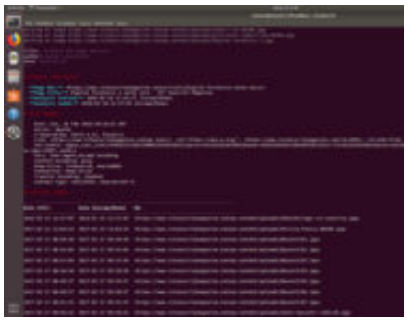
Datazione delle pagine web tramite Carbon14

Date : 13 marzo 2018



Durante le attività di OSINT (Open Source Intelligence) si può riscontrare l'esigenza di attribuire una datazione più precisa possibile ad una pagina web. Talvolta può essere necessario riuscire ad individuare un giorno o addirittura un orario ascrivibile alla creazione della pagina, ad esempio per rilevare casi di contraffazione della stessa.

Header HTTP e data del CMS

Agli albori del web, ciascun sito era composto da un insieme di pagine completamente statiche in formato HTML. In casi del genere, è possibile vedere la data di ultima modifica di una pagina semplicemente visionando gli header HTML. Consideriamo ad esempio **il primo sito web in assoluto** che è ancora consultabile sul sito del CERN.

Possiamo esplorare gli header HTTP inviando una richiesta di tipo HEAD, per esempio usando HTTPie [4]:

```
> http head 'http://info.cern.ch/hypertext/WWW/TheProject.html'  HTTP/
1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Length: 2
217  Content-Type: text/html  Date: Sun, 11 Feb 2018 21:11:19 GMT  ETa
g: "40521e06-8a9-291e721905000"  Last-
Modified: Thu, 03 Dec 1992 08:37:20 GMT  Server: Apache
```

La data di ultima modifica della pagina è di oltre 25 anni fa, proprio agli albori del web!

Purtroppo questo metodo attualmente non funziona più molto bene per svariati siti. Al giorno d'oggi moltissime pagine, specialmente sulle piattaforme di blogging o i portali di notizie, vengono *generate in modo dinamico* da parte di un CMS (Content Management System) che preleva il testo dell'articolo da un database e lo inserisce in un template grafico preimpostato.

Questo fa sì che la data di ultima modifica non dia più alcuna informazione utile. Facendo un test su un altro articolo di questa rivista [1] vediamo che l'header Last-Modified non è proprio

presente:

```
> http head 'https://www.ictsecuritymagazine.com/articoli/digital-forensics-costo-zero/'  HTTP/1.1 200 OK  Connection: Keep-Alive  Content-Encoding: gzip  Content-Length: 20  Content-Type: text/html; charset=UTF-8  Date: Sun, 11 Feb 2018 21:27:30 GMT  Keep-Alive: timeout=10, max=10000  Link: ; rel="https://api.w.org/", ; rel=shortlink  Server: Apache  Vary: User-Agent,Accept-Encoding  X-Powered-By: PHP/5.6.33  X-Powered-By: PleskLin
```

Questo sito usa WordPress, ma altri CMS potrebbero mostrare un valore di ultima modifica che corrisponde **sempre** al momento in cui l'utente richiede di vedere la pagina. Se consideriamo il fatto che soltanto WordPress fa girare oltre il 25% dei siti web [5], ci rendiamo conto che questa situazione è estremamente comune.

Pur essendo vero che la maggior parte dei CMS mostra un'indicazione sul giorno e l'ora di pubblicazione di un articolo, questa informazione ha alcuni aspetti negativi. Innanzitutto, si tratta soltanto del momento della *pubblicazione* di un post. Non è possibile usarla per stimare se quell'articolo fosse stato preparato nelle ore o addirittura nei giorni precedenti. In secondo luogo, questo dato può essere **facilmente manomesso** usando l'apposita funzione dei CMS più diffusi o altresì rimosso dal template.

Analisi delle risorse statiche

Se si vuole stimare precisamente la finestra temporale in cui è stato scritto un articolo è necessario cambiare approccio. Una tecnica semplice ma ingegnosa consiste nello sfruttare le risorse statiche linkate nella pagina [2], in altre parole **si considerano le immagini.**

Abbiamo quindi l'algoritmo per stilare una timeline relativa alla scrittura di una pagina web:

- scaricare il codice HTML della pagina
- per ogni elemento contenuto:
    - individuare l'attributo src
    - effettuare una richiesta HEAD per il file immagine
    - annotare il valore dell'header Last-Modified
- ordinare cronologicamente i risultati ottenuti

Va notato che su alcuni siti il metodo HEAD viene bloccato. Si possono comunque ottenere gli header HTTP anche effettuando una richiesta GET e interrompendo subito la connessione, prima di ricevere i dati veri e propri, così da risparmiare del tempo.

Alla fine dovremo eliminare manualmente eventuali immagini non pertinenti (ad esempio gli elementi grafici del template del sito) e avremo una timeline che considera tutte le immagini di nostro interesse. Questo procedimento non è complicato, ma può diventare tedioso.

Carbon14

Per automatizzare la raccolta di tutte le date, come spiegato sopra, ho sviluppato un tool in Python chiamato Carbon14. Lo strumento è stato pubblicato con licenza libera GPLv3 su GitHub, da dove si può effettuare il download e installare le relative dipendenze:

```
git clone https://github.com/Lazza/Carbon14.git  cd Carbon14  pip install -r requirements.txt
```

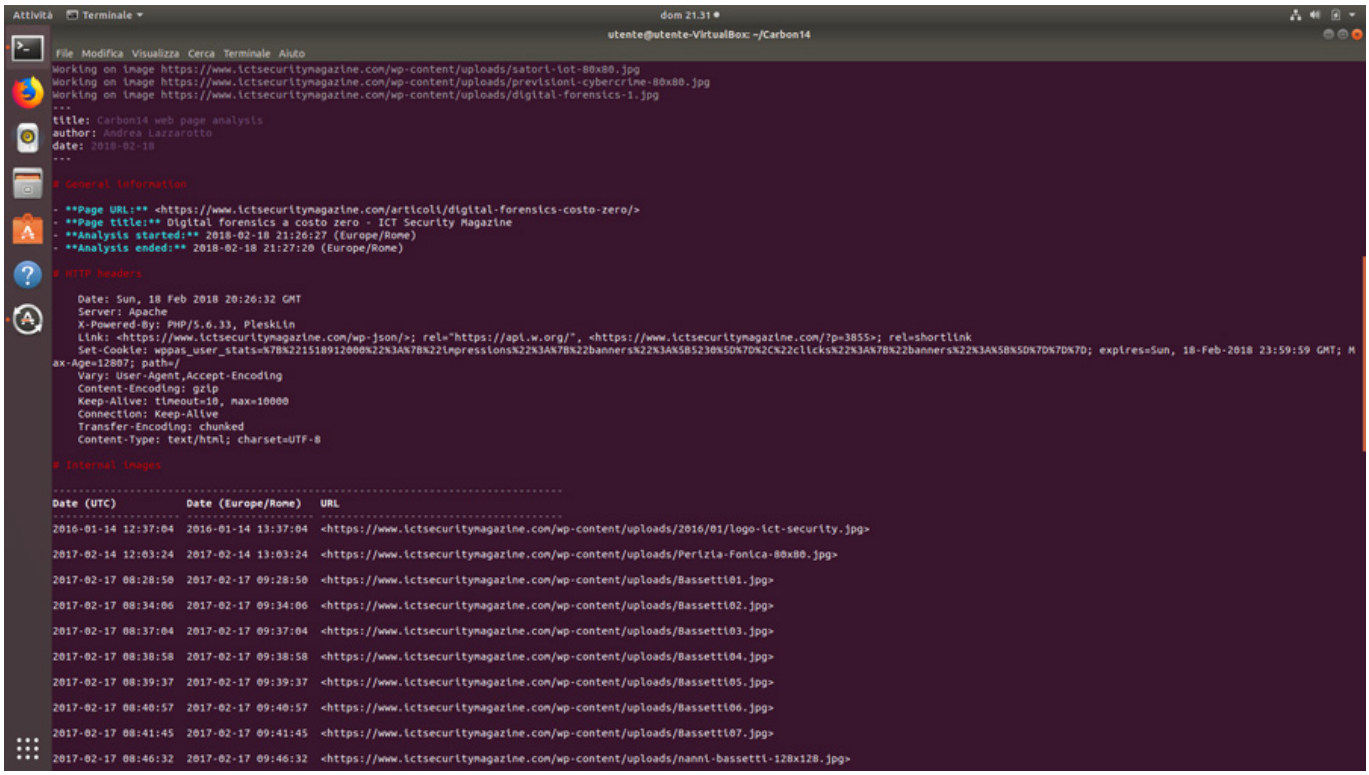
Chi lo desidera naturalmente può anche optare per l'uso di virtualenv.

L'utilizzo del programma è estremamente semplice. L'unico parametro obbligatorio dello script è l'URL che si vuole analizzare, mentre si può scegliere se indicare il nome dell'analista o meno. In caso affermativo, il nome sarà incluso nel report generato dal software.

Considerando sempre l'URL precedente, possiamo lanciare:

```
./carbon14.py 'https://www.ictsecuritymagazine.com/articoli/digital-forensics-costo-zero/' -a "Andrea Lazzarotto"
```

In output riceveremo **un report testuale in formato Markdown** con qualche tocco di colore per migliorarne la leggibilità.



Estratto dall’output di Carbon14

La visualizzazione colorata è comoda per avere una panoramica della situazione, ma la cosa migliore da fare è il redirect dell’output di Carbon14 su file in modo da poter conservare l’esito dell’analisi:

```
./carbon14.py 'https://www.ictsecuritymagazine.com/articoli/digital-forensics-costo-zero/' -a "Andrea Lazzarotto" > report.md
```

Il report generato contiene varie sezioni:

- Titolo, autore e data
- Informazioni generali sull’URL analizzato, l’ora di inizio e di fine analisi
- Header HTTP della pagina web
- Tabelle con i timestamp delle immagini

I valori relativi alle immagini sono indicati sia in UTC che nel fuso orario del computer su cui viene effettuata l’analisi. Le immagini sono suddivise in tre sezioni: interne, esterne e tutte insieme.

Valutazione dei dati estratti

Nel nostro esempio, consideriamo questo estratto dall’output di Carbon14:

```
------------------------------------------------------------------------
----------  Date (UTC)           Date (Europe/Rome)   URL  -----------
--------- ------------------- ------------------------------------
 2016-01-14 12:37:04  2016-01-14 13:37:04    2017-02-14 12:03:24  2017
-02-14 13:03:24    2017-02-17 08:28:50  2017-02-17 09:28:50    2017-02
-17 08:34:06  2017-02-17 09:34:06    2017-02-17 08:37:04  2017-02-17 0
9:37:04    2017-02-17 08:38:58  2017-02-17 09:38:58    2017-02-17 08:3
9:37  2017-02-17 09:39:37    2017-02-17 08:40:57  2017-02-17 09:40:57
   2017-02-17 08:41:45  2017-02-17 09:41:45    2017-02-17 08:46:32  20
17-02-17 09:46:32
```

Possiamo notare che la prima immagine non è pertinente, in quanto è il logo della rivista. La seconda invece è la miniatura di un altro post. Le successive immagini mostrano che chi ha lavorato all’articolo lo ha fatto il 17 febbraio 2017, a partire dalle 9:28 fino alle 9:46.

Non possiamo sapere con certezza se ci sono stati anche interventi precedenti o successivi,

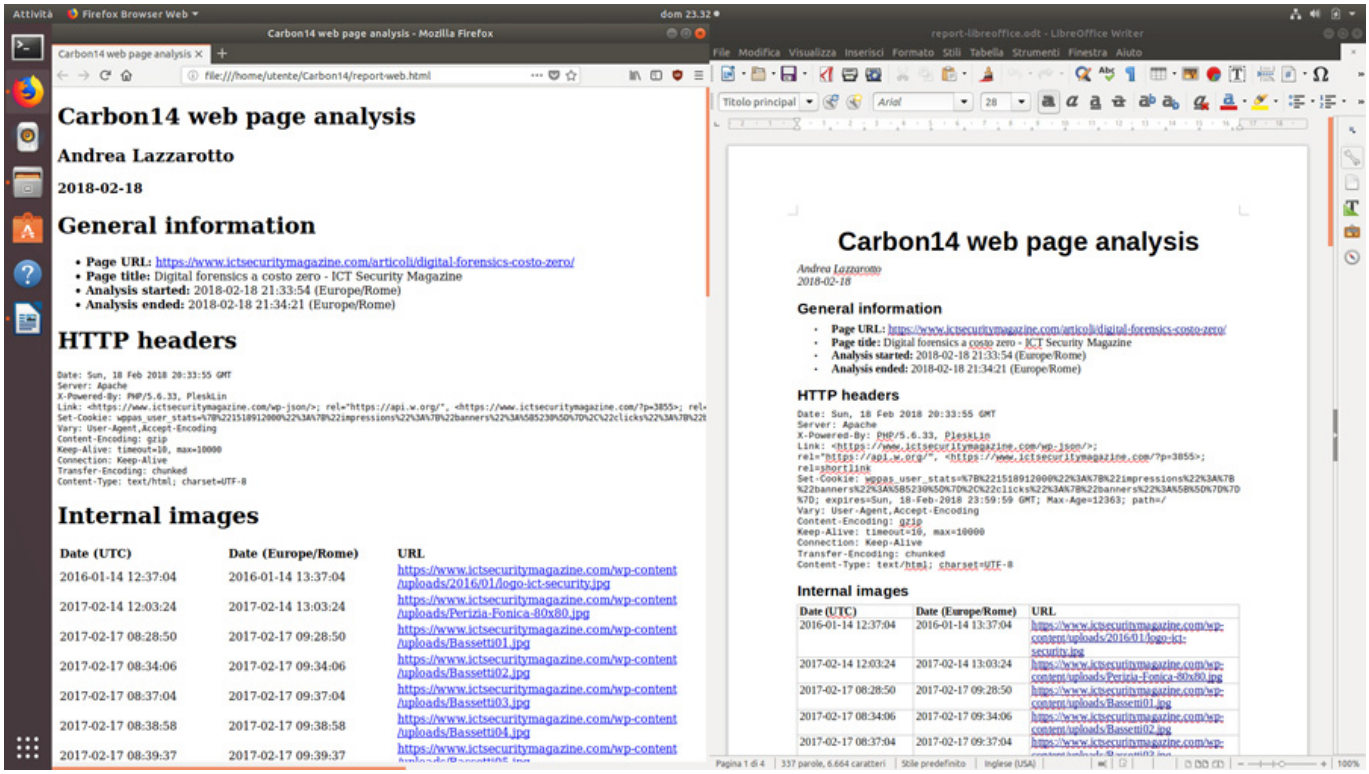
tuttavia possiamo dire che **sicuramente quei 18 minuti sono stati parte del processo di scrittura** dell'articolo all'interno del CMS del sito.

Conversione del report

Pur essendo direttamente utilizzabile come file di testo, il report in formato Markdown ha un ulteriore vantaggio: può essere convertito molto facilmente utilizzando Pandoc [3]. Per esempio, è possibile trasformare il report in HTML, ODT oppure DOCX:

```
pandoc -s report.md -o report-web.html  pandoc report.md -o report-libreoffice.odt  pandoc report.md -o report-msword.docx
```

Questo passaggio è del tutto opzionale, ma migliora la resa grafica dei dati presentati e consente di integrarli, o di modificare il report evidenziando specifiche righe di particolare interesse.



Report convertito in HTML e ODT

In futuro

Il funzionamento di Carbon14 è piuttosto semplice, infatti lo scopo principale del tool è quello di

automatizzare un compito potenzialmente noioso e ripetitivo. Non si tratta di uno strumento di *acquisizione* di pagine web, bensì si limita alla loro *datazione.*

Tuttavia, sarebbe interessante valutare eventuali soluzioni aggiuntive per dare maggior sostegno ai dati estratti. Purtroppo l'ipotesi di acquisire la pagina con i classici siti https://web.archive.org o https://archive.fo/ non è di grande aiuto. Entrambi gli strumenti non memorizzano gli header originali delle immagini, quindi non sarebbe possibile verificarli in un secondo momento.

Riferimenti

[1] Nanni Bassetti. Digital forensics a costo zero. *ICT Security Magazine,* 17/02/2017. URL https://www.ictsecuritymagazine.com/articoli/digital-forensics-costo-zero/

[2] Paolo Dal Checco. OSINT su siti web – chi si nasconde dietro quel sito? Presentazione, 23/05/2017. ISACA Rome Chapter. URL http://www.isacaroma.it/wp-content/uploads/2017/09/20170523-Paolo-Dal-Checco-OSINT-e-SITI-WEB.pdf#page=22

[3] John MacFarlane. Pandoc. URL https://pandoc.org/

[4] Jakub Rozto?il. HTTPie. URL https://httpie.org/

[5] Emil Protalinski. WordPress now powers 25% of the Web. *VentureBeat,* 08/12/2015. URL https://venturebeat.com/2015/11/08/wordpress-now-powers-25-of-the-web/

A cura di: **Andrea Lazzarotto**