [OPTIONS]
 --help      Print this help screen
 --ports     ports to scan (eg: 1-10 or 1,2,3 or 1,5-15)
 --ip        ip addresses to scan in dot format
 --file      File name containing IP addresses to scan,
 --speedup   [250 max] number of parallel threads to use
 --scan      SYN/NULL/FIN/XMAS/ACK/UDP
```

- Voici un exemple de resultat possible :

```
$> ./ft_nmap --ip x.x.x.x --speedup 70 --port 70-90 --scan SYN
Scan Configurations
Target Ip-Address : x.x.x.x
No of Ports to scan : 20
Scans to be performed : SYN
No of threads : 70
Scanning..
........
Scan took 8.32132 secs
IP address: x.x.x.x
Open ports:
Port      Service Name (if applicable)  Results                       Conclusion
-------------------------------------------------------------------------------------
80        http                          SYN(Open)                     Open

Closed/Filtered/Unfiltered ports:
Port      Service Name (if applicable)  Results                       Conclusion
-------------------------------------------------------------------------------------
90        Unassigned                    SYN(Filtered)                 Filtered
89        Unassigned                    SYN(Filtered)                 Filtered
88        kerberos                      SYN(Filtered)                 Filtered
87        link                          SYN(Filtered)                 Filtered
86        Unassigned                    SYN(Filtered)                 Filtered
85        Unassigned                    SYN(Filtered)                 Filtered
84        Unassigned                    SYN(Filtered)                 Filtered
83        Unassigned                    SYN(Filtered)                 Filtered
82        Unassigned                    SYN(Filtered)                 Filtered
81        Unassigned                    SYN(Filtered)                 Filtered
79        finger                        SYN(Filtered)                 Filtered
78        Unassigned                    SYN(Filtered)                 Filtered
77        rje                           SYN(Filtered)                 Filtered
76        Unassigned                    SYN(Filtered)                 Filtered
75        Unassigned                    SYN(Filtered)                 Filtered
74        Unassigned                    SYN(Filtered)                 Filtered
73        Unassigned                    SYN(Filtered)                 Filtered
72        Unassigned                    SYN(Filtered)                 Filtered
71        Unassigned                    SYN(Filtered)                 Filtered
70        gopher                        SYN(Filtered)                 Filtered
```

- Voici un autre exemple de resultat possible :

```
$>./ft_nmap --ip x.x.x.x --speedup 200 --port 75-85
Scan Configurations
Target Ip-Address : x.x.x.x
No of Ports to scan : 10
Scans to be performed : SYN NULL FIN XMAS ACK UDP
No of threads : 200
Scanning..
..............
Scan took 16.21338 secs
IP address: x.x.x.x
Open ports:
Port      Service Name (if applicable)  Results                                Conclusion
-------------------------------------------------------------------------------------------------
80        http                          SYN(Open) NULL(Closed) FIN(Closed)
                                        XMAS(Closed) ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Open

Closed/Filtered/Unfiltered ports:
Port      Service Name (if applicable)  Results                                Conclusion
-------------------------------------------------------------------------------------------------
85        Unassigned                    SYN(Filtered) NULL(Closed) FIN(Closed)
                                        XMAS(Closed) ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Closed
84        Unassigned                    SYN(Filtered) NULL(Closed) FIN(Closed)
                                        XMAS(Closed)ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Closed
83        Unassigned                    SYN(Filtered) NULL(Closed) FIN(Closed)
                                        XMAS(Closed) ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Closed
82        Unassigned                    SYN(Filtered) NULL(Closed) FIN(Closed)
                                        XMAS(Open|Filtered) ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Closed
81        Unassigned                    SYN(Filtered) NULL(Closed) FIN(Closed)
                                        XMAS(Closed)  ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Closed
79        finger                        SYN(Filtered) NULL(Closed) FIN(Closed)
                                        XMAS(Closed) ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Closed
78        Unassigned                    SYN(Filtered) NULL(Closed) FIN(Closed)
                                        XMAS(Closed) ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Closed
77        rje                           SYN(Filtered) NULL(Open|Filtered)
                                        FIN(Closed) XMAS(Closed) ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Closed
76        Unassigned                    SYN(Filtered) NULL(Open|Filtered)
                                        FIN(Closed XMAS(Closed) ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Closed
75        Unassigned                    SYN(Filtered) NULL(Closed) FIN(Closed)
                                        XMAS(Closed) ACK(Unfiltered)
                                        UDP(Open|Filtered)                     Closed
```

Chapitre V

Partie bonus

Les bonus ne seront comptabilisés que si votre partie obligatoire est PARFAITE. Par PARFAITE, on entend bien évidemment qu'elle est entièrement réalisée, et qu'il n'est pas possible de mettre son comportement en défaut, même en cas d'erreur aussi vicieuse soit-elle, de mauvaise utilisation, etc ... Concrètement, cela signifie que si votre partie obligatoire n'est pas validée, vos bonus seront intégralement IGNORÉS.

Des idées de bonus :

- support de l'IPv6.
- gestion DNS/version.
- Détection du systéme d'exploitation.
- Flag pour passer au dessus des IDS/firewall.
- Pouvoir camoufler son adresse source.
- Ajout de flag divers...

Chapitre VI

Rendu et peer-évaluation

- Ce projet ne sera corrigé que par des humains. Vous êtes donc libres d'organiser et nommer vos fichiers comme vous le désirez, en respectant néanmoins les contraintes listées ici.
- Vous devez coder en C et rendre un Makefile (respectant les règles habituelles).
- Vous devez gérer les erreurs de façon raisonnée. En aucun cas votre programme ne doit quitter de façon inattendue (Segmentation fault, etc).
- Rendez-votre travail sur votre dépot `GiT` comme d'habitude. Seul le travail présent sur votre dépot sera évalué en soutenance.
- Vous devez être sous une VM avec un noyau Linux > 3.14. Pour info le barême a été fait avec une Debian 7.0 stable.
- Dans le cadre de votre partie obligatoire, vous avez le droit d'utiliser les fonctions suivantes :
  - alarm
  - bind
  - close
  - connect
  - exit
  - fflush, fileno, fopen, fwrite, fclose
  - freeifaddrs
  - getservbyport, gethostbyname, getifaddrs
  - gettimeofday
  - getuid
  - htonl, htons, ntohs

- inet_addr
- inet_ntoa, inet_ntop, inet_pton
- pcap_breakloop, pcap_close, pcap_compile, pcap_dispatch
- pcap_geterr, pcap_lookupdev, pcap_lookupnet, pcap_open_live
- pcap_setfilter
- perror
- poll
- pthread_create, pthread_exit, pthread_join
- pthread_mutex_init, pthread_mutex_lock, pthread_mutex_unlock
- recvfrom, recv
- sendto
- setsockopt, socket
- sigaction, sigemptyset
- strspn
- les fonctions de la famille printf.
- les fonctions autorisées dans le cadre de votre libft(read, write, malloc, free, par exemple :-)).
- Vous avez l'autorisation d'utiliser d'autres fonctions dans le cadre de vos bonus, à condition que leur utilisation soit dûment justifiée lors de votre correction. Soyez malins.

- Vous pouvez poser vos questions sur le forum, sur jabber, IRC, slack...