

incApache

Aionei Giorgia - 4942137

Caviglia Lisa - 4993924

Gastaldi Sebastiano - 2916642

Grosso Francesco - 4114387

Peres Andrea - 4817585

Parte di debug

per rendere più pulito l'output a livello server abbiamo modificato il file `.h` inserendo altre 2 definizioni preesistenti di debug (MY DEBUG e DEBUG TREADS) che se definite a compilazione producono delle stampe per rendere più chiaro il funzionamento del programma.

Abbiamo creato degli script per la bash per effettuare in modo più rigoroso i test con parametri fissati sempre uguali, consistenti in una serie di richieste dei metodi supportati dalla nostra applicazione (GET e HEAD) utilizzando entrambi i protocolli della versione http.

Inoltre abbiamo verificato il funzionamento del GET conditional (if-modify-since) e la presenza o meno di un cookie fornito o meno dal front-end (lato client).

Problematiche

- Dava un problema di double `free`, perché all'interno della `my_timegm` pensavamo di dover fare solo mutex lock e unlock, invece doveva essere anche allocato lo spazio di memoria per la variabile `tz`
- La data di scadenza dei cookie non è un dato molto consistente, non esiste variabile o array globale, come invece avviene per l'array `UserTracker[MAX_COOKIES]`, associato ad un dato cookie che tiene l'informazione della scadenza.

Inoltre abbiamo fornito una funzione ausiliaria la quale partendo dal presupposto che se un client identificato da un determinato cookie ha già effettuato una richiesta,

una data di scadenza è già stata fornita al cookie e quindi non dovrebbe essere modificata.

L'unica informazione rilevante è quando viene inserita nell'HEADER di risposta la stringa contenente:

SET-UserID = "pippo" Expires: `COOKIE_EXPIRES`

ma questo avviene solo quando l'UID fornito non è valido.

- Abbiamo riscontrato inoltre un problema di capire se la data di scadenza del cookie potesse essere fornita lato client o meno.

Provando ad implementare questa funzionalità la syscall `strptime` ogni tanto non converte bene la data, perché veniva settato male il campo `.tm_isdst` (intero della struttura `tm`), in quanto assumeva valori anomali (quando dovrebbe essere solo -1, 0, 1).

`my_timegm` tentando di convertire la struttura `tm` prodotta dalla `strptime` con il valore anomalo, ma anche con il valore uguale a 1, (i.e. `.tm_isdst`) ritornava -1, valore non coerente.

Però considerati gli output fornitoci dal programma del professore ci siamo resi conto che da browser non venivano comunicate informazioni al server relative alla data di scadenza dei cookie capendo così che non dovesse essere implementata.

- Non è stato facile decifrare il comportamento del counting dei cookie in quanto, l'output del programma fornitoci dal professore produce output inconsistenti.

```
http_option_line: Cookie: UID=1; UID=3; UserID=1
client provided UID Cookie 1 for the 32526 time
```

Inoltre non sapevamo se prendere il valore del UserID o UID per quanto riguardo il valore da assegnare, perché ogni tanto, con la `http_option_line` fornita nell'esempio, l'eseguibile del professore sceglieva il valore dell'UserID o quello dell'UID o nessuno in maniera non deterministica.

Abbiamo dedotta dal commento sulla parte to do che dovessimo scegliere il valore dell'UserID.

- La parte sui thread inizialmente ci ha causato problemi, visto che quando veniva chiusa la connessione il programma non riusciva a fare la syscall `join` dei thread.
- Abbiamo provato a fare i test con Apache jMeter, apparentemente funziona, anche se abbiamo trovato il software poco intuitivo.

Stato di completamento

incapache 7.0 e 7.1 completate

metodo opzionale POST non implementato