Network Programming Project 4 - SOCKS4

NP TA

Deadline: Thursday, 2018/12/27 23:59

1 Introduction

In this project, you are asked to implement the **SOCKS4 firewall protocol** in the application layer of the OSI model.

SOCKS is similar to a proxy (i.e. intermediary-program) that acts as both server and client for the purpose of making request on behalf of other clients. Because the SOCKS protocol is independent of application protocols, it can be used for many different services: telnet, ftp, www, etc.

There are two types of the SOCKS operations (i.e. CONNECT and BIND). You have to implement both of them.

2 SOCKS4 Implementation

SOCKS server starts listening, if a SOCKS client connects, use fork() to tackle with it.
Each child process will do:

1. Receive **SOCKS4_REQUEST** from SOCKS client
2. Get destination ip and port from **SOCKS4_REQUEST**
3. Check firewall(socks.conf), and send **SOCKS4_REPLY** to SOCKS client if rejected
4. Check CD value and choose one of them
   (a) CONNECT (CD=1)
      i. Connect to destination
      ii. Send **SOCKS4_REPLY** to SOCKS client
      iii. Start relaying traffic on both directions
   (b) BIND (CD=2)
      i. Bind and listen a port
      ii. Send **SOCKS4_REPLY** to SOCKS client, tell the listening port
      iii. (SOCKS client tells destination to connect to SOCKS server)
      iv. Accept connection from destination and send **SOCKS4_REPLY** to SOCKS client again
      v. Start relaying traffic on both directions

If SOCKS server decides to reject a SOCKS client, the connection will be disconnected immediately.

```
SOCKS4_REQUEST packet (VN=4, CD=1 or 2)
Type 1:
      +----+----+----+----+----+----+----+----+----+....+----+
      | VN | CD | DSTPORT |       DSTIP       |    USERID    |NULL|
      +----+----+----+----+----+----+----+----+----+....+----+
bytes:   1    1      2              4              variable     1

Type 2: (No need to implement this in NP project4)
      +----+----+----+----+----+----+----+----+----+....+----+----+----+....+----+
      | VN | CD | DSTPORT |   DSTIP(0.0.0.x)  |    USERID    |NULL| DOMAIN NAME  |NULL|
      +----+----+----+----+----+----+----+----+----+....+----+----+----+....+----+
bytes:   1    1      2              4              variable     1      variable     1

e.g.
DSTIP=140.113.1.2
DSPPORT=1234 (hint: 1234 = 4*256 + 210 = DSTPORT[0]*256 + DSTPORT[1])
USERID=MOZ
      +----+----+----+----+----+----+----+----+----+----+----+----+
      | 4  | 1  | 4  |210 |140 |113 | 1  | 2  | M  | O  | Z  |    |
      +----+----+----+----+----+----+----+----+----+----+----+----+
bytes:   1 |  1 |    2    |         4         |    variable  |  1


SOCKS4_REPLY packet (VN=0, CD=90(accepted) or 91(rejected or failed))
      +----+----+----+----+----+----+----+----+
      | VN | CD | DSTPORT |       DSTIP       |
      +----+----+----+----+----+----+----+----+
bytes:   1    1      2              4
```

Please refer to these webpages for more detailed SOCKS4 specification.
SOCKS4
SOCKS4A

3 Requirements

- Part I: SOCKS4 Server Connect Mode
  - Open your browser and connect to any webpage.
  - Turn on and set your SOCKS server, then
    - Be able to connect any webpages on google search.
    - Your SOCKS server need to show messages below:

```
<S_IP>: source ip
<S_PORT>: source port
<D_IP>: destination ip
<D_PORT>: destination port
<Command>: CONNECT or BIND
<Reply>: Accept or Reject
```

- Part II: SOCKS4 Server Bind Mode
  - FlashFXP settings:
    - Set your SOCKS server
    - Connection type is FTP
    - Data connection mode is Active Mode(PORT)
  - Connect to ftp server, and upload/download file bigger than 1GB.
    e.g. Ubuntu 18.04 iso (download link)
    - Upload file and download file.
    - Check whether SOCKS server's output has used BIND mode.
- Part III: CGI Proxy
  - Modify console.cgi in Project 3 to implement SOCKS client operation
    - Accept SocksIP and SocksPort paramenter in QUERY_STRING, as **sh** and **sp**
    - Use SocksIP and SocksPort to connect to your SOCKS server(by CONNECT mode)
    - Rename console.cgi to hw4.cgi in Makefile
  - Clear browser's proxy setting
    Open your http server, connect to panel_socks.cgi
    Key in IP, port, filename, SocksIP, SocksPort
    Connect to 5 ras/rwg servers through SOCKS server and check the output Test Case (same as Project 3, no hidden test case) t1.txt-t5.txt
- Firewall
  - You only need to implement a simple firewall. Write permitted IPs into socks.conf
    (deny all traffic by default)

```
e.g.
permit c 140.114.*.*          # permit NTHU IP (connect mode)
permit c 140.113.*.*          # permit NCTU IP (connect mode)
permit b *.*.*.*              # permit all IP (bind mode)
```

  - Be able to change firewall rules without restarting SOCKS server.

4 About Submission

1. E3

   (a) Create a directory named your student ID, put your files in the **same directory layer**.

   (b) You must provide a **Makefile**, which compiles your source code into two **executables** named **hw4.cgi**(modified from Project 3) and **socks_server**. The executables should be under the same directory as the source codes. We will use these executables for demo.

   (c) Upload **only** your code and Makefile. (e.g. **console.cpp**, **socks_server.cpp**...)
   Do not upload anything else (e.g. **http_server.cpp**, **panel_socks.cgi**...)

   (d) **zip** the directory and upload the .zip file to the E3 platform
   Attention!! we only accept .zip format

```
e.g.
Create a directory 0756000, the directory structure may be:
0756000
    |-- Makefile
    |-- console.cpp
    |-- socks_server.cpp
    |...
zip the folder 0756000 into 0756000.zip, and upload 0756000.zip onto E3
```

2. Bitbucket:

   (a) Create a **private** repository: ${your_student_ID}_np_project4 inside the **nctu_np_2018** team, under **np_project4**.
   Set the ownership to **nctu_np_2018**

   ```
   e.g. 0756000_np_project4
   ```

   (b) For each project, you need to commit on bitbucket for **at least 5 times**.

   (c) You can push anything you need onto bitbucket as long as the size of the file is reasonable.

3. **We take plagiarism seriously.**

```
All projects will be checked by a cutting-edge plagiarism detector.
You will get zero points on this project for plagiarism.
Please don't copy-paste any code from the internet, this may be
considered plagiarism as well.
Protect your code from being stolen.
```