Crawler

Valentina Ceoletta VR407794
Mattia Zanotti VR411086
Nicolò Zenari VR412613

26 settembre 2018

Introduzione

Il progetto consiste nell' esplorare il web con l'obiettivo di raccogliere codice JavaScript considerato malevolo. In generale i passi eseguiti dal programma sono:

1. collezione degli URL dalla comunitá di *HpHosts*;
2. download delle pagine web e delle relative risorse;
3. interrogazione di *Virus Total* e *Google Safe Browsing*.

File di configurazione

Il file *config.json* contiene tutte le informazioni necessarie per il corretto funzionamento del progetto.

getHosts

Lo script *getHosts* richiede e parserizza le pagine HTML di HpHosts. L'obiettivo viene raggiunto in due passi principali:

1. creazione delle thread per scaricare le pagine di *HpHosts*;
2. parserizzazione e raccolta delle informazioni necessarie.

Fase 1

Dato il cospicuo numero di pagine che solitamente deve essere scaricato, al fine di velocizzare tale processo si creano tante thread quante il numero di core della macchina host; ogni thread si occupa del download di specifiche pagine. Esse terminano se:

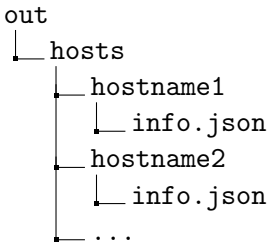
- terminano le pagine di loro competenza;
- in una pagina è presente un host già trattato;
- la data di pubblicazione è antecedente o seguente il periodo di interesse.

Fase 2

HpHosts organizza gli host all'interno di tabelle HTML. Da queste tabelle, lo script estrapola le informazioni relative ad ogni host e, se corrispondono ai criteri desiderati, vengono salvate all'interno di un file Json.
Le informazioni estrapolate sono *hostname*, *IP*, *classe*, *data di pubblicazione* su hpHosts e *data di aggiunta*.

Struttura dell directory

Alla fine dello script si genera la seguente struttura a cartelle (base per i successivi script):



Crawler

Si utilizza *Heritrix* nella versione 3.1.1. con la modifica del file di configurazione nei seguenti punti:

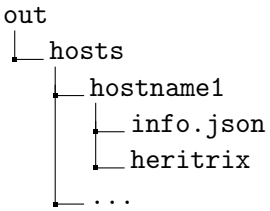
1. caricamento della lista di host da file *seeds.txt*;
2. applicazione di regole in cascata per accettazione delle pagine HTML e JS con profondità a 1 hops;
3. estrazione di pagine HTML,HTTP e JS;
4. salvataggio del crawling nel formato *.warc*.

warcExtractor

Per ogni host scaricato con successo:

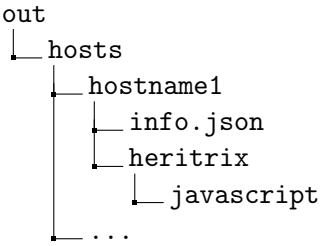
- si crea una directory *heritrix* contenente i file associati;
- si aggiorna il file *seeds.txt* per un successivo crawling;
- creazione dei file *bodyN.json* per il modulo GSB.

Nel caso di insuccesso, la cartella dell'host generato in 8 verrà rimossa.



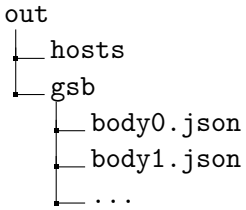
jsExtractor

Creazione di una directory in cui inserire il codice JavaScript presente nei file estratti dal crawler.



Google Safe Browsing

Utilizzando i file *bodyN.json* (organizzati come mostrato in figura), uno script Python interroga GSB.



Virus Total

Utilizzando le API pubbliche viene interrogato Virus Total che fornisce il risultato dell'analisi in due passi:

1. richiesta *post http* con la quale viene inviato l' URL da analizzare;
2. richiesta *get http* necessaria per ottenere il report dell'analisi.

Tempi di esecuzione

- **getHosts**: circa 7/10 secondi per pagina
- **Crawler**: al primo avvio molte ore se non giorni, tempo minore negli avvii successivi
- **warcExtractor** e **jsExtractor**: esecuzione nell'ordine di minuti
- **GSB**: esecuzione nell'ordine di qualche secondo
- **VT**: esecuzione nell'ordine di qualche ora

Statistiche

- **GSB**: 11/288 host taggati (periodo 13 - 31 agosto);
- **VT**: 51/288 host non malevoli (periodo 13 - 31 agosto);
- **eval**: su 288 host collezionati sono stati generati 4494 file JavaScript dei quali 5 presentano eval espliciti e 4489 impliciti.

Codice

Codice disponibile su GitHub: `https://github.com/nzenari/JScrawler.git`