Università del Molise

Dipartimento di Bioscienze e Territorio

Proposta di progetto

Identificazione Estensioni Malevove di Firefox

Author:
Daniele Albanese

Networking security and software security

Dicembre 10, 2020

Introduzione

Poiché gli utenti soddisfano sempre più le loro esigenze informatiche attraverso il *Web*, i *browser Web* moderni, devono fornire maggiori funzionalità e personalizzazione.
Una caratteristica indispensabile dei *browser* moderni è la possibilità di essere personalizzati, lato client, tramite delle *estensioni*.
Utilizzando le *estensioni*, gli utenti possono aumentare e modificare il comportamento dei loro browser per soddisfare le loro esigenze.
Una delle possibili esigenze degli utenti, per cui vengono utilizzate le *estensioni*, è quella di aumentare la propria produttività come, ad esempio: bloccando gli annunci e/o i tracciamenti indesiderati o offrendo nuovi modi per organizzare schede e segnalibri.
Dato l'aumento delle *estensioni* una volta benigne divenute maligne, nell'articolo preso in esame [1], viene proposto un nuovo metodo per il rilevamento delle *estensioni dannose* del *browser*, concentrandosi sui loro delta di aggiornamento.

Data un'estensione diventata *dannosa*, il loro sistema utilizza l'ultima versione benigna di tale estensione, per identificare il codice responsabile delle sue azioni *malevole*. Concentrandosi sulle API abusate da questo codice-delta, il sistema crea una sequenza di API, che verrà ricercata in tutte le altre estensioni presenti all'interno dello Store.
In questo modo, il sistema utilizza le estensioni dannose precedentemente trovate e le categorizza come estensioni *"seme"*, che verrano poi utilizzate per identificare estensioni con aggiornamenti simili, quindi, che utilizzano API potenzialmente dannose, che non sono state ancora rilevate dal sistema o contrassegnate dagli utenti.

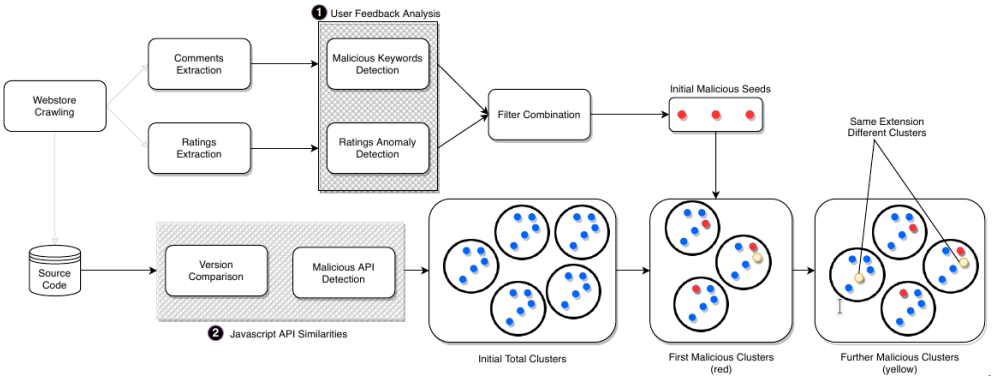
Fatta questa premessa, andremo ora ad analizzare, come verrà rapportato, il sistema descritto in precedenza, avendo come browser di analisi, non più il browser **Google Chrome**, ma il browser **Mozilla Firefox**.

Il *browser Firefox* utilizza **Firefox Browser ADD-ONS** [2] come repository ufficiale per la pubblicazione e la distribuzione delle estensioni agli utenti.

1.1 Codice sorgente

Le estensioni sono distribuite nel negozio sotto forsa di file *.xpi* ovvero, un semplice archivio ZIP con un'estensione speciale. All'interno di questo file **XPI** risiedono tutti i file dell'estensione ovvero,

FIGURA 1.1: Data sources collection and workflow of malicious extension detection pipeline. Analysus from User Feedback (**1**) and malisious JavaScript clustering (**2**) from seed extensions



il *codice sorgente* (JavaScript, HTML e CSS), le *immagini locali*, ed un file *manifest.json*. In questo file *manifest.json*, sono contenuti tutti i metadati dell'estensione in formato **JSON**, nello specifico, è presente il nome, la versione, la descrizione dell'estensione, nonché le autorizzazioni richieste.

Le due principali categorie di script presenti all'interno delle estensioni sono: i *background script* ed i *content script*. Il primo è uno script eseguito durante l'attività dell'estensione, responsabile della maggior parte delle funzionalità in backgorund. Possono esserci più *background script* ma, nella maggior parte delle estensioni, ne è presente soltanto uno. Il secondo tipo invece, è un file **JS** in esecuzione nel contesto della pagina web visitata e che utilizza i **DOM** (**D**ocument **O**bject**M**odel) per modificare le pagine web. Per questo tipo di scritp, né è presente più di uno. Oltre queste due principali categorie di script, potrebbe essere presente del codice **JS** di supporto, come, ad esempio, librerie di terze parti.

1.2 Commenti e voti

Oltre a raccogliere il codice sorgente delle estensioni, il sistesta raccoglierà altri dati disponibili sullo **Store**. Per ogni estensione attiva sullo **Store**, verrà eseguita una scansione di tutte le informazioni disponibili nella sua pagine, compreso il numero totale di valutazioni, la valutazione media totale, i download totali e l'autore dell'estensione. Inoltre, verranno raccolti tutti i commenti che, gli utenti hanno scritto per ciascuna estensione e per ogni commento verrà raccolta la *valutazione*, il *giorno di pubblicazione* ed il *nome dell'autore*.

Metodologia

Il sitema di analisi delle estensioni consisterà in due fasi principali: utilizzare i feedback degli utenti e, raggruppare il codice sorgente delle estesioni in base alle API JavaScript.

2.1 Fase 1

La logica di questa prima fase è basata sui feedback degli utenti esperti, che osservano un'estensione precedentemente *benigna*, comportarsi in modo *malizioso* e, non solo la disinstalleranno, ma almeno alcuni di loro, lasceranno un feedback negativo attraverso il sistema di recensioni. Questo feedback avrà lo scopo di allarme per altri utenti che potrebbero prendere in considerazione l'installazione dell'estensione in questione.

Per identificare quanti commenti saranno necessari per identificare le anomalie di rating, verranno eseguiti una serie di esperimenti utilizzando il pacchetto statico *Anomalize* [3]. *Anomalize* può essere utilizzato per identificare tendenze e componenti stagionali in serie temporali [4]. Come le tipiche tecniche di rilevamento delle anomalie, questo processo comprende due fasi, la *fase di addestramento* e la *fase di test/rilevalento*. Nella *fase di addestramento*, verrà utilizzata la parte iniziale della sequenza di valutazioni, per impostare una verità di base per le valutazioni tipiche, che una data estensione riceve. Quindi verrà utilizzato il resto dei dati per trovare anomalie nelle valutazioni.

2.2 Fase 2

Nella *fase 2*, verranno utilizzate le estensioni contrassegnate come dannose dalla *fase 1*, e verrà identificato l'aggiornamento del codice, che corrisponde all'estensione che passa da benigna a maligna. Verà codificato questo aggiornamento in termini di API critiche, e verrà cercato per altre estensioni con aggiornamenti simili. Attraverso questo processo, verranno identificate altre estensioni, che mostrano segni simili di aggiornamenti dannosi, ma che ancora non sono state contrassegnate, né dagli utenti né dallo **Store**.

Delivery

Per quanto riguarda la fase di *"Delivery"* verrà prodotto un documento con i risultati dello studio e tutto il codice sarà disponibile sulla piattaforma GitHub.

Bibliografia

[1] Alexandros Kapravelos Nikolaos Pantelaios Nick Nikiforakis. «You've Changed: Detecting Malicious Browser Extensionsthrough their Update Deltas». In: (2020). DOI: https://www.kapravelos.com/publications/extensiondeltas-CCS20.pdf.

[2] *Firefox ADD-ONS Store*. https://addons.mozilla.org/it/firefox/.

[3] *Anomalize package in R*. https://cran.r-project.org/web/packages/anomalize/index.html.

[4] Sudhir R Paul e Karen Y Fung. «A generalized extreme studentized residuals multiple-outlier-detection procedure in linear regression». In: (1991).