Programmazione di Sistemi in Rete (A.A. 2017/18)

Esercitazione n. 16 del 22 dicembre 2017

Prof. Eugenio Zimeo

Gli esercizi che seguono richiedono l'adozione di un analizzatore di rete (packet sniffer). Allo scopo è possibile impiegare lo strumento WireShark recuperabile dal sito www.wireshark.org. WireShark, come altri analizzatori di rete, configura l'interfaccia di rete della propria macchina in modalità "promiscua". In tale modalità di funzionamento, l'interfaccia non è più selettiva rispetto al traffico dati osservato sulla rete ma è in grado di vedere qualsiasi pacchetto.
Useremo WireShark per osservare il traffico generato da interazioni di tipo HTTP tra il proprio browser ed un server Web remoto. Questa analisi ci consente di osservare cosa accade dal livello HTTP (applicazione) in giù durante tipiche interazioni di tipo richiesta/risposta prodotte durante la navigazione Web. Per rispondere ad alcune domande potete riportare i dati catturati da WireShark ed usare eventualmente diagrammi temporali per illustrare lo scambio dei segmenti.
Negli esercizi si farà riferimento a contenuti web preparati dagli autori del libro di testo (Kurose e Ross). Potete comunque fare riferimento ad altri contenuti equivalenti, nel caso quelli indicati non fossero raggiungibili.

Esercizio 16.1
(Interazione http richiesta/risposta per file html semplici e di piccole dimensioni)

Il primo esercizio consiste nell'analizzare il traffico di dati generato da una semplice richiesta di un file HTML (non contenente riferimenti ad altre risorse) da un server Web. Per svolgere l'esercizio si suggerisce di procedere nel modo seguente:

a) Fate partire il vostro browser Web.
b) Fate partire WireShark e selezionate un filtro http
c) Inserite il seguente URL nel browser:
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

Se tutto procede regolarmente dovreste visualizzare una pagina HTML con una sola riga.
d) Interrompete la cattura dei pacchetti da parte di WireShark.
e) Se avete commesso degli errori ripetete la procedura, ma accertatevi che sia trascorso almeno un minuto tra un accesso e l'altro alla pagina HTTP-wireshark-file1.html.

Dall'analisi dei pacchetti catturati osserverete che i messaggi HTTP viaggiano all'interno di segmenti TCP e questi a loro volta sono incapsulati nell'area dati dei datagram IP ed infine i datagram IP sono inseriti nell'area dati di frame Ethernet.

Avendo a disposizione i dati catturati, rispondente alle seguenti domande:
1. Il vostro browser sta usando la versione 1.0 o 1.1 di HTTP? Quale versione di HTTP è invece in esecuzione sul server?
2. Qual è l'indirizzo IP del vostro computer e del server gaia.cs.umass.edu?
3. Qual è il codice di stato restituito dal server al vostro browser?
4. Quando è stato modificato l'ultima volta il file che avete recuperato dal server?
5. Quanti byte sono stati inviati dal server al vostro browser?
6. Osservate altre intestazioni dei messaggi di richiesta e di risposta HTTP ? Se si, elencatele.
7. Analizzate i segmenti TCP che vengono scambiati. Quante connessioni sono attivate ?

individuate e riportate i segmenti di controllo usati per l'attivazione della(e) connessione(i) e per la chiusura.

Esercizio 16.2
(richiesta GET condizionata per file html semplici e di piccole dimensioni)

Da quanto studiato, sappiamo che i browser possono mantenere in cache le risorse restituite da un server Web. In questo caso, la richiesta di una risorsa ottenuta in precedenza viene realizzata usando una GET condizionata. Procedete come suggerito.

a) Fate partire il vostro browser web, e assicuratevi che la cache sia vuota.
b) Fate partire WireShark e selezionate il filtro http.
c) Inserite il seguente URL nel browser
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html
Se tutto procede regolarmente, il browser dovrebbe visualizzare una semplice pagina HTML di cinque righe.
d) Inserite nuovamente lo stesso URL nel browser (o premete il pulsante di refresh del vostro browser).
e) Premete ora il pulsante refresh del vostro browser, tenendo premuto nel contempo il tasto *Shift* (o *Maiuscolo*)
f) Interrompete la cattura dei pacchetti da parte di WireShark

Avendo a disposizione i dati catturati, rispondente alle seguenti domande:
1. La prima richiesta GET dal vostro browser al server contiene una linea di intestazione "If-Modified-Since" ?
2. Il server ha spedito effettivamente il contenuto del file? Spiegate come fate a capirlo?
3. Analizzando la seconda richiesta GET dal browser al server osservate una linea di intestazione "If-Modified-Since:" ? Se sì, qual è il valore del campo "If-Modified-Since" ?
4. Qual è il codice di stato e la frase restituita dal server in risposta al secondo GET? Il server ha spedito effettivamente il contenuto del file? Spiegate cosa è successo.
5. La terza richiesta GET dal browser al server contiene un'intestazione "If-Modified-Since:" ? Se sì, che informazione segue il campo "If-Modified-Since" ?
6. Qual è il codice di stato e la frase restituita dal server in risposta al terzo GET? Il server ha spedito effettivamente il contenuto del file? Spiegate cosa è successo.

Esercizio 16.3
(Interazione http richiesta/risposta per file html di grandi dimensioni)

Gli esercizi precedenti hanno riguardato lo scambio di messaggi HTTP di piccole dimensioni. Se i messaggi sono invece di grandi dimensioni non possiamo più pensare che un messaggio HTTP viaggi all'interno di un solo segmento TCP. Vediamo cosa succede con file di grandi dimensioni.

a) Fate partire il browser web, e assicuratevi che la cache sia vuota.
b) Fate partire WireShark e selezionate il filtro HTTP.
c) Inserite il seguente URL nel vostro browser
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html
Il browser dovrebbe visualizzare una pagina piuttosto voluminosa.
d) Interrompete la cattura dei pacchetti da parte di WireShark

Nella finestra con l'elenco dei pacchetti dovreste vedere un messaggio GET, seguito da una risposta. Selezionando quest'ultima, nella finestra dei dettagli, prima delle righe relative al protocollo HTTP, dovrebbe apparire una nuova riga denominata “Reassembled TCP Segments”. La riga contiene un elenco di numeri.

Tenete presente che il file recuperato è particolarmente lungo, circa 4500 byte, e pertanto non può essere contenuto all'interno di un solo segmento TCP. Il messaggio HTTP è stato quindi frammentato dal protocollo TCP, e ogni frammento viene mostrato nella sezione “Reassembled TCP Segments”.

Avendo a disposizione i dati catturati, rispondente alle seguenti domande:
1. Quanti messaggi HTTP GET sono stati inviati dal vostro browser? Quale numero di frammento nella traccia contiene il messaggio GET?
2. Qual è il codice di stato e la frase associata nella risposta?
3. Quanti segmenti contenenti dati sono stati necessari per trasportare l'unica risposta HTTP?
4. Quale numero di frammento nella traccia contiene il codice di stato e la frase associata con la risposta alla richiesta HTTP GET.
5. Ci sono righe di stato HTTP trasmesse nei pacchetti TCP di risposta successivi al primo?
6. Quali sono i numeri di sequenza e di riscontro dei segmenti TCP contenenti le porzioni del messaggio HTTP di risposta?
7. Quanto vale il thoughput relativo alla trasmissione della pagina HTML richiesta dal browser (throughput della comunicazione dal server verso il browser)?

Esercizio 16.4
(Interazione http richiesta/risposta per file html con contenuti integrati)

Vediamo ora con il supporto di WireShark cosa accade quando il browser scarica un file HTML che contiene riferimenti ad altri oggetti (per esempio immagini).

a) Fate partire il browser web, e assicuratevi che la cache sia vuota.
b) Fate partire WireShark e selezionate il filtro http.
c) Inserite il seguente URL nel browser
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html
Il browser dovrebbe visualizzare una piccola pagina con due immagini. Le due immagini non sono contenute ma referenziate nel file base HTML,
d) Interrompete la cattura dei pacchetti da parte di WireShark

Avendo a disposizione i dati catturati, rispondente alle seguenti domande:
1. Quante richieste HTTP GET sono state inviate dal vostro browser? A quali indirizzi IP sono state inviate queste richieste?
2. Potete affermare se il vostro browser ha scaricato la due immagini sequenzialmente, o se le ha scaricate in parallelo? Spiegate come siete giunti alle vostre conclusioni.

Esercizio 16.5
(Autenticazione HTTP)

Analizziamo adesso il flusso di messaggi HTTP utilizzati per gestire l'accesso a risorse protette mediante username e password.
La risorsa che si tenterà di recuperare è accessibile dopo aver specificato come username *wireshark-students* e come password *network*.

a) Assicuratevi che la cache del browser sia vuota, e uscite dal browser.
b) Fate ripartire il browser
c) Fate partire WireShark e selezionate il filtro HTTP.
d) Inserite il seguente URL nel browser
http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html
e digitate le credenziali per l'accesso quando vengono richieste.
f) Premete il pulsante refresh del vostro browser, tenendo premuto nel contempo il tasto *Shift* (o *Maiuscolo*)
g) Interrompete la cattura dei pacchetti da parte di WireShark.

Avendo a disposizione i dati catturati, rispondente alle seguenti domande:

1. Quante richieste GET sono state inviate al server?
2. Qual è la risposta del server (codice di stato e frase) per il messaggio GET iniziale del browser?
3. Quando il browser ha inviato il messaggio GET per la seconda volta, quali campi in più ha aggiunto nelle intestazioni?
4. Il terzo messaggio GET, che è stato inviato quando avete premuto il tasto di refresh, contiene le intestazioni presenti nel secondo messaggio oppure no?

Lo username (wireshark-students) e la password (network) che avete inserito sono codificati nella stringa di caratteri (d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=) che segue l'intestazione "Authorization: Basic" nel messaggio GET.

Sebbene possa sembrare che username e password siano criptate, esse sono semplicemente codificate nel formato noto come Base64. Risulta pertanto abbastanza evidente che l'uso delle sole credenziali in chiaro previste dallo schema di autenticazione BASIC non è sicuro.

Consegna
Le soluzioni degli esercizi devono essere consegnate (su dropbox) cinque giorni prima dell'esame e preferibilmente entro il 12 gennaio 2018.