UNIVERSITY OF MOLISE

DEPARTEMENT OF BIOSCIENCE AND TERRITORY

PROJECT PROPOSAL

Cryptojacking attack and defence based on Graphics Processing Units

---

Author:
Federico ZAPPONE

Networking security and software security
December 01, 2020

Indice

Elenco delle figure

Introduzione

In quest'ultimo decennio si è molto discusso di monete digitali e delle così dette criptovalute, da qualche anno infatti sulla bocca di tutti risaltano parole di innovazione e opportunità in seguito all'avvento della nuova tecnologia blockchain basata sulla logica di database distribuito. Proprio grazie a quest'ultima è nata la prima criptovaluta al mondo, più precisamente, il 3 gennaio 2009 veniva alla luce il blocco genesi di Bitcoin composto allora da 50 *BTC* dal valore complessivo di neppure un dollaro statunitense dato il prezzo iniziale di $0.0008 per Bitcoin. In seguito all'ottimo riscontro in molteplici campi il 6 novembre 2010 un Bitcoin si presentava con un valore di $0,50, in meno di due anni il prezzo era aumentato di 625 volte, e da allora in poco più di sette anni, Bitcoin raggiunse il suo massimo storico di quasi $20.000 ovvero circa 40.000 volte di più. Divenuto un caso più unico che raro, Bitcoin si è posto da apripista a più di 5000 altre criptovalute fino ad essere definito come l'oro del XXI secolo. Questa definizione non è dovuta solo all'incredibile aumento del prezzo di Bitcoin negli ultimi anni ma anche ad una delle caratteristiche chiave che l'oro e la maggior parte delle criptovalute condivide, il processo di "estrazione" definito appunto come *mining* nel campo delle monete digitali.
Il *mining* di criptovalute consiste nel creare monete virtuali attraverso la risoluzione di alcune funzioni crittografiche necessarie per la validazione delle transazioni e dei blocchi che compongono una blockchain. Questa operazione termina con un compenso da parte della blockchain al miner, il quale avendo offerto la sua potenza di calcolo per la validazione del blocco viene premiato con la stessa criptovaluta minata. Questi calcoli vengono eseguiti da sistemi informatici dedicati a questo specifico processo, essi sono divisi in due grandi macro sezioni: quelli che sfruttano la Central Processing Unit (CPU) e quelli che sfruttano invece la Graphics processing unit (GPU). I primi prendono il nome di *ASIC* acronimo di *Application Specific Integrated Circuit* ovvero circuiti costruiti per la risoluzione di un calcolo ben specifico che risultano però molto inefficienti su altri tipi di algoritmi. I sistemi basati su GPU sono invece molto più prestanti grazie proprio al fatto che le schede video riescono ad effettuare più calcoli al secondo rispetto alle CPU che risultano quindi meno redditizie nella maggior parte dei casi. D'altro canto per le GPU risulta essere molto più difficile la gestione delle temperature e inoltre comportano costi maggiori sia in termini di assemblaggio che di costi energetici per il mantenimento. Proprio questi ultimi aspetti sono quelli di notevole impatto quando si parla di mining, il costo di acquisto delle componenti è infatti nettamente aumentato negli ultimi anni, questo sia a causa delle nuove scoperte tecnologiche più performanti, sia proprio grazie alle grandi farm di mining sparse per il mondo che costantemente acquistano sempre nuovi componenti per ingrandire i propri centri di estrazione. Il costo energetico che questi sistemi comportano ha portato invece i grandi centri di mining a svilupparsi maggiormente nei paesi con costi dell'elettricità più bassi e allo stesso tempo ha reso molto più impegnativo l'operazione di mining per chi volesse entrarne a far parte.
È così che con l'aumentare dei costi, e allo stesso tempo delle opportunità di guadagno offerte dal

mondo delle blockchain sempre più in crescita, le criptovalute hanno iniziato a risaltare agli occhi del crimine informatico. Più precisamente la tecnologia blockchain, e di conseguenza le criptovalute, posseggono, per lo meno la maggior parte di esse, una caratteristica molto importante per i cyber criminali ovvero garantiscono una certa forma di anonimato. Le criptovalute infatti soffrono dello stesso problema del contante che comunemente si utilizza, non sono infatti in alcun modo collegabili ad un'entità specifica, l'unico ad averne pieno controllo è colui che le possiede. Questa caratteristica ha portato a un'enorme utilizzo delle criptovalute per lo svolgimento di azioni illecite attraverso il web, nel 2017 infatti si è riscontrato un netto aumento dei *ransomware*, virus che bloccano in qualche modo il sistema al quale accedono con l'intento di ottenere un riscatto per il suo sblocco. L'incremento di questi attacchi è dovuto infatti principalmente alla pratica dei cyber criminali di richiedere il riscatto sotto forma di criptovalute così da risultare irrintracciabili. Successivamente si è arrivati a una forma di attacco più intelligente e meno invasiva che utilizza i *cryptominers*. Definito appunto come "L'anno dei cryptominers", il 2018 vede la stessa impennata di casi di *ransomware* dell'anno precedente in una nuova tipologia di attacco che si appropria sì delle criptovalute in modo illegale ma sfruttando un'operazione lecita come quella del mining. Questa tecnica detta *cryptojacking* consiste in virus che si insidiano all'interno del computer della vittima e utilizzano la sua potenza computazionale del per l'estrazione di criptovalute, ma diversamente dal mining legittimo, il guadagno di questo processo sarà poi attribuito non ai possessori del calcolatore bensì all'attaccante. Il tutto avviene seguendo un basso profilo tramite programmi infetti che vengono installati sul computer o, come più recentemente si è affermato, utilizzando script malevoli presenti all'interno di pagine web. La sostanziale differenza tra le due tipologie di *cryptominers* sta principalmente nella loro rintracciabilità, infatti quelli che operano attraverso il web sono più difficili da individuare e analizzare rispetto a programmi che vengono installati e che sono quindi sotto l'occhio diretto di antivirus e delle politiche dettate dal sistema operativo.
L'ultima ma sostanziale differenziazione che va fatta quando si parla di *cryptojacking* in ambito web risiede nelle componenti del calcolatore che si va ad utilizzare: appurato che le GPU offrono dei ricavi maggiori in termini di *mining* rispetto alle CPU, ma di contro queste non sono facilmente utilizzabili attraverso il web. Ciò ha portato infatti ad uno sviluppo maggiore dei *cryptominers* basati su CPU e di conseguenza a sistemi di difesa che si concentrino su di esse, mentre si sono trascurate le minacce derivanti da attacchi più sofisticati che puntano invece alla Graphics Processing Unit.

Background e stato dell'arte

Come fatto notare in precedenza lo stato dell'arte si è mosso principalmente verso la l'aspetto riproposto più frequentemente ovvero il *cryptojacking* basato su CPU. In generale sia Musch et al. [1] che Saad, Khormali e Mohaisen [2] mostrano la diffusione di *cryptominers* nei siti web, analizzando l'efficacia di tecniche di blacklisting che risultano però essere una protezione insufficiente e poco pratica. Wang et al. [3] forniscono un metodo di analisi di firme basate su istruzioni della CPU durante l'esecuzione dei moduli *WebAssembly*. Konoth et al. [4] introducono invece *MineSweeper* basato anch'esso sul precedente principio di firme ma aggiunge inoltre il rilevamento di eccessive chiamate di sistema di natura crittografica durante l'esecuzione di un programma. Kharraz et al. [5] dimostrano invece che come l'utilizzo della CPU generi una grande quantità di falsi positivi da sola all'interno della navigazione web. Differentemente da quelli precedentemente citati, Tahir et al. [6] presentano *MineGuard*, un sistema che rileva tramite l'analisi delle prestazioni hardware associate agli algoritmi di mining i *cryptominers*. Il sistema monitora costantemente sia CPU che GPU ed inoltre offre un'ottima soluzione sia in termini di efficienza che di utilizzo di risorse per la sua esecuzione. Belkin, Gelernter e Cidon [7] offrono infine una soluzione più pratica in ambito web, dedicata esclusivamente a *cryptominers* di GPU che utilizzano la libreria grafica *WebGL*.

Scopo ed obiettivi

Obiettivi primari del progetto:

(i) Sviluppo di un *cryptominer* basato sull'utilizzo di Graphics Processing Units.

(ii) Sviluppo di un sistema per l'inserimento del *cryptominer* all'interno di pagine web.

Obiettivi secondari del progetto:

(iii) Identificazione e sviluppo di una modalità di difesa contro l'attacco.

Lo scopo ultimo del progetto è quello di sviluppare un attacco basato sulla tecnica del *cryptojacking* che operi su GPU e partendo da questo sviluppare una difesa per attacchi simili.
Per il raggiungimento di tale scopo è innanzitutto necessario (i) sviluppare un programma in ambito web che effettui l'operazione di mining attraverso l'utilizzo della Graphics Processing Unit e testarne le funzionalità. Successivamente (ii) creare un sistema che permetta al codice malevolo di raggiungere le vittime attraverso pagine web e testarne l'efficacia in un ambiente simile a quello reale. Il raggiungimento del terzo e ultimo obiettivo è strettamente legato ai due punti principali, infatti una volta sviluppata e testata l'efficienza del sistema di attacco, sarà poi possibile (iii) analizzarlo e identificare una modalità di difesa efficiente e, se possibile, utilizzarla per implementare un sistema di difesa contro attacchi dello stesso genere.
Per il raggiungimento degli obiettivi primari si è pensato all'utilizzo combinato di librerie che permettono l'interazione con la GPU in ambito web e quelle tipologie di attacco che permettono l'iniezione di codice malevolo all'interno di applicativi web.
Si è quindi preso in considerazione l'utilizzo di alcune librerie basate su *Open Graphics Library* conosciuta maggiormente come *OpenGL* [8]. Quest'ultima è una specifica che definisce delle *API* per applicazioni che operano in ambienti 3D su molteplici piattaforme e tramite diversi linguaggi di programmazione. In particolare è divenuta ormai lo standard per quanto riguarda la grafica tridimensionale in sistemi operativi *Unix-like*, grazie anche al fatto che è stata pensata per architetture altamente parallelizzabili e potenti come le GPU. Su questo standard sono basate le librerie *WebGL* [9] e *GPU.js* [10], più precisamente *WebGL* è una *Web-based Graphics Library* ovvero una libreria pensata per la gestione di elementi grafici all'interno del web, essa fornisce delle *API* per grafica 3D in un contesto *HTML5* tramite l'utilizzo del *Document Object Model (DOM)*. In aggiunta si è pensato anche all'utilizzo di *GPU.js* ovvero una libreria scritta interamente in *JavaScript* pensata per sfruttare l'accelerazione hardware delle GPU in ambito web utilizzando a sua volta *WebGL*.
Infatti, per poter raggiungere le vittime ed effettuare l'operazione di mining tramite pagine *HTML*, è necessario che lo script creato sia presente nella struttura della pagina. Per fare ciò si potrebbe

creare un'applicazione web con il codice malevolo ma questa non sembra essere la soluzione migliore in quanto, oltre a raggiungere un numero molto limitato di vittime, aumenterebbe il rischio di essere scoperti nel caso si stesse effettuando un attacco reale. Proprio per rendere l'attacco il più veritiero possibile, si è pensato invece di inserire in qualche modo il codice malevolo all'interno di siti web vulnerabili, il modo che è stato individuato è quindi quello di sfruttare attacchi di tipo *XSS*.
Sia *WebGL* che *GPU.js* sono state ideate al fine di computare più velocemente della sola CPU il caricamento di elementi grafici molto complessi, ma operando in un contesto di grafica tridimensionale permettono quindi di effettuare operazioni anche complesse attraverso la GPU dell'utente che visualizza la pagina web dove sono utilizzate. L'idea è quindi quella di sviluppare un *cryptominer* attraverso l'utilizzo di queste librerie e del linguaggio *Javascript*, la scelta di questo linguaggio è dovuta non solo al fatto che è uno dei più utilizzati e supportati all'interno del web, ma anche al fatto che si presta benissimo per l'iniezione di codice malevolo.
Gli attacchi di tipo *Cross Site Script (XSS)* sono attacchi eseguiti attraverso linguaggi di scripting e sono tra i più diffusi nel web, infatti secondo il report di Positive Technologies Security [11], la percentuale di questi attacchi è salita dall'occupare il 77,9% nel 2017 all'88,5% nel 2018 di tutte le tipologie di attacchi web registrate. Secondo Precise Security [12] invece, in rapporto a tutti i tipi di attacchi sferrati verso le grandi compagnie di Europa e Nord America nel 2019, la percentuale di attacchi *Cross Site Script* risulta essere del 39%, superando più del doppio la percentuale degli attacchi di *SQL Injection*.
L'*XSS* consiste nell'introdurre del codice arbitrario lato client all'interno dei siti web con il fine di eseguire una serie di attacchi rivolti ad altri utenti o anche all'amministratore del sito stesso. Questa vulnerabilità affligge i siti dinamici che non effettuano un controllo sugli input lato client, il che porta ad una "fusione" tra il payload, ovvero il codice inserito dall'attaccante, e il codice reale della pagina. Esistono tre tipi di attacchi *Cross Site Script*:

- **XSS reflected:** il payload viene eseguito solo nel momento dell'iniezione;
- **XSS stored:** il payload viene salvato all'interno della struttura dell'applicativo ed eseguito ad ogni accesso del contenuto;
- **XSS dom-based:** il codice sorgente e la risposta del server non vengono modificate, il payload viene eseguito a runtime senza inoltrare richieste al server ma utilizzando il codice già presente nella pagina.

Esistono vari strumenti in grado di analizzare la struttura delle pagine web e di identificare vulnerabilità *XSS*, alcuni di questi sono *XSStrike* [13], *Traxss* [14] e *XSSer* [15]. Questi sono tutti tool open-source di analisi molto validi sviluppati in *Python*, il più interessante è però *XSSer*, uno dei tool preinstallati presenti nelle distribuzioni Linux atte al *penetration testing*. Si è quindi pensato di utilizzare uno di questi tool per individuare in modo semi-automatico/automatico le vulnerabilità *XSS* all'interno di siti web, una volta individuati una serie di siti vulnerabili si creerà un ambiente di lavoro dove effettuare i test di *injection*. L'ambiente di test dell'attacco sarà infatti composto da copie dell'intera struttura dei siti vulnerabili identificati, questo per rendere i test il più veritieri possibile.
Una volta testata l'efficacia del sistema di attacco sarà effettuata un'analisi per individuare un modo per difendere gli utenti degli applicativi web dal *cryptominer*. Una possibile idea sarebbe quella di sviluppare un'estensione per il browser che individui la presenza di librerie web che utilizzano la GPU e avvertire di ciò l'utente che sta accedendo alla pagina. In questo modo si potrebbe evitare

che l'utente acceda al contenuto della pagina sospetta o che dopo una verifica scelga se disabilitare o meno alcune funzionalità della pagina. Non saranno effettuate analisi dal punto di vista del *Cross Site Script* in quanto modi per evitare questo tipo di vulnerabilità sono già stati studiati ed analizzati a fondo come ad esempio è stato fatto da Bisht e Venkatakrishnan [16].
Lo sviluppo dei sistemi avverrà con l'ausilio di una macchina virtuale, questa scelta non è dovuta tanto allo sviluppo dei sistemi di attacco o di difesa in sé ma al fatto che non si è a conoscenza delle operazioni effettuate dai siti vulnerabili che si andranno a riproporre in copia locale. Il progetto sarà reso pubblico sulla piattaforma *GitHub* all'indirizzo `https://github.com/ZappaBoy/hi-jacket` nel quale saranno presenti (i) lo script del *cryptominer* iniettabile, (ii) il sistema di *injection* dello script semi-automatico/automatico che sfrutti le vulnerabilità *XSS* e (iii) un possibile metodo di difesa contro l'attacco sviluppato.

Bibliografia

[1] Marius Musch et al. «Web-based Cryptojacking in the Wild». In: *arXiv preprint arXiv:1808.09474* (2018).

[2] Muhammad Saad, Aminollah Khormali e Aziz Mohaisen. «End-to-end analysis of in-browser cryptojacking». In: *arXiv preprint arXiv:1809.02152* (2018).

[3] Wenhao Wang et al. «Seismic: Secure in-lined script monitors for interrupting cryptojacks». In: *European Symposium on Research in Computer Security*. Springer. 2018, pp. 122–142.

[4] Radhesh Krishnan Konoth et al. «Minesweeper: An in-depth look into drive-by cryptocurrency mining and its defense». In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 2018, pp. 1714–1730.

[5] Amin Kharraz et al. «Outguard: Detecting in-browser covert cryptocurrency mining in the wild». In: *The World Wide Web Conference*. 2019, pp. 840–852.

[6] Rashid Tahir et al. «Mining on someone else's dime: Mitigating covert mining operations in clouds and enterprises». In: *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer. 2017, pp. 287–310.

[7] Alex Belkin, Nethanel Gelernter e Israel Cidon. «The Risks of WebGL: Analysis, Evaluation and Detection». In: *European Symposium on Research in Computer Security*. Springer. 2019, pp. 545–564.

[8] The Khronos Group Inc. *OpenGL*. https://www.opengl.org/.

[9] The Khronos Group Inc. *WebGL*. https://github.com/KhronosGroup/WebGL.

[10] gpujs. *GPU.js*. https://github.com/gpujs/gpu.js.

[11] Positive Technologies Security. *Penetration testing of corporate information systems: statistics and findings, 2019*. https://www.ptsecurity.com/ww-en/analytics/corp-vulnerabilities-2019/.

[12] Precise Security. *Cross-Site Scripting (XSS) Makes Nearly 40% of All Cyber Attacks in 2019*. https://www.precisesecurity.com/articles/cross-site-scripting-xss-makes-nearly-40-of-all-cyber-attacks-in-2019/.

[13] s0md3v. *XSStrike*. https://github.com/s0md3v/XSStrike.

[14] M4cs. *Traxss*. https://github.com/M4cs/traxss.

[15] epsylon. *XSSer*. https://github.com/epsylon/xsser.

[16] Prithvi Bisht e VN Venkatakrishnan. «XSS-GUARD: precise dynamic prevention of cross-site scripting attacks». In: *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer. 2008, pp. 23–43.