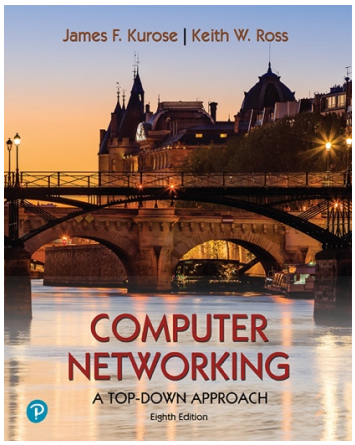
Laboratorio Wireshark: http versione 8.1



Supplemento al *Computer Networking: A Top-Down Approach, 8a ed.*, JF Kurose e KW Ross

"Dimmi e io dimentico. Mostramelo e io ricordo. Coinvolgimi e io capisco. proverbio cinese

---

Dopo esserci bagnati i piedi con lo sniffer di pacchetti Wireshark nel laboratorio introduttivo, ora siamo pronti per utilizzare Wireshark per indagare sui protocolli in funzione. In questo lab esploreremo diversi aspetti del protocollo HTTP: l'interazione GET/risposta di base, i formati dei messaggi HTTP, il recupero di file HTML di grandi dimensioni, il recupero di file HTML con oggetti incorporati e l'autenticazione e la sicurezza HTTP. Prima di iniziare questi laboratori, potresti voler rivedere la Sezione 2.2 del testo.[1]

1. L'interazione GET/risposta HTTP di base

Iniziamo la nostra esplorazione di HTTP scaricando un file HTML molto semplice, molto breve e che non contiene oggetti incorporati. Eseguire le seguenti operazioni:

1. Avvia il tuo browser web.
2. Avvia lo sniffer di pacchetti Wireshark, come descritto nel laboratorio introduttivo (ma non iniziare ancora l'acquisizione dei pacchetti). Immettere "http" (solo le lettere, non le virgolette e in minuscolo) nella finestra delle specifiche del filtro di visualizzazione, in modo che solo i messaggi HTTP acquisiti vengano visualizzati successivamente nella finestra dell'elenco dei pacchetti. (Siamo interessati solo al protocollo HTTP qui e non vogliamo vedere il disordine di tutti i pacchetti catturati).
3. Aspetta un po' più di un minuto (vedremo il motivo a breve), quindi inizia l'acquisizione dei pacchetti Wireshark.
4. Inserisci quanto segue nel tuo browser
   http://gaia.cs.umass.edu/wireshark-labs/HTT P -wireshark-file1.html
   Il tuo browser dovrebbe visualizzare il file HTML molto semplice di una riga.
5. Arresta l'acquisizione dei pacchetti Wireshark.

---

[1] I riferimenti a figure e sezioni si riferiscono all'ottava edizione del nostro testo, *Computer Networks, A Top-down Approach, 8 h ed., JF Kurose e KW Ross, Addison-Wesley/Pearson, 2020.*

La finestra di Wireshark dovrebbe essere simile alla finestra mostrata nella Figura 1. Se non sei in grado di eseguire Wireshark su una connessione di rete attiva, puoi scaricare una traccia del pacchetto che è stata creata quando sono stati seguiti i passaggi precedenti.[2]

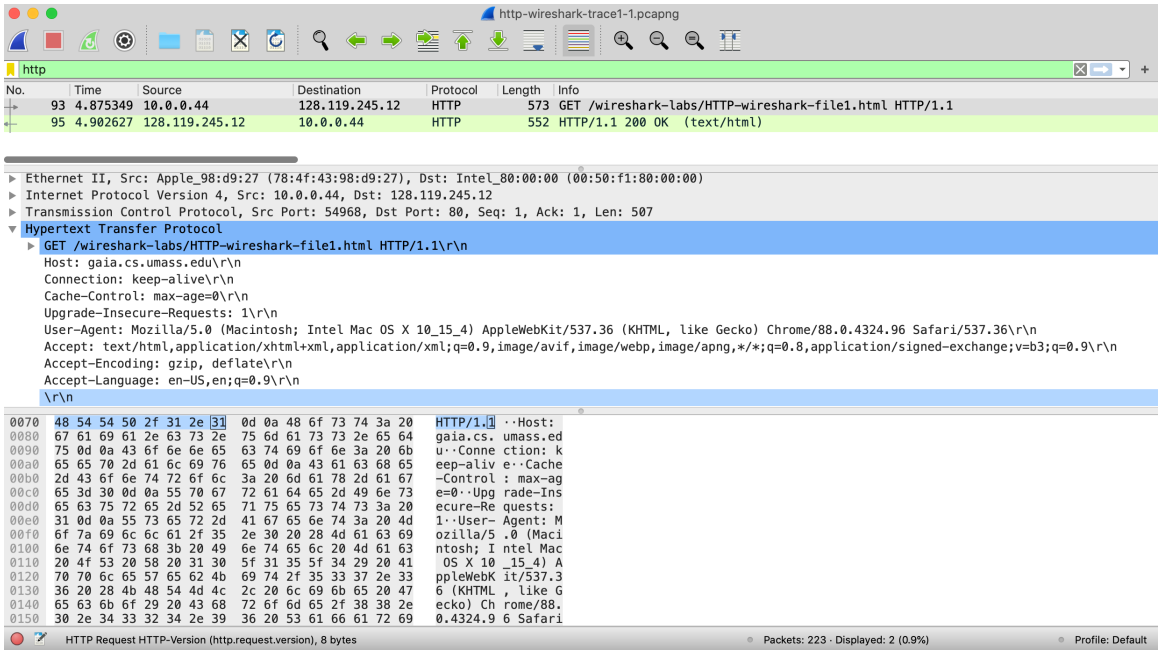


Figura 1: Visualizzazione di Wireshark dopo che http://gaia.cs.umass.edu/wireshark-labs/ HTTP-wireshark-file1.html è stato recuperato dal browser

L'esempio in Figura 1 mostra nella finestra di elenco dei pacchetti che sono stati catturati due messaggi HTTP: il messaggio GET (dal tuo browser al server web gaia.cs.umass.edu) e il messaggio di risposta dal server al tuo browser. La finestra del contenuto del pacchetto mostra i dettagli del messaggio selezionato (in questo caso il messaggio HTTP OK, evidenziato nella finestra dell'elenco dei pacchetti). Ricordiamo che poiché il messaggio HTTP è stato trasportato all'interno di un segmento TCP, che è stato trasportato all'interno di un datagramma IP, che è stato trasportato all'interno di un frame Ethernet, Wireshark visualizza anche le informazioni sui pacchetti Frame, Ethernet, IP e TCP. Vogliamo ridurre al minimo la quantità di dati non HTTP visualizzati (siamo interessati all'HTTP qui e indagheremo su questi altri protocolli in laboratori successivi), quindi assicurati che le caselle all'estrema sinistra di Frame, Ethernet, IP e Le informazioni TCP hanno un segno più o un triangolo rivolto verso destra (il che significa che sono presenti informazioni nascoste e non visualizzate) e la riga HTTP ha un segno meno o un triangolo rivolto verso il basso (il che significa che vengono visualizzate tutte le informazioni sul messaggio HTTP) .

> (*Nota:* dovresti ignorare qualsiasi HTTP GET e risposta per favicon.ico. Se vedi un riferimento a questo file, è il tuo browser che chiede automaticamente al server se (il server) ha un piccolo file

[2]È possibile scaricare il file traccia http-wireshark-trace1-1 dal google classroom del corso.

icona che dovrebbe essere visualizzato accanto a l'URL visualizzato nel tuo browser. I riferimenti a questo fastidioso file verranno ignorati in questo lab.).

Osservando le informazioni nell'HTTP GET e nei messaggi di risposta, rispondi alle seguenti domande.

1. Il tuo browser esegue HTTP versione 1.0, 1.1 o 2? Quale versione di HTTP è in esecuzione sul server?
2. Quali lingue (se presenti) il tuo browser indica che può accettare sul server?
3. Qual è l'indirizzo IP del tuo computer? Qual è l'indirizzo IP del server gaia.cs.umass.edu?
4. Qual è il codice di stato restituito dal server al tuo browser?
5. Quando è stato modificato l'ultima volta sul server il file HTML che stai recuperando?
6. Quanti byte di contenuto vengono restituiti al tuo browser?
7. Ispezionando i dati grezzi nella finestra del contenuto del pacchetto, vedi delle intestazioni all'interno dei dati che non sono visualizzate nella finestra dell'elenco dei pacchetti? Se è così, nominane uno.

Nella tua risposta alla domanda 5 sopra (supponendo che tu stia eseguendo Wireshark "live", invece di utilizzare un file di traccia registrato in precedenza), potresti essere sorpreso di scoprire che il documento che hai appena recuperato è stato modificato l'ultima volta entro un minuto prima hai scaricato il documento. Questo perché (per questo particolare file), il server gaia.cs.umass.edu sta impostando l'ora dell'ultima modifica del file in modo che sia l'ora corrente, e lo fa una volta al minuto. Pertanto, se si attende un minuto tra un accesso e l'altro, il file sembrerà essere stato modificato di recente e quindi il browser scaricherà una "nuova" copia del documento.

2. L'interazione HTTP CONDITIONAL GET/risposta

Ricordiamo dalla Sezione 2.2.5 del testo che la maggior parte dei browser Web esegue la memorizzazione nella cache degli oggetti e quindi spesso esegue un GET condizionale durante il recupero di un oggetto HTTP. Prima di eseguire i passaggi seguenti, assicurati che la cache del tuo browser sia vuota[3]. Ora fai quanto segue:

- Avvia il browser Web e assicurati che la cache del browser sia stata svuotata, come discusso in precedenza.
- Avvia lo sniffer di pacchetti Wireshark
- Inserisci il seguente URL nel tuo browser
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html
  Il tuo browser dovrebbe visualizzare un semplice file HTML di cinque righe.
- Inserisci di nuovo rapidamente lo stesso URL nel tuo browser (o seleziona semplicemente il pulsante di aggiornamento sul tuo browser)
- Interrompi l'acquisizione dei pacchetti Wireshark e inserisci "http" (di nuovo, in minuscolo senza virgolette) nella finestra delle specifiche del filtro di

[3]Vedere https://www.howtogeek.com/304218/how-to-clear-your-history-in-any-browser/ per istruzioni su come svuotare la cache del browser.

visualizzazione, in modo che solo i messaggi HTTP acquisiti vengano visualizzati successivamente nella finestra dell'elenco dei pacchetti.

Se non sei in grado di eseguire Wireshark su una connessione di rete live (o non riesci a far sì che il tuo browser emetta un campo If-Modified-Since sulla seconda richiesta HTTP GET), puoi scaricare una traccia del pacchetto che è stata creata quando i passaggi precedenti sono stati seguiti[4]. Rispondi alle seguenti domande:

8. Ispeziona il contenuto della prima richiesta HTTP GET dal tuo browser al server. Vedi una riga "IF-MODIFIED-SINCE" nell'HTTP GET?
9. Esaminare il contenuto della risposta del server. Il server ha restituito esplicitamente il contenuto del file? Come puoi dirlo?
10. Ora ispeziona il contenuto della seconda richiesta HTTP GET dal tuo browser al server. Vedi una riga "IF-MODIFIED-SINCE:" in HTTP GET [5]? In tal caso, quali informazioni seguono l'intestazione "IF-MODIFIED-SINCE:"?
11. Qual è il codice di stato HTTP e la frase restituiti dal server in risposta a questo secondo HTTP GET? Il server ha restituito esplicitamente il contenuto del file? Spiegare.

3. Recupero di documenti lunghi

Nei nostri esempi finora, i documenti recuperati sono stati semplici e brevi file HTML. Vediamo ora cosa succede quando scarichiamo un lungo file HTML. Eseguire le seguenti operazioni:

- Avvia il browser Web e assicurati che la cache del browser sia stata svuotata, come discusso in precedenza.
- Avvia lo sniffer di pacchetti Wireshark
- Inserisci il seguente URL nel tuo browser
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html
  Il tuo browser dovrebbe visualizzare la piuttosto lunga US Bill of Rights.
- Interrompi l'acquisizione dei pacchetti Wireshark e inserisci "http" nella finestra delle specifiche del filtro di visualizzazione, in modo che vengano visualizzati solo i messaggi HTTP acquisiti.

Nella finestra di elenco dei pacchetti, dovresti vedere il tuo messaggio HTTP GET, seguito da una risposta TCP a più pacchetti alla tua richiesta HTTP GET. Assicurati che il tuo filtro di visualizzazione Wireshark sia cancellato in modo che la risposta TCP multi-pacchetto venga visualizzata nell'elenco dei pacchetti.

---

[4] Se non è possibile eseguire Wireshark su una connessione di rete attiva, puoi scaricare il file di traccia dal google classroom del corso.

[5] *Suggerimento:* idealmente, dovresti vedere un'intestazione If-Modified-Since poiché hai appena scaricato questa pagina pochi secondi fa. Tuttavia, a seconda del browser che stai utilizzando e del formato della risposta precedente del server al tuo GET iniziale, il tuo browser potrebbe non includere un If-Modified-Since anche se il documento è stato scaricato e memorizzato nella cache. Il browser Chrome è piuttosto bravo a utilizzare regolarmente If-Modified-Since. Ma Safari e Firefox sono molto più pignoli su quando usare If-Modified-Since. La vita non è sempre facile in pratica come in teoria!

Questa risposta a più pacchetti merita una piccola spiegazione. Ricordiamo dalla Sezione 2.2 (si veda la Figura 2.9 nel testo) che il messaggio di risposta HTTP consiste in una riga di stato, seguita da righe di intestazione, seguite da una riga vuota, seguita dal corpo dell'entità. Nel caso del nostro HTTP GET, il corpo dell'entità nella risposta è l' *intero* file HTML richiesto. Nel nostro caso qui, il file HTML è piuttosto lungo e con 4500 byte è troppo grande per stare in un pacchetto TCP. Il singolo messaggio di risposta HTTP viene quindi suddiviso in più parti dal TCP, con ciascuna parte contenuta all'interno di un segmento TCP separato (vedere la Figura 1.24 nel testo). Nelle versioni recenti di Wireshark, Wireshark indica ogni segmento TCP come un pacchetto separato e il fatto che la singola risposta HTTP sia stata frammentata su più pacchetti TCP è indicato dal "segmento TCP di una PDU riassemblata" nella colonna Info del display Wireshark.

Rispondi alle seguenti domande:

12. Quanti messaggi di richiesta HTTP GET ha inviato il tuo browser? Quale numero di pacchetto nella traccia contiene il messaggio GET per Bill o Rights?
13. Quale numero di pacchetto nella traccia contiene il codice di stato e la frase associati alla risposta alla richiesta HTTP GET?
14. Qual è il codice di stato e la frase nella risposta?
15. Quanti segmenti TCP contenenti dati sono stati necessari per trasportare la singola risposta HTTP e il testo della Carta dei diritti?

4. Documenti HTML con oggetti incorporati

Ora che abbiamo visto come Wireshark mostra il traffico di pacchetti catturato per file HTML di grandi dimensioni, possiamo vedere cosa succede quando il tuo browser scarica un file con oggetti incorporati, cioè un file che include altri oggetti (nell'esempio seguente, file immagine) che sono memorizzati su un altro server.

Eseguire le seguenti operazioni:

- Avvia il browser Web e assicurati che la cache del browser sia stata svuotata, come discusso in precedenza.
- Avvia lo sniffer di pacchetti Wireshark
- Inserisci il seguente URL nel tuo browser
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html
  Il tuo browser dovrebbe visualizzare un breve file HTML con due immagini. Queste due immagini sono referenziate nel file HTML di base. Cioè, le immagini stesse non sono contenute nell'HTML; invece gli URL per le immagini sono contenuti nel file HTML scaricato. Come discusso nel manuale, il tuo browser dovrà recuperare questi loghi dai siti web indicati. Il logo del nostro editore è recuperato dal sito web gaia.cs.umass.edu. L'immagine della nostra copertina dell'ottava edizione (una delle nostre copertine preferite) è memorizzata su un server in Francia.
- Interrompi l'acquisizione dei pacchetti Wireshark e inserisci "http" nella finestra delle specifiche del filtro di visualizzazione, in modo che vengano visualizzati solo i messaggi HTTP acquisiti.

Rispondi alle seguenti domande:

16. Quanti messaggi di richiesta HTTP GET ha inviato il tuo browser? A quali indirizzi Internet sono state inviate queste richieste GET?
17. Puoi dire se il tuo browser ha scaricato le due immagini in serie o se sono state scaricate dai due siti web in parallelo? Spiegare.

5 Autenticazione HTTP

Infine, proviamo a visitare un sito Web protetto da password ed esaminiamo la sequenza dei messaggi HTTP scambiati per tale sito. L'URL http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html è protetto da password. Il nome utente è "wireshark-students" (senza virgolette) e la password è "network" (di nuovo, senza virgolette). Allora accediamo a questo sito "sicuro" protetto da password. Eseguire le seguenti operazioni:

- Assicurati che la cache del browser sia stata svuotata, come discusso in precedenza, e chiudi il browser. Quindi, avvia il browser
- Avvia lo sniffer di pacchetti Wireshark
- Inserisci il seguente URL nel tuo browser
  http: / /gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html
  Digita il nome utente e la password richiesti nella casella a comparsa.
- Interrompi l'acquisizione dei pacchetti Wireshark e inserisci "http" nella finestra delle specifiche del filtro di visualizzazione, in modo che solo i messaggi HTTP acquisiti vengano visualizzati successivamente nella finestra dell'elenco dei pacchetti.
- *Nota:* se non sei in grado di eseguire Wireshark su una connessione di rete attiva, puoi utilizzare la "classica" traccia del pacchetto http-ethereal-trace-5 o altre tracce aggiuntive, come indicato nella nota 2, per rispondere alle domande seguenti.

Esaminiamo ora l'output di Wireshark. Potresti voler prima documentarti sull'autenticazione HTTP rivedendo il materiale di facile lettura su " HTTP Access Authentication Framework " su http://frontier.userland.com/stories/storyReader$2159

Rispondi alle seguenti domande:

18. Qual è la risposta del server (codice di stato e frase) in risposta al messaggio HTTP GET iniziale dal tuo browser?
19. Quando il tuo browser invia il messaggio HTTP GET per la seconda volta, quale nuovo campo viene incluso nel messaggio HTTP GET?

Il nome utente (wireshark-students) e la password (rete) immessi sono codificati nella stringa di caratteri (d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=) che segue l'intestazione " `Autorizzazione: Basic ` " nel messaggio HTTP GET del client. Sebbene possa sembrare che il nome utente e la password siano crittografati, sono semplicemente codificati in un formato noto come formato Base64. Il nome utente e la

password *non sono* crittografati! Per vederlo, vai su http://www.motobit.com/util/base64-decoder-encoder.asp e inserisci la stringa con codifica base64 d2lyZXNoYXJrLXN0dWRlbnRz e decodifica. *Ecco!* Hai tradotto dalla codifica Base64 alla codifica ASCII e quindi dovresti vedere il tuo nome utente! Per visualizzare la password, inserire il resto della stringa Om5ldHdvcms= e premere decode. Dal momento che chiunque può scaricare uno strumento come Wireshark e sniffare pacchetti (non solo i propri) passando attraverso il proprio adattatore di rete, e chiunque può tradurre da Base64 ad ASCII (l'hai appena fatto!), dovrebbe esserti chiaro che semplici password su WWW i siti non sono sicuri a meno che non vengano prese misure aggiuntive.

Non aver paura! Come vedremo nel Capitolo 8, ci sono modi per rendere più sicuro l'accesso al WWW. Tuttavia, avremo chiaramente bisogno di qualcosa che vada oltre il framework di autenticazione HTTP di base!