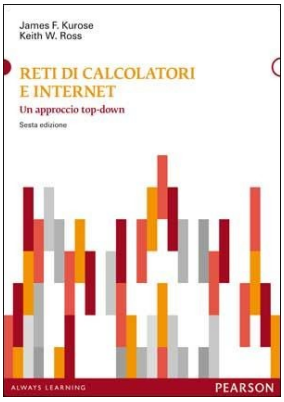
Laboratorio Wireshark: HTTP



Versione 6.1 italiano

Traduzione italiana di G. Amato, Ottavio M. D'Antona.
Modifiche e adattamenti per il CLEII di G. Amato.

Dopo aver rotto il ghiaccio con Wireshark nella lezione di laboratorio introduttiva, siamo pronti ad utilizzarlo per studiare i protocolli in azione. In questa lezione esploreremo vari aspetti del protocollo HTTP: l'interazione di base richiesta/risposta, i formati dei messaggi HTTP, il recupero di file HTML molto grossi, il recupero di file HTML con riferimenti a oggetti e infine la sicurezza e l'autenticazione. Prima di dare inizio alla lezione, potrebbe far comodo riguardare la Sezione 2.2 del libro di testo.

L'interazione di base: richiesta/risposta

Iniziamo ad esplorare il protocollo HTTP scaricando un semplice file HTML, uno molto piccolo che non contiene nessun altro oggetto. Fate quanto segue:

1. Fate partire il vostro browser.
2. Fate partire Wireshark, come descritto nella lezione di laboratorio introduttiva (ma non iniziate ancora la cattura dei pacchetti). Digitate “http” (solo le lettere, senza le virgolette) nel campo di specifica del filtro, in modo che, nella finestra con l'elenco dei messaggi catturati, vengano visualizzati solo quelli relativi al protocollo HTTP (siamo interessati solo a questi, e tutti gli altri farebbero confusione inutilmente).
3. Inserite la seguente URL nel browser:
   http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html
   Il vostro browser dovrebbe visualizzare una semplice pagina HTML con una sola riga.
4. Interrompete la cattura pacchetti di Wireshark.
5. 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.

La finestra di Wireshark dovrebbe essere simile a quella mostrata in Figura 1. Se non siete in grado di eseguire Wireshark da una computer connesso direttamente a Internet, potete scaricare una traccia di pacchetti che è stata creata seguendo i passi di cui sopra[1].

1 Scaricare il file zip http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip ed estrarre il file http-ethereal-trace-1. Le tracce nel file zip sono state raccolte da Wireshark da uno dei computer dell'autore, seguendo i passati indicati nella lezione. Una volta scaricata la traccia, potete caricarla dentro Wireshark usando il menù a discesa *File*, scegliendo *Open* e selezionando il file http-ethereal-trace-1. La finestra di Wireshark dovrebbe ora apparire come in Figura 1.

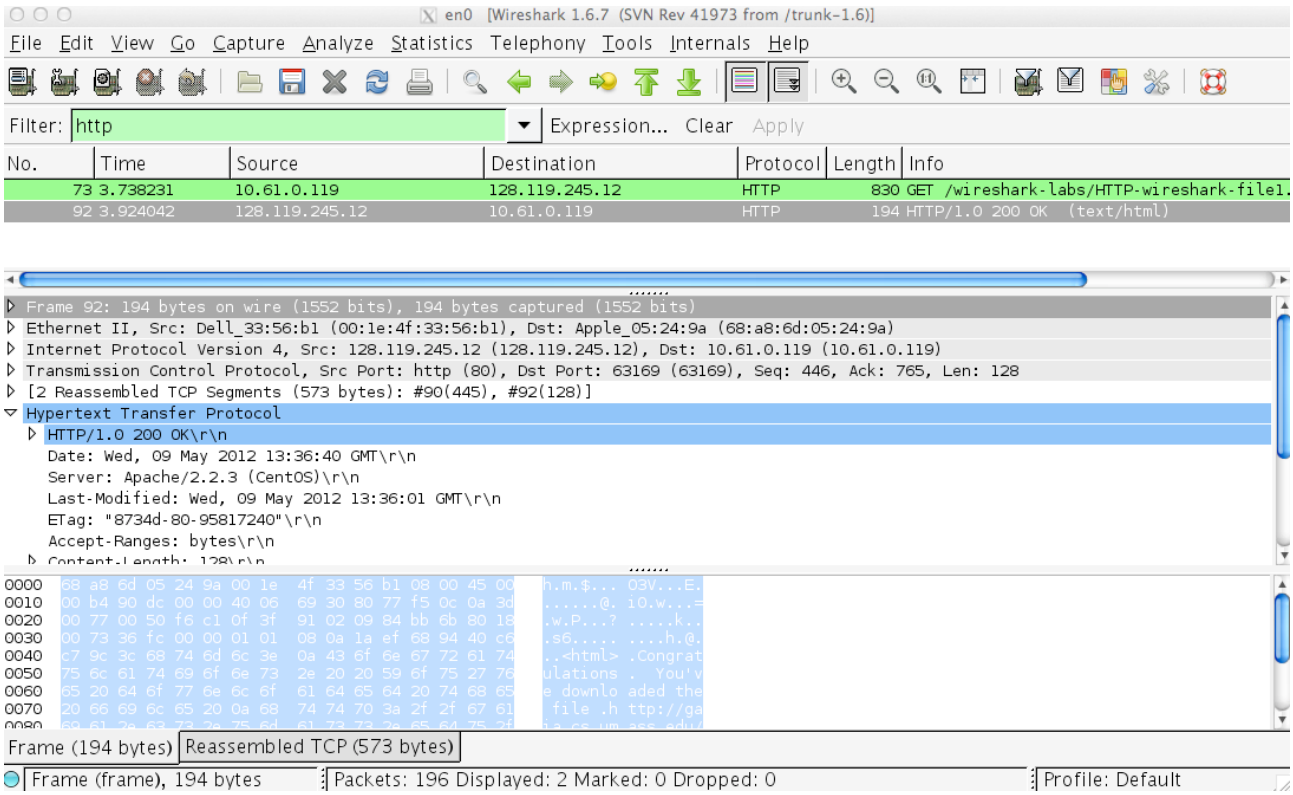


Figura 1: Finestra di Wireshark dopo aver caricato la pagina
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

L'esempio in Figura 1 mostra che l'elenco dei pacchetti catturati contiene due messaggi HTTP: il messaggio GET (dal vostro browser a gaia.cs.umass.edu) e il messaggio di risposta dal server al vostro browser. La finestra di dettaglio mostra informazioni dettagliate sul messaggio selezionato (in questo case il messaggio GET, che è evidenziato nell'elenco dei pacchetti). Ricordate che, poiché il messaggio HTTP è stato trasportato all'interno di un segmento TCP, a sua volta all'interno di un datagramma IP, quest'ultimo all'interno di un frame Ethernet, Wireshark visualizza anche le informazioni sui pacchetti Ethernet, IP e TCP. Vogliamo minimizzare l'ammontare di informazioni non pertinenti ad HTTP che vengono visualizzate (per ora siamo interessati ad HTTP, ci occuperemo dopo degli altri protocolli), pertanto assicuratevi che i quadrati a sinistra delle linee Frame, Ethernet, IP e TCP abbiano un segno più o un triangolo con la punta verso destra (che significa che c'è informazione nascosta non visualizzata), e che la linea HTTP abbia un segno meno o un triangolo con la punta verso il basso (che vuol dire che tutte le informazioni su HTTP sono visualizzate).

> Nota: dovete ignorare qualunque messaggio HTTP GET (o risposta) relativa al file favicon.ico. Se vedete un riferimento a questo file, vuol dire che il vostro browser sta chiedendo automaticamente al server una piccola icona da visualizzare accanto alla URL nella finestra del browser.

Guardando le informazioni nel messaggio GET e nella risposta, rispondete alle seguenti domande. Quando rispondete alle domande dovreste stampare i messaggi corrispondenti (la lezione introduttiva spiega come fare) e indicare dove, nel messaggio, avete trovato le informazioni che vi sono richiese.

1. Il vostro browser sta usando la versione 1.0 o 1.1 di HTTP? Quale versione di HTTP è in esecuzione sul server?
2. Quale linguaggi (se ce ne sono) dichiara il vostro browser di accettare?
3. Qual è l'indirizzo IP del vostro computer. E del server gaia.cs.umass.edu?
4. Qual è il codice di stato restituito dal server al vostro browser?
5. Quando è stato modificato l'ultima volta il file che avete recuperato dal server?
6. Quanti byte sono stati inviati al vostro browser?

7. Ispezionando i dati grezzi nella finestra con i contenuti dei pacchetti, vedete delle intestazioni che non sono state visualizzate nell'elenco dei pacchetti? Se sì, indicatene una.

Nella vostra risposta al punto 5 di cui sopra, potreste rimanere sorpresi dal fatto che il documento che avete caricato è stato modificato l'ultima volta un minuto prima del vostro download. Questo perché (per questo file particolare) il server gaia.cs.umass.edu cambia il tempo di ultima modifica, e lo fa una volta per minuto. Pertanto, se aspettate un minuto tra due accessi, sembrerà che il file sia stato modificato nel frattempo, e il vostro browser scaricherà una nuova copia del documento.

Per seguire in maniera più semplice il flusso di messaggi che browser e server si scambiano, potete cliccare col tasto destro su uno qualunque dei pacchetti che fanno parte della connessione TCP e selezionare "*Follow TCP Stream*". Apparirà una finestra con i messaggi scambiati a livello applicazione, in entrambe le direzioni, in ordine temporale.

L'interazione GET condizionale/risposta

Ricordate dalla Sezione 2.2.6 del libro di testo che la maggior parte dei browser mantiene una cache con gli ultimi oggetti scaricati, ed esegue una GET condizionale quando deve scaricare un oggetto già presente nella cache. Prima di eseguire i passi successivi, assicuratevi che la cache del vostro browser sia vuota (per Firefox, selezionate *Cronologia → Cancella cronologia recente → Cacnella Adesso*). Ora fate quanto segue:

- Fate partire il browser web, e assicuratevi che la cache sia vuota, come discusso sopra.
- Fate partire il packet sniffer Wireshark.
- Immettete la seguente URL nel browser
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html
  Il vostro browser dovrebbe visualizzare una semplice pagina HTML di cinque righe.
- Velocemente, inserite nuovamente la stessa URL nel browser (o semplicemente premete il pulsante di refresh nel vostro browser).
- Premete il pulsante refresh del vostro browser, questa volta tenendo premuto nel contempo il tasto *Shift* (o *Maiuscolo*)
- Interrompete la cattura pacchetti di Wireshark, e inserite "http" nella finestra del filtro, in modo da visualizzare solo i messaggi relativi al protocollo HTTP.
- (*Nota:* se non siete in grado di eseguire Wireshark su un computer connesso a Internet, potete usare la traccia di pacchetti http-ethereal-2 per rispondere alle domande di cui sotto; vedi nota 1 a piè di pagina. Questo file di traccia è stato raccolto da uno dei computer degli autori, eseguendo i passi elencati sopra)

Rispondete alle seguenti domande:

8. Ispezionate il contenuto della prima richiesta GET dal vostro browser al server. C'è una linea di intestazione "IF-MODIFIED-SINCE" ?
9. Ispezionate il contenuto della risposta del server. Il server ha spedito effettivamente il contenuto del file? Come fate a dirlo?
10. Ora ispezionate il contenuto della seconda richiesta GET dal browser al server. C'è una linea di intestazione "IF-MODIFIED-SINCE:" ? Se sì, che informazione segue l'intestazione "IF-MODIFIED-SINCE" ?
11. 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.
12. Ora ispezionate il contenuto della terza richiesta GET dal browser al server. C'è una linea di intestazione "IF-MODIFIED-SINCE:" ? Se sì, che informazione segue l'intestazione "IF-

MODIFIED-SINCE” ?

13. 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.

Recuperare documenti di grandi dimensioni

Nei nostri esempi, fino ad ora, i documenti scaricati sono stati file HTML semplici e piccoli. Vediamo adesso cosa succede quando scarichiamo un file HTML lungo. Fate la seguente:

- Fate partire il browser web, e assicuratevi che la cache sia vuota, come discusso sopra.
- Fate partire il packet sniffer Wireshark.
- Immettete la seguente URL nel browser
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html
  Il vostro browser dovrebbe visualizzare una pagina piuttosto voluminosa.
- Interrompete la cattura pacchetti di Wireshark, e inserite “http” nella finestra del filtro, in modo da visualizzare solo i messaggi relativi al protocollo HTTP.
- (*Nota:* se non siete in grado di eseguire Wireshark su un computer connesso a Internet, potete usare la traccia di pacchetti http-ethereal-3 per rispondere alle domande di cui sotto; vedi nota 1 a piè di pagina. Questo file di traccia è stato raccolto da uno dei computer degli autori, eseguendo i passi elencati sopra)

Nella finestra con l'elenco dei pacchetti dovreste vedere un messaggio GET, seguito da un 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. Su questo dobbiamo dilungarci un attimo. Ricordate dalla Sezione 2.2 (Figura 2.9 nel libro di testo) che i messaggi di risposta HTTP consistono di una riga di stato, seguita da varie righe di intestazione, seguite da una riga vuota, seguita dal corpo dell'entità. Nel nostro caso, il corpo dell'entità è il file HTML che è stato richiesto. Questo file è particolarmente lungo, circa 4500 byte, troppo per entrare all'interno di un pacchetto TCP. Il messaggio HTTP è stato quindi spezzato in vari pezzi dal protocollo TCP, ogni pezzo contenuto all'interno di un segmento TCP separato (vedi Figura 1.24 nel libro di testo), e il fatto che la singola risposta HTTP sia stata frammentata su più pacchetti TCP è indicata dalla riga “Reassembled TCP Segments” seguita dall'elenco dei pacchetti che formano l'intero messaggio HTTP.

Rispondete alle seguenti domande:

14. Quanti messaggi HTTP GET sono stati inviati dal vostro browser? Quale numero di pacchetto nella traccia contiene il messaggio GET?
15. Qual è il codice di stato e la frase associata nella risposta?
16. Quanti segmenti contenenti dati sono stati necessari per trasportare l'unica risposta HTTP?
17. Quale numero di pacchetto nella traccia contiene il codice di stato e la frase associata con la risponsa alla richiesta HTTP GET.
18. Ci sono righe di stato HTTP trasmesse nei pacchetti TCP di risposta successivi al primo?

Documenti HTML con oggetti integrati

Ora che abbiamo visto come Wireshark visualizza il traffico catturato per grossi file HTML, possiamo controllare cosa accade quando il browser scarica un file che contiene riferimenti ad altri oggetti (per esempio, immagini contenuti su un altro server). Fate quanto segue:

- Fate partire il browser web, e assicuratevi che la cache sia vuota, come discusso sopra.
- Fate partire il packet sniffer Wireshark.
- Immettete la seguente URL nel browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html

  Il vostro browser dovrebbe visualizzare una piccola pagina con due immagini. Queste due immagini sono referenziate nel file base HTML. Ovvero, le immagini non sono contenute nel file HTML; invece, il file HTML contiene le loro URL. Come discusso nel libro di testo, il vostro browser deve recuperare queste immagini dai siti web indicati.
- Interrompete la cattura pacchetti di Wireshark, e inserite “http” nella finestra del filtro, in modo da visualizzare solo i messaggi relativi al protocollo HTTP.
- (*Nota:* se non siete in grado di eseguire Wireshark su un computer connesso a Internet, potete usare la traccia di pacchetti http-ethereal-4 per rispondere alle domande di cui sotto; vedi nota 1 a piè di pagina. Questo file di traccia è stato raccolto da unodei computer degli autori, eseguendo i passi elencati sopra)

Rispondete alle seguenti domande:

19. Quante richieste HTTP GET sono state inviate dal vostro browser? A quali indirizzi Internet sono state inviate queste richieste?
20. 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.

Autenticazione HTTP

Finalmente, proviamo a visitare un sito web protetto da password, ed esaminiamo la sequenza di messaggi HTTP scambiati per tale sito. La URL http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html è protetta da password. L'username è “wireshark-students” (senza le virgolette), e la password è “network” (sempre senza le virgolette). Proviamo ad accedere a questo sito “sicuro”. Fate quanto segue:

- Assicuratevi che la cache del browser sia vuota, come discusso sopra, e uscite dal browser. Dopo, fate ripartire il browser
- Fate partire il packet sniffer Wireshark.
- Immettete la seguente URL nel browser http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html Digitate quando richiesto la username e la password.
- Premete il pulsante refresh del vostro browser, tenendo premuto nel contempo il tasto *Shift* (o *Maiuscolo*)
- Interrompete la cattura pacchetti di Wireshark, e inserite “http” nella finestra del filtro, in modo da visualizzare solo i messaggi relativi al protocollo HTTP.
- (*Nota:* se non siete in grado di eseguire Wireshark su un computer connesso a Internet, potete usare la traccia di pacchetti http-ethereal-5 per rispondere alle domande di cui sotto; vedi nota 1 a piè di pagina. Questo file di traccia è stato raccolto da uno dei computer degli autori, eseguendo i passi elencati sopra)

Ora esaminiamo l'output di Wireshark. Potreste voler leggere prima qualche informazione sulla autenticazione HTTP dando un'occhiata al materiale “HTTP Access Authentication Framework” disponibile su http://frontier.userland.com/stories/storyReader$2159 . Rispondete alle seguenti

domande:

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

La username (wireshark-students) e la password (network) che avete inserito sono codificate 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. Per rendervene conto, andate su http://ostermiller.org/calc/encode.html, inserite la stringa d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms= e premete "Decode" sulla riga Base64. Voilà! Avete tradotto dalla codifica Base64 ad ASCII, e dovreste riuscire a vedere il vostro usernamee la password, separati da due punti. Poiché chiunque può scaricare uno strumento come Wireshark e catturare i pacchetti (non solo i suoi) che passano per una interfaccia di rete, e poiché chiunque può tradurre da Base64 ad ASCII (lo avete appena fatto!), dovrebbe essere chiaro che delle semplici password su un sito web non sono sicure, a meno che non vengano prese delle misure addizionali.

Non abbiate paura! Come vedremo nel Capitolo 8, ci sono dei modi per rendere l'accesso web più sicuro. Ma per far ciò, avremo bisogno di qualcosa che vada ben oltre il semplice schema di autenticazione di HTTP.

ATTENZIONE!! La parte di esecitazione che segue si può svolgere solo dai computer dell'aula informatica.

Proxy e Aula Informatica

I computer dell'aula informatica possono accedere in maniera diretta soltanto ai server del G@SL. Tutti gli altri accessi al web avvengono passando attraverso un proxy che si trova installato sul server ipa. In quest'ultima parte dell'esercitazione, vedremo come si svolge il dialogo tra il browser e il proxy.

Questo dialogo è piuttosto complesso perché il server proxy richiede agli utenti di autenticarsi. Inoltre, la procedura di autenticazione utilizzata è più complessa di quella usata nell'esercizio precedente. Fate quanto segue:

- Uscite dal browser (dovrebbe essere sufficiente chiudere tutte le finestre ad esso relative)
- Fate ripartire il browser
- Iniziate la cattura dei pacchetti
- Inserite la seguente URL nel browser:

  http://nodycosy.unich.it/

- Interrompete la cattura dei pacchetti.

Rispondete alle seguenti domande:

25. Quante richieste GET sono state inviate in seguito al caricamento della pagina? A quale indirizzo IP sono state inviate le richieste? Si tratta dell'indirizzo IP del server nodycosy.unich.it?

26. Che differenza c'è tra il comando GET che appare in queste richieste e tutti i comandi GET che avete incontrato precedentemente?
27. Cosa ha risposto il server alla prima richiesta GET? Che tipo di protocollo viene richiesto per l'autenticazione?
28. Quante richieste GET sono state inviate? Spiegare il risultato.