Esercizio 1

Passo 1. Si deve progettare un semplice script Perl, chiamato *askScholar*, che può essere invocato con la seguente sintassi:

`askScholar <paroleChiaveArgomentoSeparateDaSpazi>`

ad esempio:

`askScholar higgs boson`

il comando deve interrogare il portale `scholar.google.com` con un URL nel formato

```
http://scholar.google.com/scholar?hl=en&q=<paroleChiaveArgomentoSeparateda+>
```

e ritornare in standard output il *numero di risultati* che viene indicato in output nella frase "About N results"

Suggerimenti e osservazioni:

- Si può catturare l'output del comando shell *wget*, opportunamente invocato con `"wget -O - URL"`
  (stampa in standard output la risorsa appena recuperata)
- Il comando di cui sopra può ritornare una stringa vuota e/o un messaggio di errore dal server. Perché?

Passo 2. Si modifichi lo script di cui al passo 1 per accettare da standard input un file contenente una raccolta di diverse parole chiave, messe ciascuna su una linea separata.
Il nuovo script deve produrre in output una statistica, ordinata per valori decrescenti delle parole chiave con più risultati.
Alcuni dati di esempio sono disponibili su
http://www.mat.unical.it/ianni/storage/scholartestsetSOR.csv

Ad esempio, se il file di input contiene

TURING Alan
BASSOTTI Banda
MOUSE Mickey

L'output atteso potrebbe essere

TURING Alan	56600
MOUSE Mickey	32200
ASSOTTI Banda	353

Osservazioni: che cosa succede dopo aver interrogato in pochi minuti scholar.google.com con un certo numero di queries?

Esercizio 2. Wireshark on the wire (DNS, TCP, HTTP, SMTP)

Obiettivo

In questo esercizio analizzeremo che cosa succede sui livelli trasporto e applicazione durante una sessione (insieme di connessioni multiple) HTTP e una conversazione SMTP.

Procedura 1.

1. Fate partire Wireshark.
2. Aprite un browser web indicando un indirizzo non usato recentemente (per evitare che il valore DNS si trovi in cache) ma non premete ENTER.
3. Fate partire una sessione di cattura frame con Wireshark.
4. Premete ENTER nel browser.
5. Attendete che la pagina sia completamente caricata. Determinate l'indirizzo IP del vostro compagno di banco ed effettuate un comando ping <indirizzoIPdelvostrocollega>. Per sapere l'indirizzo IP di una macchina, digitate ipconfig (Windows) oppure ifconfig (Linux).
6. Salvate la pagina web come riferimento e infine terminate e poi salvate la cattura wireshark.

Protocol Analysis Questions

Analizzando l'elenco dei frame catturati, rispondete alle seguenti domande.

1. ***Protocolli catturati***
   - Esaminate la colonna dei protocolli catturati nella finestra di Wireshark, stendete un elenco di tutti i protocolli catturati, che dovrebbero contenere DNS, TCP, HTTP, ICMP. Ci sono delle sigle di protocolli a voi sconosciute? Completate la lista con una breve descrizione di ogni protocollo incontrato (sconosciuto e non)
   - Ci sono tra i frame catturati, dati di cui **non siete** destinatari? Stendete una lista degli indirizzi IP destinatario non corrispondenti all'indirizzo IP della vostra interfaccia di rete primaria.

2. ***Frame IP e UDP***
   - Cercate il primo frame DNS inviato da voi.
     a. Identificate l'indirizzo IP del mittente. Dovrebbe trovarsi nell'intestazione IP (che studieremo in dettaglio in seguito).
   - Guardate l'intestazione UDP del primo frame DNS
     a. Identificate il numero di porta sorgente e destinazione. Qual è la porta destinazione?

3. ***DNS***
   - Esaminate il payload (I dati) del primo frame DNS inviato.

a. Dove sta l'informazione che dice se il messaggio è una query o una risposta?
b. Il corpo della query cosa dice?
c. Quant'è il codice (ID) della query?
d. Qual'è il TIPO della query? In che campo viene codificato?

- Ora cercate la risposta alla query DNS fatta.
  a. Quali dovrebbero essere gli indirizzi IP di mittente e destinatario di questo frame? Verificate che corrispondano a ciò che vi aspettate.
  b. Quanti byte occupa il datagramma di risposta? E' più piccolo o più grande del datagramma di query?
  c. Verificate che l'ID della risposta corrisponda con l'ID della query.
  d. Quante risposte ci sono in totale nel messaggio di risposta? Hanno tutte lo stesso TTL? Cos'è il TTL nelle risposte DNS?

4. HTTP GET

- Trovate quei frame che trasportano dei comandi "HTTP GET". Identificate quante connessioni sono state usate.
  a. Confermate che i numeri di sequenza e acknowledgement sono quelli che vi aspettavate.
  b. Guardate i flag? Sapete spiegare perché i valori sono tali?
  c. Quant'è lungo il segmento TCP? Quanti byte sono invece i dati effettivi?
- Guardate ora il contenuto del comando GET.
  a. Confrontate i valori grezzi nel terzo riquadro in basso con i valori decodificati presentati nel secondo riquadro.
  b. Contate il numero di byte nel messaggio e verificate che questo numero corrisponda al campo lunghezza nell'intestazione TCP.
  c. Che numero di sequenza vi aspettate nel prossimo frame in arrivo dal server? Che numero di acknowledgement?

Procedura 2.

1. Fate partire una cattura Wireshark.
2. Aprite la vostra soluzione per la prima esercitazione che è stata condotta per il modulo di Reti di Calcolatori (Esercizio 2, implementazione del comando *sendMail*), o in alternativa compilate e poi usate i sorgenti disponibili qui: https://www.mat.unical.it/ianni/storage/SendMail.java
3. Provate a invocare *sendMail* indicando come mittente il VOSTRO indirizzo e-mail e come destinatario iannir@mat.unical.it. Usate come mail server SMTP *ml.mat.unical.it*.
4. Fermate la cattura.
5. Individuate in Wireshark i frame relativi alla conversazione SMTP appena svolta. Controllate di saper individuare l'IP del mail server SMTP che viene contattato. Qual è il suo nome? E qual è l'IP del client che vi si connette?
6. Cambiate ora l'indirizzo mittente in sorcio@sorcino.it e lanciate di nuovo il programma. Questa volta dovreste leggere un errore da console. Perché? Usate Wireshark per capire come si è esattamente svolta la conversazione SMTP.