

Laborator 9. Clienți de rețea

Cunoștințe și abilități ce vor fi dobândite

- Identificarea porturilor deschise pe un sistem
- Folosirea clienților de rețea în linia de comandă în Linux (`netcat`, `wget`, `curl`, `mail`)
- Accesarea serviciilor de web, transfer de fișiere (`FTP`) și e-mail

Cheat sheet

- Cheat Sheet

Pregătire infrastructură de laborator

- **Reminder:** avem nevoie de o mașină virtuală a laboratorului. Vă rugăm urmăriți [pagina aceasta pentru instrucțiuni](#), apoi reveniți.
- Schimbați utilizatorul curent în `root` folosind comanda

```
student@host:~$ sudo su
```

- Pentru a pregăti configurația de laborator, pe mașina virtuală (stația `host`) folosiți comenzile următoare din contul utilizatorului `root` de pe stația `host` (puteți da copy/paste la comenzi în terminal):

```
root@host:~# update_lab --force
root@host:~# start_lab clients
```

- Deschideți trei noi tab-uri în terminal (folosiți combinația de taste `Ctrl+Shift+T`), și conectați-vă, din nou, la mașina virtuală folosind comanda `ssh` de mai sus.
- De pe cele trei noi tab-uri, conectați-vă la cele trei containere (`red`, `green` și `blue`).
- Pentru o conectare mai ușoară puteți folosi aliasul `go` (ex. `go red`)

În mod implicit folosiți contul `root` pentru conectare pe toate stațiile. Aveți nevoie de drepturi privilegiate pentru configurare. Folosiți contul `student` doar unde vi se cere explicit.

Navigare

Laboratorul 9

- [01. \[15p\] Observare porturi deschise pe o stație](#)
- [02. \[10p\] Port forwarding pentru accesare serviciu web](#)
- [03. \[10p\] Accesare serviciu web folosind wget](#)
- [04. \[15p\] Descărcare recursivă de resurse web](#)
- [05. \[5p\] Accesare URL conținând caractere speciale](#)
- [06. \[5p\] Accesare de pagini web folosind curl](#)
- [07. \[5p\] Download FTP folosind curl \(fără autentificare\)](#)
- [08. \[10p\] Download FTP folosind curl \(cu autentificare\)](#)
- [09. \[10p\] Trimitere e-mail din linia de comandă](#)
- [10. \[15p\] Trimitere e-mail cu adresă personalizată din linia de comandă](#)
- [11. \[BONUS - 10p\] Upload prin FTP](#)
- [12. \[BONUS - 10p\] Trimitere e-mail cu atașament din linia de comandă](#)

Exerciții

În cadrul exercițiilor din laboratoarele de Linux vom folosi [topologia de mai sus](#).

01. [15p] Observare porturi deschise pe o stație

Utilitarul `netcat` este folosit pentru a crea conexiuni server-client. Acesta poate îndeplini atât rolul de server, cât și de client.

Pentru a crea un server care ascultă conexiuni pe portul TCP `2024` rulăm comanda

```
root@host:~# netcat -l 2024
```

Parametrul `-l` înseamnă că `netcat` este în modul `listen` (server), iar `2024` este portul pe care ascultă: `netcat` va crea un socket de tip server și va aștepta conexiuni pe portul `2024`. Terminalul va rămâne în așteptare, pentru ca un client să se conecteze serverul deschis cu `netcat`. **NU** închideți terminalul.

Dorim să verificăm faptul că pe stația `host` un serviciu ascultă pe portul `2024`. Deschidem un nou terminal către stația `host` și rulăm comanda

```
root@host:~# netstat -tlnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
[...]
```

```
tcp      0      0 0.0.0.0:2024      0.0.0.0:*      LISTEN    4958/netcat
[...]
```

Comanda **netstat** afișează serviciile TCP care așteaptă conexiuni pe stația locală. Observăm că programul **netcat** ascultă pe portul **2024**. Opțiunile comenzii **netstat** sunt:

- **t** - afișează conexiuni TCP
- **l** - afișează serviciile care **așteaptă** (*listen*) conexiuni
- **n** - dezactivează rezolvare de nume pentru adrese IP și porturi; de obicei dezactivăm rezolvarea numelor pentru a elimina latența procesului de rezolvare;
- **p** - afișează numele și PID-ul procesului de tip serviciu de rețea care așteaptă conexiuni pe portul dat.

În afișare construcția **0.0.0.0:2024** din coloana **Local Address** înseamnă că serviciul ascultă conexiuni pe toate adresele IP/interfețele disponibile sistemului pe portul **2024**.

În terminalul nou deschis către stația **host** conectați-vă la serviciul proaspăt creat. Folosiți pentru aceasta tot comanda **netcat** în modul client:

```
root@host:~# netcat localhost 2024
```

Comanda a creat un socket de tip client și s-a conectat la serviciul de mai sus (conexiune TCP pe portul **2024**). În continuare comanda așteaptă intrare de la utilizator. Mesajele pe care le scriem vor fi transmise serverului care le va afișa la ieșirea standard. Scrieți câteva linii și observați afișarea acestora pe partea serverului. Închideți folosind combinația de taste **Ctrl+d**. Se închide și serverul.

Pe stația **host**, realizați pașii de mai sus pentru UDP:

- creați un server care ascultă pachete UDP pe portul 2024;
- listați serviciile care ascultă pachete UDP pe stația **host**;
- porniți un client UDP și comunicați cu serverul de mai sus.

Pentru a afla ce opțiune trebuie transmisă comenzii **netcat** pentru a porni un server sau client UDP, consultați pagina de manual a comenzii și căutați șirul **UDP**.

La fel, pentru a afla ce opțiune trebuie transmisă comenzii **netstat** pentru a afișa socketii UDP, consultați pagina de manual a comenzii și căutați șirul **udp**.

02. [10p] Port forwarding pentru accesare serviciu web

Pe stația **red** este instalat un server de HTTP pe care dorim să îl accesăm de pe stația **fep.grid.pub.ro**. Pentru acest lucru trebuie să redirecțăm portul **8080** de pe stația **host** spre portul **80** de pe stația **red** folosind DNAT (vezi [05. \[10p\] Port forwarding](#)).

Pentru testare, deschideți terminal pe **fep.grid.pub.ro** și accesați site-ul **http://10.9.X.Y:8080** [**http://10.9.X.Y:8080**] (înlocuiți cu IP-ul) prin intermediul utilitarului **curl**. Vă va apărea un cod **HTML** cu mesajul *Laboratorul 10 - pe red*. Exemplu:

```
student@mjohnir:~$ ssh -X mihai.carabas@fep.grid.pub.ro
mihai.carabas@fep.grid.pub.ro's password:
[your.name@fep-62-2 ~]$ curl http://10.9.X.Y:8080
```

03. [10p] Accesare serviciu web folosind wget

Sunt situații în care:

- vrem doar să descărcăm o resursă, nu să accesăm o pagină web și să o parcurgem;
- vrem să testăm serviciul web;
- nu avem acces la o interfață grafică (pe un server sau pe o mașină virtuală) pentru accesarea web;
- vrem să automatizăm, într-un script, procesul de acces al unor resurse în web.

În aceste situații, este util să folosim un client web în linie de comandă. Astfel de clienți web în linie de comandă sunt **wget** sau **curl**. În general aceste utilitare sunt folosite pentru descărcarea de fișiere sau alte resurse disponibile în web.

Pentru a descărca fișierul **HTML**, accesat anterior (la [02. \[10p\] Port forwarding pentru accesare serviciu web](#)) executați, pe stația **fep.grid.pub.ro**, respectiv pe stația **host** (urmăriți **promptul** pentru a determina stația), comenzile de mai jos (**<adresa-ip-mv>** o puteți obține din dashboard-ul OpenStack):

```
[mihai.carabas@fep-62-2 ~]$ wget http://<adresa-ip-mv>:8080
--2013-11-26 18:20:10-- http://<adresa-ip-mv>:8080/
Connecting to <adresa-ip-mv>:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 31 [text/html]
Saving to: 'index.html'

100%[=====] 31          --.-K/s   in 0s

2013-11-26 18:20:10 (4.92 MB/s) - 'index.html' saved [31/31]

[mihai.carabas@fep-62-2 ~]$ cat index.html
<h1>Laborator 10 - pe red</h1>

root@host:~# wget -q http://red/
root@host:~# ls
PacketTracer5  index.html
root@host:~# cat index.html
<h1>Laborator 10 - pe red</h1>
```

Avem același conținut afișat în browser-ul web grafic: șirul *Laboratorul 10 - pe red* formatat **heading 1**.

În a doua comandă am folosit opțiunea **-q** a **wget** pentru a dezactiva afișarea informații legate de transfer a **wget**. Opțiunea trebuie folosită cu grijă. Informațiile de eroare furnizate de **wget** sunt utile pentru depanare și sunt dezactivate la folosirea opțiunii **-q**.

La adresa `http://red/file.dat` [`http://red/file.dat`] se află un fișier cu dimensiunea de 1 KB. Descărcați fișierul pe stația `host` folosind `wget`. Folosiți comanda `ls -l` pentru a verifica descărcarea corectă și dimensiunea sa.

Serverul web este configurat să folosească și conexiuni sigure (protocolul HTTPS). Descărcăm fișierul de mai sus (`file.dat`) pe stația `host` folosind protocolul HTTPS:

```
root@host:~# wget https://red/file.dat
--2013-11-23 17:50:52-- https://red/file.dat
Resolving red (red)... 192.168.1.2
Connecting to red (red)|192.168.1.2|:443... connected.
ERROR: cannot verify red's certificate, issued by `CN=red':
  Self-signed certificate encountered.
To connect to red insecurely, use `--no-check-certificate'.
```

Observăm că ni s-a afișat o eroare că nu poate fi verificat certificatul. Această verificare poate fi eliminată folosind opțiunea `--no-check-certificate`:

```
root@host:~# wget --no-check-certificate https://red/file.dat
--2013-11-23 17:52:02-- https://red/file.dat
Resolving red (red)... 192.168.1.2
Connecting to red (red)|192.168.1.2|:443... connected.
WARNING: cannot verify red's certificate, issued by `CN=red':
  Self-signed certificate encountered.
HTTP request sent, awaiting response... 200 OK
Length: 1024 (1.0K)
Saving to: `file.dat'

100%[=====>] 1,024      --.-K/s   in 0s

2013-11-23 17:52:02 (18.2 MB/s) - `file.dat' saved [1024/1024]
```

04. [15p] Descărcare recursivă de resurse web

La adresa `http://red/folder` se află o structură de directoare și fișiere. Pe stația `host`, descărcați întreaga structură folosind o singură comandă `wget` cu parametrul corespunzător.

Pentru a identifica opțiunea corespunzătoare, căutați în pagina de manual a `wget` după șirul `recursive`.

Folosiți utilitarul `tree` pentru a lista conținutul directorului descărcat. Este posibil ca pachetul `tree` să fie deja instalat pe `host`. Dacă nu este instalat, atunci puteți folosi următoarea comandă pentru a-l instala:

```
root@host:~# apt-get install tree
```

Când rulați `tree` pe directorul descărcat observați prezența mai multor fișiere cu nume de forma `index.html?C=S;O=D`. Aceste fișiere sunt fișiere de tip directory listing generate de serverul web și nu avem nevoie de ele. Ștergeți structura de directoare pe care tocmai ați descărcat-o pe stația `host` și descărcați-o din nou ignorând aceste fișiere.

Urmăriți discuția de aici [<http://stackoverflow.com/questions/273743/using-wget-to-recursively-fetch-a-directory-with-arbitrary-files-in-it>]. Să aveți în vedere și partea din discuția care conține șirul *"without the hostname"*.

Structura de directoare trebuie să fie similară cu:

```
root@host:~# tree folder/
folder/
├── embedded
│   ├── Qualcomm
│   │   ├── Krait
│   │   │   └── info.txt
│   │   └── Scorpion
│   │       └── info.txt
│   └── TI
│       ├── OMAP3
│       │   └── info.txt
│       └── OMAP4
│           └── info.txt
└── mobile
    ├── Apple
    │   ├── iPadAir
    │   │   └── info.txt
    │   └── iPhone5S
    │       └── info.txt
    └── Google
        ├── Nexus4
        │   └── info.txt
        └── Nexus7
            └── info.txt

14 directories, 8 files
```

05. [5p] Accesare URL conținând caractere speciale

Unele URL-uri pot avea caractere interpretate special în shell rezultând într-un comportament ciudat la descărcare.

Pentru a evidenția acest lucru, pe stația `host` descărcați resursa de la link-ul `http://localhost/login.php?name=Lab10&email=rl@upb.ro`:

```
root@host:~# wget http://localhost/login.php?name=Lab10&email=rl@upb.ro
[1] 3284
root@host:~# --2013-12-05 09:02:10-- http://localhost/login.php?name=Lab10
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 72 [text/html]
Saving to: `login.php?name=Lab10'
```

```
100%[=====>] 72      --.-K/s   in 0s
```

```
2013-12-05 09:02:10 (13.3 MB/s) - 'login.php?name=Lab10' saved [72/72]
```

```
[1]+ Done wget http://localhost/login.php?name=Lab10
root@host:~#
```

După output-ul comenzii de mai, pentru a vă reapărea prompt-ul consolei, apăsați pe **Enter** ca să vă reapară prompt-ul.

Observați că procesul intră în background (linia cu **[1] 3284** înseamnă că a fost pornit un job). Acest lucru se întâmplă întrucât simbolul **&** (*ampersand*) nu este escapat și se folosește rolul său special din shell: pornirea unui proces în background. Se observă din output că este separată comanda înainte de **&**, adică rezultă comanda `wget http://localhost/login.php?name=Lab10 [http://localhost/login.php?name=Lab10]`.

Stergeți fișierul descărcat:

```
root@host:~# ls
PacketTracer5 folder login.php?name=Lab10
root@host:~# rm login.php?name=Lab10
root@host:~# ls
PacketTracer5 folder
```

Folosiți escaping în shell [<http://steve-parker.org/sh/escape.shtml>] pentru a folosi în mod corect URL-ul și pentru a descărca resursa disponibilă la acel link.

Fișierul final descărcat trebuie să arate în acest fel (să conțină *Lab10* și *rl@upb.ro*). Folosiți `cat log<TAB>` pentru rula comanda de mai jos, ca să obțineți *autocompletion* din partea shell-ului (<TAB> înseamnă apăsarea tastei TAB):

```
root@host:~# cat login.php?name=Lab10&email=rl@upb.ro
<html>
<body>

Welcome Lab10<br>
Your email address is: rl@upb.ro
</body>
</html>
```

06. [5p] Accesare de pagini web folosind curl

Utilitarul **wget** este folosit pentru a descărca fișiere prin intermediul web. Utilitarul **curl** îndeplinește un rol similar, doar că afișează în mod implicit conținutul resurselor la consolă. Este indicat pentru afișarea de pagini web (format HTML) sau alte fișiere în format text.

Spre exemplu, dacă vrem să afișăm conținutul paginii de la adresa `http://red/index.html [http://red/index.html]` vom folosi comanda:

```
root@host:~# curl http://red/index.html
<h1>Laborator 10 - pe red</h1>
```

Observăm că output-ul comenzii **curl** este conținutul fișierului `index.html` descărcat anterior cu **wget**.

Un exemplu în mod particular util este aflarea adresei IP (externe) prin interogarea site-ului **icanhazip.com**:

```
root@host:~# curl icanhazip.com
141.85.225.204
```

Observăm că în urma interogării site-ului **icanhazip.com**, output-ul comenzii **curl** este adresa noastră IP externă (rutabilă).

În general output-ul întors de **curl** poate fi prelucrat folosind filtre de text (**grep**, **cut**, **tr**, **sort**, **awk**, **sed** etc.) pentru a obține informațiile de interes.

07. [5p] Download FTP folosind curl (fără autentificare)

Utilitarele **curl** și **wget** sunt folosite atât pentru transfer HTTP cât și pentru transfer FTP sau alte protocoale. Avantajul folosirii acestor utilitare în fața altora este că sunt neinteractive și pot fi adăugate în scripturi.

Vom folosi **curl** pentru a descărca un fișier prin FTP. Pe stația **red** rulează un server FTP și pe stația **host** vom folosi comanda **curl** pentru a descărca fișierul localizat la `/download/file-10M.dat`:

```
root@host:~# curl ftp://red/download/file-10M.dat
Warning: Binary output can mess up your terminal. Use "--output -" to tell
Warning: curl to output it to your terminal anyway, or consider "--output
Warning: <FILE>" to save to a file.
```

Observăm că apare un warning legat de output-ul de fișiere binare în terminal.

Pentru a salva într-un fișier conținutul de la distanță folosim opțiunea **-O** a comenzii **curl**:

```
root@host:~# curl ftp://red/download/file-10M.dat -O file-10M.dat
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 10.0M 100 10.0M 0 0 86.8M 0 --:--:-- --:--:-- --:--:-- 91.7M
root@host:~# ls -l file-10M.dat
-rw-r--r-- 1 root root 10485760 Nov 26 19:54 file-10M.dat
```

Mai sus utilitarul **curl** a afișat un sumar al transferului și apoi am validat, folosind comanda **ls -l** lungimea fișierului descărcat. URL-ul folosit începe cu **ftp://** pentru a semnaliza folosirea protocolului FTP pentru transfer.

Același efect ca mai sus poate fi obținut prin rularea comenzii **wget**:

```
root@host:~# wget ftp://red/download/file-10M.dat
--2013-11-26 19:55:49-- ftp://red/download/file-10M.dat
=> 'file-10M.dat'
Resolving red (red)... 192.168.1.2
```

```
Connecting to red (red)|192.168.1.2|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done. ==> PWD ... done.
==> TYPE I ... done. ==> CWD (1) /download ... done.
==> SIZE file-10M.dat ... 10485760
==> PASV ... done. ==> RETR file-10M.dat ... done.
Length: 10485760 (10M) (unauthoritative)

100%[=====] 10,485,760 --K/s in 0.07s

2013-11-26 19:55:49 (138 MB/s) - `file-10M.dat' saved [10485760]
```

08. [10p] Download FTP folosind curl (cu autentificare)

Ne propunem să folosim curl pentru descărcarea unui fișier FTP însă folosind autentificare. Fișierul căutat se găsește în directorul home al utilizatorului **ana** de pe red și se numește **ana-ftp-file-5M.dat**. Utilizatorul **ana** are parola **student**.

Pe stația **host** folosiți **curl** pentru a descărca, prin FTP, fișierul **ana-ftp-file-5M.dat** din directorul home al utilizatorului **ana**.

URL-ul folosit va fi **ftp://red/ana-ftp-file-5M.dat**. Va fi relativ la directorul home al utilizatorului autentificat.

Pentru autentificarea ca utilizatorul **ana** în **curl**, folosiți opțiunea **-u user**. Vedeți informații în pagina de manual a utilitarului **curl**. Comanda trebuie să fie neinteractivă: trebuie să furnizați în linia de comandă și utilizatorul și parola.

09. [10p] Trimitere e-mail din linia de comandă

Pentru activități de automatizare (transmiterea de notificări și anunțuri) este util să putem transmite e-mail-uri din linia de comandă în mod neinteractiv. Trimiterea din linia de comandă presupune scrierea a minim trei componente așa cum se întâmplă și într-un client de e-mail:

- destinatarul
- subiectul mesajului
- conținutul mesajului

Pentru transmiterea unui mesaj în linia de comandă folosim comanda **mail**. Pentru a transmite un mesaj de la utilizatorul **bogdan** către utilizatorul **corina** (ambii pe stația **host**), ne autentificăm ca utilizatorul **bogdan**:

```
root@host:~# su - bogdan
bogdan@host:~$
```

și folosim comanda (apăsăți **ENTER** acolo unde vă cere **CC**; nu vrem să introducem adresă de tipul *Carbon Copy*):

```
bogdan@host:~$ mail -s "Invitatie la film" corina
Cc:
Buna, Corina!

Vii la filmul "Teambuilding"?

Bogdan
^D
```

Mesajul de mai sus a fost finalizat folosind combinația de taste **Ctrl+d**, însemnând *End of File* (**EOF**).

Pentru a verifica faptul că mesajul e transmis, ne autentificăm ca utilizatorul **corina** și rulăm comanda:

```
corina@host:~$ mail
"/var/mail/corina": 1 message 1 new
>N 1 bogdan@host Tue Nov 26 21:07 17/419 Invitatie la film
? 1
[...]
Subject: Invitatie la film
To: <corina@host>
[...]
From: bogdan@host

Buna, Corina!

Vii la filmul "Teambuilding"?

Bogdan
? q
Saved 1 message in /home/corina/mbox
Held 0 messages in /var/mail/corina
```

Comanda **mail**, fără argumente citește căsuța poștală implicită a utilizatorului (din **/var/mail/<username>**). Promptul comenzii este **?** și permite introducerea de comenzi. Mai sus am folosit comanda **l** pentru a citi primul mesaj și apoi comanda **q** pentru a părăsi utilitarul. În conținutul mesajului apare și destinatarul (câmpul **To**) și subiectul mesajului (câmpul **Subject**).

Mesajul nu este obligatoriu să fie scris la intrarea standard ci poate fi redirectat dintr-un fișier sau o altă comandă. Un mesaj simplu poate fi trimis și folosind comanda:

```
bogdan@host:~$ echo 'Hello, Corina' | mail -s "Salutari" corina
```

Verificăm mesajul folosind comanda **mail** în contul utilizatorului **corina**:

```
corina@host:~$ mail
"/var/mail/corina": 1 message 1 new
>N 1 bogdan@host Tue Nov 26 21:13 13/371 Salutari
? 1
[...]
Subject: Salutari
To: <corina@host>
[...]
```

```
From: bogdan@host
Hello, Corina
? q
Saved 1 message in /home/corina/mbox
Held 0 messages in /var/mail/corina
```

Trimiteți un mesaj simplu din contul utilizatorului **bogdan** către o adresă a voastră de e-mail. Verificați și în Spam/Junk primirea mesajului. Dacă nu îl regăsiți, este posibil ca provider-ul de mail să filtreze mail-urile pe baza unor euristici.

Puteți folosi un serviciu precum Temp Mail [<https://temp-mail.org/>] pentru obținerea unei adrese de mail temporare pe care puteți testa trimiterea de mail-uri în cadrul laboratorului.

10. [15p] Trimitere e-mail cu adresă personalizată din linia de comandă

La exercițiul anterior, mesajul primit apare ca sosind de la **bogdan@host**. Nu este foarte elegant și am vrea să schimbăm adresa expeditorului.

Pentru a specifica adresa expeditorului vom folosi comanda **mail** disponibilă în pachetul **s-nail** care permite această opțiune. Este posibil ca pachetul **s-nail** să fie deja instalat pe **host**. Dacă nu este instalat, atunci puteți folosi următoarea comandă pentru a-l instala:

```
root@host:~# apt-get install s-nail
Reading package lists... Done
Building dependency tree
[...]
```

Odată ce pachetul **s-nail** a fost instalat, comanda **mail** este furnizată de acest pachet în mod implicit.

Trimiteți un mesaj la adresa voastră de e-mail astfel încât expeditorul mesajului să fie **bogdan.georgescu@rl.cs.pub.ro**.

Comanda **mail** din pachetul **s-nail** are opțiunea **-r** cu ajutorul căreia puteți preciza adresa de e-mail sursă care să fie folosită pentru trimiterea e-mail-ului.

Puteți folosi un serviciu precum Temp Mail [<https://temp-mail.org/>] pentru obținerea unei adrese de mail temporare pe care puteți testa trimiterea de mail-uri în cadrul laboratorului.

În forma sa simplă, protocolul SMTP nu face verificarea adresei sursă a utilizatorului. Dacă utilizatorul se conectează la *Mail Transfer Agent* de pe o stație considerată sigură (de exemplu chiar de pe stația locală – *localhost*), atunci acesta poate folosi orice adresă. Pentru autentificare se folosește SMTP cu suport SASL [http://en.wikipedia.org/wiki/Simple_Authentication_and_Security_Layer], care obligă autentificarea utilizatorului înainte de a folosi serviciul SMTP.

Detalii despre fenomen și prevenirea *e-mail forging/spoofing* găsiți aici [http://www.cert.org/tech_tips/email_spoofing.html].

11. [BONUS - 10p] Upload prin FTP

Dorim ca pe lângă download, să uploadăm fișiere prin FTP. Din rațiuni de securitate, nu se recomandă upload anonim, ci doar cu autentificare.

În directorul home al utilizatorului **bogdan** de pe stația **host** se găsește fișierul **bogdan-ftp-data-3M.dat**. Folosiți **curl** pentru a uploada acest fișier în directorul home al utilizatorului **ana** de pe stația **red**.

Căutați, în pagina de manual a comenzii **curl**, șirul **upload-file**.

După ce ați realizat upload-ul, descărcați prin FTP, din directorul home al utilizatorului **ana**, fișierul **bogdan-ftp-data-3M.dat** în directorul home al utilizatorului **corina** de pe stația **host**.

12. [BONUS - 10p] Trimitere e-mail cu atașament din linia de comandă

Dorim să trimitem din linia de comandă mesaje cu atașament.

În contul utilizatorului **bogdan** creați o arhivă (în ce format doriți) a directorului **/etc/init.d/**. Trimiteți arhiva ca atașament într-un e-mail trimis în linia de comandă către adresa voastră.

Pentru trimiterea atașamentului, va trebui să folosiți utilitarul **uuencode** oferit de pachetul **sharutils**. Este posibil ca pachetul **sharutils** să fie deja instalat pe **host**. Dacă nu este instalat, atunci puteți folosi următoarea comandă pentru a-l instala:

```
root@host:~# apt-get install sharutils
Reading package lists... Done
Building dependency tree
[...]
```

```
Setting up sharutils (1:4.11-1) ...
```

Pentru indicații legate de trimiterea de atașamente în e-mail consultați această pagină [<http://www.shelldorado.com/articles/mailattachments.html#uuencode>].

În modul uzual de folosire comanda **uuencode** primește două argumente cu aceeași valoare (de exemplu **store.zip store.zip**). Aceasta întrucât primul argument este numele fișierului care este codificat, iar al doilea argument este numele arhivei așa cum apare ea în atașament. Teoretic, este posibil să codifici un fișier și să aibă alt nume în atașament; în practică se folosește același nume, motiv pentru care apar două argumente cu aceeași valoare.

Folosiți opțiunea **-s "<subiect>"** la comanda **mail** pentru a preciza subiectul mesajului; înlocuiți **<subiect>** cu subiectul mesajului. Eticheta în Internet dictează să folosiți subiecte pentru e-mail-uri.

Navigare

Laboratorul 9

- 01. [15p] [Observare porturi deschise pe o stație](#)
- 02. [10p] [Port forwarding pentru accesare serviciu web](#)
- 03. [10p] [Accesare serviciu web folosind wget](#)
- 04. [15p] [Descărcare recursivă de resurse web](#)
- 05. [5p] [Accesare URL conținând caractere speciale](#)

- 06. [5p] Accesare de pagini web folosind curl
- 07. [5p] Download FTP folosind curl (fără autentificare)
- 08. [10p] Download FTP folosind curl (cu autentificare)
- 09. [10p] Trimitere e-mail din linia de comandă
- 10. [15p] Trimitere e-mail cu adresă personalizată din linia de comandă
- 11. [BONUS - 10p] Upload prin FTP
- 12. [BONUS - 10p] Trimitere e-mail cu atașament din linia de comandă