UNA MIRADA MÁS MODESTA AL CRYPTOJACKING

Steven Bernal Tovar Ivan Dario Chacon Uribe Juan Camilo Cubillos Ayala

steven.bernal@correo.icesi.edu.co ivan.chacon@correo.icesi.edu.co juan.cubillos@correo.icesi.edu.co

***Abstract*—En la actualidad, el cryptojacking en el navegador es el ataque con más auge entre los ciberdelincuentes, ya que, con solo insertar un script en un sitio web, el hacker puede secuestrar la potencia de la CPU de todos los visitantes. Este script se ejecutará desde el lado del cliente sin consentimiento, utilizando los recursos del host de la víctima para realizar cálculos computacionales con el fin de obtener una criptodivisa, la cual se le pagará al atacante. Esta situación ha provocado diversas perdidas a las víctimas, y es por esto que, algunos navegadores bloquean a partir de listas negras las actividades mineras maliciosas desde el nivel de protocolo de red, pero no tienen la capacidad de detectar muestras mineras por sí mismos. En este documento describiremos que es cryptojacking en navegador y el modelo propuesto en el documento A Novel Approach for Detecting Browser-based Silent Miner, para mitigar sus acciones. También explicaremos como utilizamos el script ofrecido por Coinhive para simular un ataque, con el fin de obtener una captura de tráfico de red con un sniffer, para poder detectar un patrón de un minero a través de las tramas de una red.**

Keywords

I. INTRODUCCIÓN

El sistema descentralizado de la cadena blockchain ha facilitado las transacciones de criptodivisas, permitiendo un almacenamiento transparente y distribuido [1]. Para prevenir el abuso y mejorar la confiabilidad en las transacciones, se implementan mecanismos de prueba como el Proof-of-Work (PoW) y el Proof of Stake (PoS) en la extracción de criptomonedas. Con Pow, los mineros compiten entre ellos para extraer una criptomoneda realizando extensas operaciones hash para resolver complejos problemas matemáticos, que luego son verificados por nodos distribuidos en una red peer-to-peer (P2P) [2].

Cuando un minero soluciona un problema para extraer una criptomoneda, la cual usa PoW, la complejidad de los problemas matemático se incrementa exponencialmente y a su vez los mineros tienen que incrementar el potencial de la CPU que usan. Esto ha dado lugar a que los atacantes utilices diversas técnicas para abusar de los recursos de otros dispositivos con fines mineros.

Entre las técnicas usadas, se encuentra el cryptojacking en navegador, la cual implica el secuestro de los recursos de la CPU de un host, para la externalización de los cálculos hash en transacciones basadas en PoW [2]. Estas transacciones se realizan sin el consentimiento del propietario del host.

El ataque minero al usar un determinado número de recursos de un host debe generar un rastro en el tráfico de red. Uno de los propósitos de este trabajo es tratar de identificar un patrón usando un sniffer. La motivación de este trabajo es encontrar un indicio en el tráfico de la red para poder lograr identificar a un minero.

II. FUNCIONAMIENTO

El cryptojacking realiza una inyección de un código JavaScript en un sitio web, lo que le permite secuestrar la capacidad de procesamiento del dispositivo de un visitante para explotar una criptomoneda específica [2]. Generalmente, JavaScript se ejecuta automáticamente cuando se carga un sitio web. Al visitar un sitio web con código minero, el host del visitante inicia una actividad minera, al convertirse en parte de un grupo de minería sin saberlo. Una característica del cryptojacking y que lo vuelve tan peligroso se puede ejecutar en cualquier plataforma.

Los hosts secuestrados realizan el cálculo de un bloque válido como resultado de la generación de nuevas monedas en el sistema. Sin embargo, el cálculo de un bloque válido es un proceso no trivial en el que los hosts tienen que resolver problemas matemáticos y proporcionar un PoW para sus soluciones, consumiendo una gran cantidad de recursos.

III. Entidades perpetuadoras del cryptojacking

El cryptojacking presenta variedad de entidades que pueden inyectar scripts de minería en la base de código del sitio web . Los resumimos aquí.

A. Webmaster

Un administrador de sitio web puede añadir un script de minería a su página web, con o sin informar a los usuarios.

B. Servios de terceros

Los terceros como los dueños de la publicidad o extensiones pueden inyectar scripts de cryptojacking en los sitios que los utilizan, ya sea intencionalmente o como resultado de una violación.

C. Extensiones de navegador

Las extensiones pueden insertar el script directamente en el navegador. Esta práctica es más peligrosa, ya que con solo ejecutar el navegador empezara a minar el computador

D. Hombre en el medio:

La web de un usuario, traffic, es a menudo enrutada a través de intermediarios que pueden tener acceso al contenido en texto plano. Por ejemplo, los proveedores de servicios de Internet o los enrutadores inalámbricos públicos gratuitos pueden inyectar scripts de cryptojacking en sitios que no sean HTTPS [3].

IV. MODELO PROPUESTO POR EL DOCUMENTO A NOVEL APPROACH FOR DETECTING BROWSER-BASED SILENT MINER

Para el proceso de verificación de la funcionalidad y el desempeño del prototipo BMDetector, los autores seleccionaron diferentes tipos de scripts, mirando el nivel de reconocimiento de los scripts para la prueba funcional y el impacto en la experiencia de usuario para la prueba de desempeño.

A. *Ambiente experimental*

Se utilizó para el cliente un sistema con Windows 7, 4GB de memoria, una tarjeta gráfica Intel GMA HD 4000 y un navegador Chrome personalizado. Para el servidor se utilizo un sistema con Ubuntu 16.04, 8GB de memoria y una tarjeta NVIDIA GeForce GTX 970. En el servidor se desplego el sistema activo de detección y la base de datos con las funcionalidades mineras.

B. *Proceso experimental y análisis*

Se incluyeron 115 páginas de inicio de sitios infectados con mineros y sitios normales, de las cuales se obtuvieron un total de 1159 muestras de scripts, de los cuales 236 eran mineros y de estos 59 se encontraban encriptados y ofuscados. A estas muestras se le realizo el análisis, calculando las observaciones reales y las inferencias teóricas. Estas muestras además se sometieron al análisis de la RNN, obteniendo una mayor exactitud que el set de entrenamiento. De estos análisis se pudieron obtener las 10 funcionalidades más frecuentes en los scripts, diferenciados entre los normales y los mineros. Como se puede apreciar en la siguiente tabla:

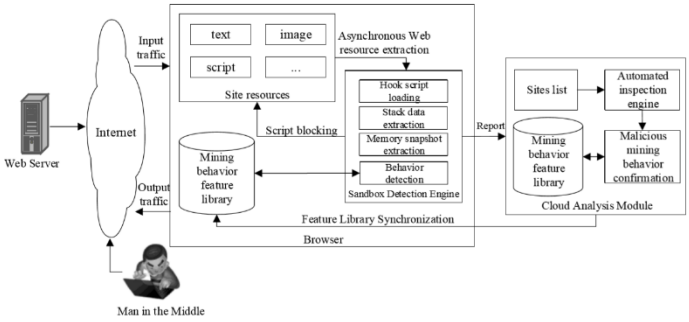
TABLE III. TOP 10 API FEATURES OF NORMAL SCRIPTS AND MINING SCRIPTS

ID	Normal Scripts Features	Frequency (%)	Mining Scripts Features	Frequency (%)
1	data	65.37	onReceiveMsg	58.51
2	value	62.48	protocal	56.23
3	test	61.74	connect	52.39
4	text	58.83	anonymous	49.85
5	prototype	55.31	verify	47.16
6	display	54.68	setJob	44.72
7	show	53.52	callSite	43.67
8	getElementById	52.95	encodeURI	41.34
9	name	51.79	random	39.46
10	referrer	50.87	addEventListener	37.42

Los autores mencionan que como se puede observar, las funcionalidades asociadas a los scripts mineros tienen que ver con temas de conexión y las de los scripts normales usualmente con temas de HTML.

C. *Explicación del proceso*

El modelo introduce JavaScript en la fuente del Kernel de Chrome Webkit, donde conduce las muestras de minería maliciosa conocidas, analiza las características de la estructura de datos desde la instantánea de la pila de datos del navegador y los datos de la pila después de la ejecución del script utilizando las funciones de la capa de análisis de la clave Hook del navegador, extrae las características de comportamiento dinámico de los mineros maliciosos y realiza la detección de forma automática sobre la base de RNN. La instantánea de la memoria y la información de la pila después de que el script haya sido analizado y ejecutado se utilizan para restaurar dinámicamente el código ofuscado y encriptado y obtener las características del código después de que el navegador se ejecute [4].



D. *Test funcional*

Para probar la funcionalidad del BMDetector, se enfocaron principalmente en tres tipos de objetos:

- Scripts originales: Sin alteraciones, obtenidos directamente de los sitios comerciales como CoinHive.
- Scripts variantes: Encriptados o ofuscados, basados en los scripts originales, con las mismas funcionalidades.
- Scripts homólogos: Utilizan los protocolos de los sitios web, las direcciones IP y los puertos para comunicarse.

El proceso de entrenamiento selecciona inicialmente 200 instancias de las muestras maliciosas y extrae 200 instancias de la pagina de inicio, que son confirmadas manualmente como muestras normales. Con estas 400 instancias, se crea el clasificador inicial. Con este clasificador inicial se clasifican el resto de las muestras y se realiza un proceso de aprendizaje incremental. Al final de este proceso, los autores nos refieren los siguientes resultados:

TABLE IV. RECOGNITION EFFECT OF BROWSER SILENT MINING SCRIPT

Category	Precision（%）	Recall（%）	F1（%）
original	97.92	98.72	98.32
encrypted	94.56	95.87	95.21
confused	90.47	88.35	89.40
encrypted and confused	87.73	89.85	88.78
homology	94.54	93.21	93.87

Donde se puede apreciar que la precisión y la exhaustividad de la prueba es relativamente alta, en especial con los scripts originales.

E. *Test de desempeño*

Dado que el uso del BMDetector requiere cargar los scripts y el contexto en el sandbox del browser mientras se carga la página, se podrían generar retardos en el tiempo de carga de la página o mayor uso del CPU y la memoria. Por esta razón, los autores deciden evaluar el uso del CPU y de la memoria con y sin el BMDetector, durante el proceso de carga de las páginas, de estos valores se genera la siguiente gráfica:

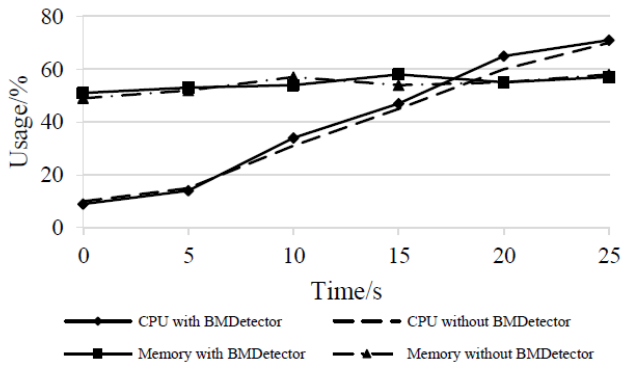


Fig. 9. Performance comparison before and after the deployment of BMDetector

Cómo se puede observar la diferencia del porcentaje de uso es poca en ambos recursos, los autores establecen que esta diferencia esta en un rango aceptable y no interfiere de manera significativa con la experiencia del usuario

V. PROYECTO COINHIVE

Coinhive es un proyecto que ofrece un minero de JavaScript para Monero Blockchain, el cual se puede insertar en un sitio web [5]. Ese código se ejecuta directamente en el navegador y extrae XMR para la persona que inyecto el código. El propósito de Coinhive es brindar a los administradores de sitios web una alternativa diferente al uso de la publicidad para obtener ingresos, estos se obtienen si el visitante entrega los recursos de su CPU.

```
<script src="https://coinhive.com/lib/coinhive.min.js"></script>
<script>
    var miner = new CoinHive.User('SITE_KEY', 'john-doe');
    miner.start();
</script>
```

El uso del código del script de Coinhive se volvió popular entre los atacantes por su simplicidad y la capacidad de ejecutarse con aproximadamente un 65% del rendimiento del host de la víctima.

VI. HIPÓTESIS

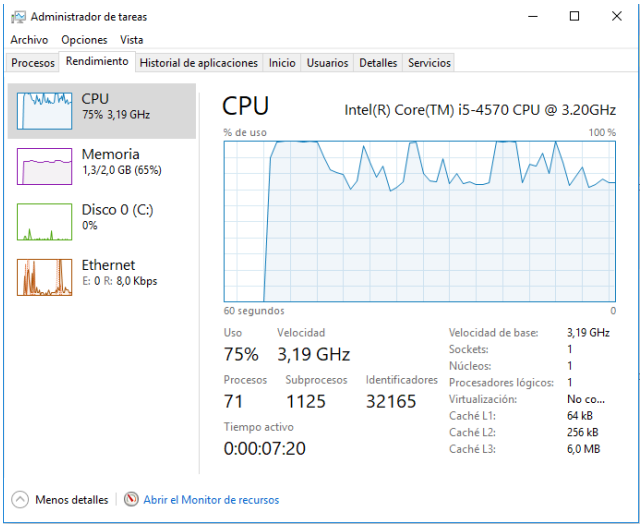
Al inicio del proyecto se planteó la siguiente pregunta: ¿Los mineros que limitan la ocupación de la CPU, pueden ser rastreados a través del tráfico de red? Para esto escogimos el script de Coinhive, con el propósito de capturar el tráfico que este genera.

A. *Ambiente experimental*

Se utilizó para el cliente un sistema con Windows 2 GB de memoria, y un navegador Chrome. Para el servidor se emulo una página web con el script minero de Coinhive que se ejecuta desde el local host. También se utilizó el sniffer.

B. *Proceso experimental y análisis*

Se ejecutó el script cuando se ingresó a la página web, el administrador de tareas mostro un incremento en el uso de los recursos de la CPU, tal como se muestra en la siguiente imagen:



Después de haber ejecutado el script se inició una captura de tráfico utilizando un sniffer conocido como wireshark. Cuando se empezó a realizar el análisis sobre el tráfico se encontró una solicitud a un sitio web conocido como https://www.iana.org/. Analizando este sitio, encontramos que tiene en su servidor un número de ip´s las cuales utiliza un tercero, el sitio web hace referencia que no se hace responsable. Inferimos que el mecanismo en el cual fue diseñado la criptomoneda Monero puede estar haciendo uso de estas ip´s para encubrir las transacciones.

Un número de solicitudes hacia un servidor privado se pudo identificar, cuan exploramos el paquete encontramos que se estaba realizando exploración para encontrar periféricos que presumiblemente estén involucrados con el script.

VII. CONCUSIONES Y TRABAJO FUTURO

El cryptojacking se perfila para ser uno de los ataques más utilizados y peligrosos para los siguientes años, por su sencillez y la facilidad de ejecutarse en múltiples plataformas. Pese a esta premisa, todo puede cambiar si se implementa otro mecanismo de prueba distinto al POW, ya que el sentido de este ataque es usar los recursos de CPU de host secuestrados. En un trabajo futuro tenemos que comprobar la hipótesis ¿Los mineros que limitan la ocupación de la CPU, pueden ser rastreados a través del tráfico de red? pero con criptomonedas las cuales sus transacciones sean transparentes, como ethereum. A partir de los resultados obtenidos por comprobar la hipótesis, planeamos diseñar un modelo que pueda identificar la actividad minera, teniendo en cuenta los métodos de evasión que vallamos descubriendo.

Con criptomonedas con mecanismo de privacidad que tiene Monero es más complejo, ya que las transacciones de esta criptomoneda tienen un mecanismo que hace difícil su trazabilidad, puesto que sus transacciones son ofuscadas y no se puede identificar quien la recibe. Con esta criptomoneda tenemos una hipótesis, la cual es, que sí identificamos como es que consume recursos de la tarjeta madre del computador podemos identificar el inicio de la transacción, y de esta manera saber cómo buscar en el tráfico de red. Cabe resaltar que para comprobar esta hipótesis necesitaremos investigar más y no es garantízale un resultado favorable.

Referencias

[1] S. Nakamoto, «Satoshi Nakamoto,» *Bitcoin: A Peer-to-Peer Electronic Cash System.*

[2] A. K. A. M. Muhammad Saa, «End-to-End Analysis of In-Browser Cryptojacking,» 2018.

[3] A. l. a. b.-b. cryptojacking, « Andreas Leoutsarakos».

[4] Z. Z. X. C. Z. W. Q. L. Jingqiang Liu, «A Novel Approach for Detecting Browser-based Silent Miner,» *IEEE Third International Conference on Data Science in Cyberspace,* 2018.

[5] Coinhive, «Coinhive,» [En línea]. Available: https://coinhive.com/. [Último acceso: 2018].

[6] A. L. Shayan Eskandari, «Afirst look at browser-based cryptojackin,» [En línea].