Behavior-based detection of In-browser Javascript mining code

Hyunki Kim, Junghwan Park, Juhyung Song, Seonyong Jeong

Graduate School of Information Security
Korea Advanced Institute of Science and Technology
Yuseong, Daejeon
{ *saykim*0727, *ahnmo*, *sjh*8563, *s.jeong* } @ *kaist.ac.kr*

***Abstract*—As a cryptocurrency comes up, a lot of people started cryptocurrency mining by using mining programs in their computers. However, if a user executes mining programs in his local computers, they are overloaded by using CPU, GPU and so on. So, many people thought a way, how to mine without damaging their computers. As services like CoinHive are made, many sites have consisted of web-based mining codes to use a visitor's computer resources. Using these services, web administrators involved a javascript mining code instead of advertisement or hide a javascript mining code to run in background. Visitors can't know that web administrators use their computer resources to mine cryptocurrencies in visitors' computer. Because some people thought that the web mining services were morally problematic, several programs are made to detect and prevent these services from mining. However, most of the programs detect javascript mining code based on blacklist which have some specific domain. This way is practically impossible to detect mining codes, because most of web administrators can bypass a anti mining programs. So, we suggest a practical program to prevent mining code from working on web browsers and a way to detect malicious codes by analyzing them.**

I. Introduction

From long ago, some people used to get users' personal secret information by hacking and got profits by selling them. However, personal information leakage attacks become more harder than before because of the high attention on privacy, many ways for a profit are introduced instead of it. For example, in 2016, Mirai botnet, which primarily composed of embedded and IoT devices, infected other IoT devices and performed DDoS attack. A number of botnets were introduced following the Mirai[2]. Some people got a profit by lending or selling the botnets to someone who need them[10]. Also ransomware, which has been a hot issue in recent years such as Trojan.Gpcoder in 2005 and WannaCry or Scarab in 2017, breaks users' computer down or encrypts all files and extorts money from users by decrypting the encrypted files.[23] In recent years, as a cryptocurrency has become an issue, they foucs on it.

Starting with Bitcoin of Satoshi Nakamoto in 2009, many cryptocurrencies are made by programers like Ethereum, Ripple, Litecoin, and so on. A cryptocurrency is a kind of electronic currency mined by algorithms using blockchain based on cryptology. The mining process is called 'Proof-of-work'. The goal of 'Proof-of-work' is finding a nonce value which makes required zero-bits after hashing. By incrementing the value in the block, they calculate iteratively to find the value gives required zero bits after hashing.

Acknowledgement : This paper is a result of term project for IS511 in KAIST, spring semester of 2018.

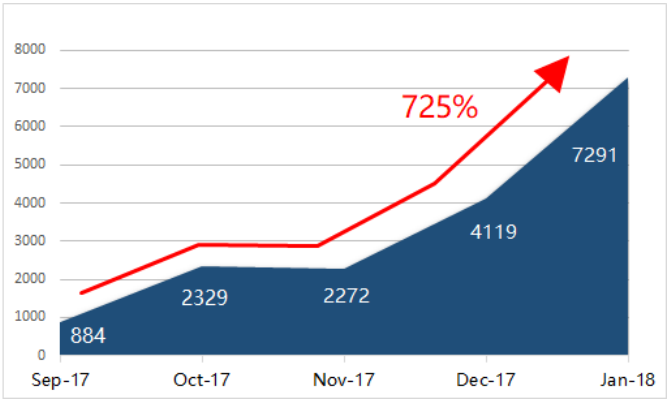

Fig. 1. Increasing rate of web mining usage from September 2017 to January 2018

A cryptocurrency is a platform supporting electronic transactions without relying on trusted third party. One of the major features of a cryptocurrency is electronic transactions without relying on trusted third party. All transactions are announced to the public, all users guarantee the validation of transaction. So the users can transact their cryptocurrencies all over the world with a minimized fee and high speed[14]. Besides, a wallet of cryptocurrency has advantages against general bank account. A bank account directly linked to the owner's physical identity, but a cryptocurrency wallet doesn't. It can be generated anytime without any authentication, so it is difficult to find the real owner from the digital wallet[3]. As cryptocurrencies become more popular for these reasons, many people have mined cryptocurrencies by using the mining programs in their computers to make a profit.

However, it is a problem that the mining programs are working with computer resources likes CPU, GPU, and so on. Because heat emitted by the computer resources has a bad effect on the computers like damage on IC or processing speed slowdown[8]. Therefore some computer experts warn

that people should not mine the cryptocurrency in desktop or labtop. Taking the advice, the people find ways to do mining without problems. For this reason, a web-based javascript mining codes which mine the cryptocurrency using visitor's computer resources on website are developed.

The first developed web mining service is Bitcoin-Plus to mine Bitcoin in 2011 and Coinhive is developed to mine Monero in 2017. Bitcoin-Plus is insufficient to mine cryptocurrencies, but Coinhive that is web mining service for Monero is suitable to mine on website. According to Figure 1, mining web sites using these services soared up about 725 in the four-month period from September 2017 to January 2018.

The original purpose of the web mining services is not evil. Web mining services originally developed to make a profit instead of advertisement. Nowadays, most of web administrator offers information to make a profit from users as a reward. Most of cases, the way to monetize is placing advertisement on their web site. However, most of users do not prefer advertisement. Advertisement makes the web pages dirty and messy. Sometimes, the advertisement even covers the content of web pages. So, the web mining services are introduced for the replacement of advertisement. If a user grant the mining process, the administrators can make a money by using only a little computer resource. However, it is malicious to mobilize user's computer resources by using these services without any their consent. An attacker can hide the script on the hidden iframe to steal the user's computing power secretly. Moreover, some of them not only mines crypto-currency but also adds the advertisement on their sites too. As the number of malicious web mining scripts grows, a way to prevent this kind of attack is required.

In this paper, we will give a information about whole story of web mining in Section II. In Section III we will discuss how existing web mining services work and compare what they are different. we focused Coinhive and JSEcoin because they are very popular than other services. we analyze existing mitigation extensions like Nocoin [12], CoinHive-blocker [1], Minerblock [22], and so on in Section IV. we will discuss our enhanced web mining blocker extension in Section V. In Section VI, we evaluated our program with randomly implemented sites and compare out tool's efficiency with others.

II. Web mining

In the website, which is popular with people, it is possible to mine cryptocurrencies without using miner's local computer resources. Because, they borrow a visitor computer resources on website to mining cryptocurrencies while the visitor is connected to the website instead of using their computer resources.

The first javascript mining codes were JSminer and Bitcoin-plus that is a web-based Bitcoin mining script written in Java in 2011. To get a profit, they were developed to mine cryptocurrencies instead of advertisement, but they weren't often used because their script had insufficient mining speed. After a certain time, Monero (XMR) is an open-source cryptocurrency alternative to Bitcoin developed in April 2014. Compare with a other cryptocurrency, Monero has advantages. First, Monero can ensure a speed of web mining Unlike Bitcoin and other existing cryptocurrencies. Because, Monero is based on the CryptoNight proof-of-work hash algorithm that reduces the performance gap between GPUs and CPUs for mining, and CPU friendly algorithm[20]. This makes it profitable to mine with not powerful computer or inexpensive mining hardware. Monero uses ring signatures technology and one time key to hide who really traded since ring signatures is a available signature to all people. It offers increased privacy by obfuscating the participants in a transaction. So, people can not trace who a transaction belongs to. Because of this technology, this cryptocurrency is popular on called dark web markets. Besides, there is no limit to the quantity of Monero. Bitcoin and other cryptocurrencies have halving since they have limited production. So, when the halving event occurs, a cryptocurrency mining reward will be reduced[7]. On the other hand, Monero have not halving event, which means that the Monero mining reward always constant. Through the development of Monero, a lot of web-based mining script was developed, and the most popular one is CoinHive.

CoinHive, which is cryptocurrency mining service for Monero, was developed in September 2017. the service provide a way for website owners to make a profit without running intrusive or annoying advertisements. CoinHive is easy to implement and configure, and also doesn't interfere with the design or user experience of a site in any way. In other words, a website owner just input two javascript line to use CoinHive service. Because of these advantages, CoinHive is most used in the world , about 91%, than any other web mining services[9]. In Section III-A, we will give the information about Coinhive in detail.

Most of web miners mine cryptocurrencies with a mining pool since it is difficult to mining alone against some miners who have powerful computer resources. It is a mining pool that miners who have been mined alone can gather their own computing power, increase their profits, and distribute profits. they do not make a big money in mining pool, but they steadily get a profit.

However, mining pool system has some problems. If a mining pool group has more than half the hash power of a cryptocurrency block chain network, the cryptocurrency block chain will lose credibility, this is called "51% attack"[5]. The mining pool group makes a malicious chain and inserts the chain in public chain. Than, normal block chains that have transactions become invalid, people who traded normally may not receive the cryptocurrencies, and people who have succeeded in mining will not receive reward. That means that normal block chains that have traded so far have become invalid, people who traded normally may not receive the money, and people who have succeeded in mining will not receive it either In fact, Ghash.io mining pool had a hash power of over 50% for about 12 hours. Ghash.io protected value of block chain network by blocking new subscriptions

Nation	Desktop	Nation	Desktop	Nation	Desktop
Greece	39%	Denmark	25%	Australia	20%
Ireland	39%	Canada	24%	Israel	19%
Poland	33%	New Zealand	24%	Spain	19%
Germany	29%	Finland	23%	Switzerland	18%
Singapore	29%	Slovenia	23%	U. States	18%
Iceland	27%	Croatia	22%	Italy	17%
Sweden	27%	Bulgaria	21%	Latvia	17%
Austria	26%	Lithuania	21%	Malta	17%
Estonia	26%	Portugal	21%	Netherlands	17%
Hungary	26%	Romania	21%	Serbia	17%

Fig. 2. Ad blocker usage by country

on its own.

Web advertisements that is the most popular way for profit on the web. However, web mining has good advantages than web advertisements.

- There are many ad blockers and they are very effective.
- income of web mining is in proportion to the time that visitors spend on the websites.
- a lot of advertisements have a bad effect on visitors.

First, There are many extensions that effectively block web advertisements on the Internet. So, many people use a ad blocker to block web advertisements and websites owners who upload advertisements on their web pages can not make a money.

According to PageFair who is the global authority on ad blocking, cited by The New York Times, The Financial Times, and etc and eMarketer who help is to help our clients make better decisions in a world being transformed by digital, 11% of the global population uses an ad blocker, and some countries show almost 30% to 40% ad blocker usage in Figure 2[16]. In other words, website owners will not be able to make a money by advertising. However, mining blocker is insufficient to prevent javascript code from mining. we will give the information in section IV. Second, when a user visits a website, only one ad impression increases regardless of how long the user stays on the website. This means that web advertisements is inefficient on some sites that have involved time-consuming content like movie/sport/lecture, etc. However, web mining specialize in that case compare to web advertisements. Because web mining service gets a result by hash computation continuously, web miner makes a money in proportion to the time that visitors spend on the website. Third, it is better than web ad that User Experience Design of web mining. Website, which has a lot of advertisements, makes visitors feel uncomfortable and website has an negative effect on their next visit. Even, some contents of web site are screened by many advertisements and visitors can't read them.

For these reasons, the number of mining sites is increasing. To mining, website administrators upload a mining script on their website or hackers inject a mining scripts on other websites. According to Cyren who Internet security technology company, they have found a 725% increase in the number of domains running mining scripts on one or more pages in the four-month period from September 2017 to January 2018 in Figure 1[6]. Mining script was found embedded in all Web pages served by a Free WiFi hotspot at a Starbucks in Buenos Aires and Argentina. in January, it was found hidden inside of YouTube advertisements on Google's double-click platform in a lot of countries including Japan, France, Taiwan, Italy and Spain. Also in February, It was found on Browsealoud, which is assistive technology software that adds text-to-speech functionality to websites. The service is widely used on many UK government websites, in addition to a few US and Canadian government sites[4]. Even around 4200 websites of United Kingdom government are injected a mining script by hackers[9]. As you can see in many cases, many sites have been mining by arbitrary or other means, and people do not know this.

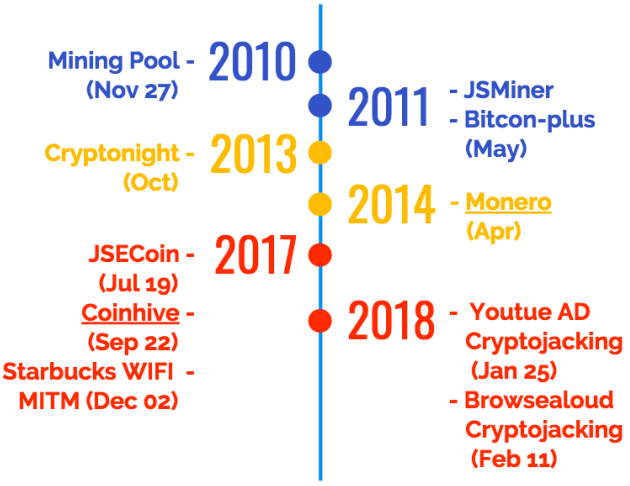

Fig. 3. Timeline of events related with web mining

III. Mining services analysis

Over time, the number of mining library sites and mining sites that use library sites have been increasing significantly. In this section, we will analyze characteristics of a sites that provide mining services or mining libraries and a sites that use these library to mining.

According to PublicWWW statistics on December 24th, 2017, the two mining codes are used in approximately 95% of mining sites[9]. JSECoin is developed to replace CoinHive, but JSECoin can not keep up with CoinHive usage. So, we will focus on the two web service and analyze them.

For mining operation, most websites must be connected to a mining pool that is a special group to mine cryptocurrencies. First, web miners authenticate that they belong to the mining pool. If their verification is success, they can communicate with mining pool. Web miners receive tasks called 'Job' and 'Verify' from a mining pool, perform the tasks, and resend the result of the tasks to the mining pool. The mining pool

verifies the results and if the verification is success, it offers rewards to the miners. To get a available hash value faster than other people, it is very important that a communication speed between web miners and a mining pool. This should be processed in real time so that the miner can hold a dominant position on a mining race.

Most mining scripts, including CoinHive and JSECoin, make a main object named 'Miner' by an initializing code(but this is code-specific). 'Miner' object exchanges data with the mining pool by using web socket. In the first connection to a mining pool, 'Miner' sends a certification with 'sitekey' and if the certification is valid, gets tasks from the mining pool. after receiving tasks, 'Miner' creates job threads and distributes tasks to them. In terms of thread-usages and algorithms, it has two ways to mine: 'Web Worker' in CoinHive; 'Web Crypto' in JEScoin.

JSECoin is a cryptocurrency and also a mining service site[11] to mine JSECoin. About JSECoin whitepaper, the CPU usage for JSECoin hashing algorithm is less resource demanding than the loading of a video advertisement. JSECoin just use 10 15% CPU resource, and it never really has a huge impact on visitor's other behavior. By using surplus CPU availability within the browser they can efficiently hash data at scale to secure the JSE block chain.

A. CoinHive

The architecture of CoinHive service is in figure 4. In CoinHive service architecture, each job threads creates a web worker and send the tasks. A web worker makes it possible to run a script operation in background thread separate from the main execution thread of a web application. The advantage of this is that laborious processing can be performed in a separate thread, allowing the main (usually the UI) thread to run without being blocked/slowed down.[13]

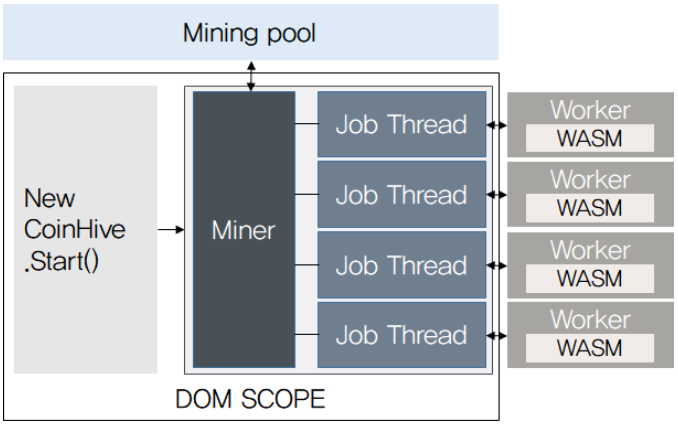

Fig. 4. Web mining architecture using CoinHive

One important component is a communication between a web worker scope and DOM scope. Because of the separation between DOM scope and a web worker scope, they can communicate with each other by using 'postMessage' and 'onmessage' function. Each scope sends a data by using 'postMessage' function with some argument strings that are verify, job, and so on, and receives a data by using 'onmessage' function in Listing 1. Web workers call 'verify' or 'work' function to calculate with an algorithm compiled to a web assembly, if they receive job.verify_id or job_id.

```
CryptonightWASMWrapper.prototype.onMessage =
    (function(msg) {
   var job = msg.data;
   if (job.verify_id) {
      this.verify(job);
      return
   }
   if (!this.currentJob ||
       this.currentJob.job_id !== job.job_id)
       {
      this.setJob(job)
   }
   if (job.throttle) {
      this.throttleWait = 1 / (1 -
          job.throttle) - 1;
      this.throttledStart = this.now();
      this.throttledHashes = 0;
      this.workThrottled()
   } else {
      this.work()
   }
}
```

Listing 1. onMeessage Method on Web Worker code

A web assembly is a new type of code that can be run in modern web browsers. - It is a low-level assembly-like language with a compact binary format that runs with near-native performance and provides languages such as C/C++ with a compilation target so that they can run on the web. It is also designed to run alongside JavaScript, allowing both to work together[13]. Also, a web assembly algorithm has a merit that it is 84 times faster than same one written in javascript[17].

A web assembly algorithm that is used in a web worker depends on a cryptocurrency that a mining service want to mine. For example, 'Scrypt' algorithm is used to mine Litecoin and 'Neo Scrypt' algorithm is used to mine Feathercoin at Minechurch in 2014.

Currently, CoinHive uses CryptoNight algorithm to mine Monero. Cryptonight, which is a kind of 'proof-of-work' algorithm, is designed for using a CPU on website. it relies on random access to a slow memory and emphasizes latency dependence. Because it relies on random access to a slow memory, it is optimized for a CPU using L3 cache rather than a GPU using GDDR5 that has slower speed than L3 cache[17].

B. JSECoin

Some mining services use web crypto instead of web worker, and JSECoin is the most popular site using web crypto. A web crypto API is an interface allowing a script to use cryptographic primitives in order to build systems using

cryptography. A fundamental feature of this API is to allow the manipulation and storage of private and secret keys without requiring the underlying bits of the key to be made available to javascript[24].

Architectures using web crypto have common steps with architectures using web workers, especially in steps before making web workers, but it doesn't make any worker. Because web crypto depends on threading technology provided by browser. So, JSECoin uses 'window.crypto' object provided by web browser instead of job threads or web worker.

Using web crypto is as fast as using web assembly, because it works as native codes in a web browser. About the difference in speed between web crypto and web assembly, this site[25] says that web assembly just 20% slower than web native code execution speed.

However, not all web browsers support the web crypto module. If the browser doesn't support web crypto api, they cannot use it. To solve this problem, JSECoin made fallback function for web browsers does not support web crypto. Fallback function is executed if a contract is called and no other function matches the specified function identifier, or if no data is supplied. Since it is not native code but javascript code, the speed of fallback function is slow than native code or web assembly. In addition, the web crypto has strong dependency on web browser, it is difficult to optimize algorithms which web crypto doesn't provide.

```
function cryptoSha256(str, nonce) {
   var buffer = textEncoderUTF8(str);
   return crypto.subtle.digest("SHA-256",
       buffer).then(function(hash) {
      return hex(hash) + "," + nonce
   })
}
```

Listing 2. Native sha256 function in JSECoin Code

IV. Existing Mitigations

As we mentioned in section II, the number of crypto mining pages are increasing. Crypto mining could degrade the user experience and also has a bad effect on the user's computer resources. Above all things, it is illegal to extort users' computer resource without their consent. So, the web developers implemented some chrome extensions which can detect lurked mining process: Nocoin, MinerBlock, CoinHive Blocker, Nominers.

Table I shows four techniques for detecting and blocking lurked mining scripts: Blacklist, Whitelist, DOM element checking, CPU Usage.

We will explain the features of each techniques and its advantages and disadvantages in Section IV-A, Section IV-B and Section IV-C.

A. Blacklist and Whitelist

The first technique to mitigate the damage of cryptojacking is Blacklist. An extension using blacklist technique keeps a website's list that are web mining sites. Before a connection is established, the extension checks whether the destination site is on the blacklist. If the destination site is on the blacklist, the site is blocked by the extension and the extension notifies it to the user. However, as we mentioned in Section I, not all mining script is malicious. Sometimes, a web site administrator uses mining script to make a profit instead of advertisement with the user's consent. To allow these sites some extensions keep another website's list, called 'Whitelist'. A user can add sites on the whitelist, if the user wants to access a site blocked by extension. So, the extension keeping 'Whitelist' allows the mining process if the web page granted by the user.

TABLE I
Anti Mining Program

	Nocoin	MinerBlock	CoinHiveBlocker	Nominers
Blacklist	O	O	O	O
# of Blacklist	158	370	1229	43
Whitelist	O	O	O	-
DOM element	-	O	-	-
CPU Usage	-	-	O	-

Blacklist and whitelist are the easiest and fastest way to mitigate the damage of cryptojacking. The developer can successfully block the malicious mining scripts by keeping only single or double lists of websites. So as we can see in Table I, four extensions use blacklist method and three extensions use whitelist excepts Nominers.

However, there are some critical issues on blacklist technique, bypassability and evocation. First, the blacklist method can be bypassed easily. Blacklist cannot detect the behavior of the mining process. It is hard to update the blacklist in real time, so an attacker can bypass by compromising another websites if the original one is listed on blacklist. To prevent this, web developers added the URLs which are required for mining process. For example, an attacker should download CoinHive libraries from "*http://coinhive.com/lib/coinhive.min.js*". Also, CoinHive should establish a connection with mining pool via a proxy server. By listing these URLs on the blacklist, web developers successfully blocks cryptojacking on websites which are not in blacklist. However, this technique also can be detoured by uploading the library on the private website or making private proxy server.

Second, the blacklist method has an evocation issue. If the administrator removed the mining script injected by an attacker, the website is not a malicious site anymore. For example, "*clod.pw*" is on CoinHive-blocker's blacklist but the site is not a mining script embedded website anymore. If the website is already in blacklist, the extension don't check the mining script is alive in blacklist technique. For this reason, some sites may be blocked in spite of the fact that mining scripts are no longer included.

B. DOM Element checking

It is not efficient to detect mining scripts only based on blacklist. So, MinerBlock introduced a technique, DOM element checking. When an attacker instantiate a mining instance

with the CoinHive constructor, the instance initiated by the attacker is allocated in the web page's DOM scope. So by checking the variables of the web page object, we can judge whether the mining instance is running or not.

TABLE II
MINERBLOCK'S DOM ELEMENTS BLACKLIST

Name	Type	Description
isRunning	function	checks whether the miner is currently running
stop	function	stops mining and disconnect from a mining pool
_siteKey	string	user public Site-Key for authentication

Table II shows the significant properties of the CoinHive, which is selected by MinerBlock. If a variable which has these properties exists, MinerBlock make a judgment that the variable initialized with a CoinHive constructor. To check whether the variable exists or not, MinerBlock runs the web scripts in seconds because the variable is initialized after the script finishes its execution. If a variable which contains mining instance is detected, MinerBlock can disable the web mining script by calling stop function.

However, this technique can be detoured by manipulating the library. MinerBlock detects the mining scripts based on the properties' name like '_siteKey' and 'isRunning' and so on. So if an attacker renames the properties, the extension cannot detect the mining instance. For example, if the attacker renames '_isRunning' to '_runningState' by modifying library, the extension cannot find the variable because of the absence of the property.

In addition, if the attacker does not allocate a instance in a variable, the extension cannot detect the mining script. Listing 3 shows a script which can bypass the MinerBlock's DOM element checking technique. The attacker only instantiate the CoinHive instance. She didn't allocate it into a variable. So, the instance floats in heap area and we cannot check the properties of it even if the instance is working.

C. CPU Usage

The last technique already exists is CPU Usage checking. Mining a crypto-currency is a hard-loaded task. Because of the successive hash calculation, they uses user's computer resources. As mentioned in Section II, web browser based mining script uses CPU resources a lot to mine Monero.

```
var a=document.createElement('script');
a.src=
   'https://coinhive.com/lib/coinhive.min.js';
a.onload=function(){
   new CoinHive.Anonymous('site-key',{throttle
   :0.3}).start()
};
document.body.appendChild(a);
```

Listing 3. Can bypass MinerBlock program

So, we can detect Mining scripts by checking side channel of the mining script. CoinHive Blocker monitors the CPU usage and notifies users that the website is suspected to have a mining scripts when the usage soars up.

This is not an efficient way to detect mining scripts. A lot of web browser use CPU resource for many purposes. They cannot distinguish the high CPU usages are from mining scripts or not. Also, it can be detoured by lowering the CPU usages. As we can see in Listing 3, there is a property in CoinHive named throttle. By setting this value, they can limit the usage of CPU. The coinhive-blocker reports the website to the developers when the CPU usage is over 50%, so we can bypass it by setting this value lower. This can degrade the performance of cryptocurrency mining, but the attacker can make a profit silently.

V. MINING LOCK

So we implemented a chrome extension, '*Mining Lock*', to reinforce the existing method mentioned in Section IV. *Mining Lock*'s goal is to detect mining processes running on background without any consents from the user. If a lurking mining process is detected, our extension creates a notification about it and offers two choices to the user; kill the process or not. If the user doesn't respond to our notification about the mining process, the extension kills the Miner or refuses the web connection automatically.

To minimize false negative/false positive rate and maximize the efficiency, we designed our extension with 3 steps : Web request blacklist, DOM Element based detection, Javascript event-logging based Detection.

A. Web Request Blacklist

Web request blacklist is the fastest way to detect whether the website is mining or not. However, blacklists of existing anti-mining programs have a strong weakness. It is hard to revoke the blacklist when a web administrator removes the lurked mining process. A website, which does not mine anymore, is also blocked by existing anti-mining programs.

So, we minimized the blacklist. Every mining process has its own initiating process. In case of CoinHive, they should download mining libraries form "https://coinhive.com/library/coinhive.min.js". Also, JSECoin needs loading process by making a get request to "https://load.jsecoin.com/load/{:AccountNo}/{:PublisherSite}/{:OptionalSubID}/0/.

If the library downloading is successfully done by moving the library files on another server, the process sends the results of hash calculation back to the mining pool through proxy server. By adding the proxy server addresses on the blacklist, we can successfully prevent the mining process.

Web request blacklist is the fastest way to detect malicious mining website. It blocks initializing process and message passing route with the mining pool. However, the blacklist can be detoured by some techniques. The attacker can post the library files in another server and create the proxy servers continuously. Blacklist system cannot response these kind of detour immediately.

B. DOM Element based detection

Blacklist based mining detection is the fastest way to find malicious mining code. However, this way can be easily

detoured by techniques mentioned in [Section V-A]. So after the blacklist detection, *Mining Lock* checks DOM elements to detect Web based Miner. After downloading the library file, the browser instantiates it and the instance becomes an element of the DOM. To detect miner, it checks all DOM variables and find the instance which has Mining feature properties. The properties are mentioned in TABLE III.

TABLE III
MININGLOCK'S DOM ELEMENTS BLACKLIST

Name	Type	Description
isRunning	function	checks whether the miner is currently running
stop	function	stops mining and disconnect from a mining pool
_siteKey	string	user public Site-Key for authentication
getToken	function	checks whether the miner have Mining Token
_setjob	function	set a job to threads
_verifyThread	object	a thread to verify a result of thread caclulation

DOM Element based detection can catch the mining scripts bypassing the blacklist. If the detector substitutes the trigger function to null before the function call, the miner even cannot starts the work. However, if the instance is not allocated to a variable like Listing 3, the detector cannot notice it. Also, if the attacker modifies properties' name, the detection process breaks up.

C. Event-log based detection

Lastly, we designed a powerful way to detect lurking mining process: Event-logging based detection. In case of CoinHive, dominant library for cryptojacking, creates WebWorker instances for hash calculation. Also it establishes a web socket connected to proxy server and communicate with mining pool with messages in Table IV. By getting logs from the events and aggregating them, we can successfully detect the mining script on the background.

TABLE IV
WEB SOCKET DATA OF MINING SITE

Type	data
request	[type:"auth",params:[site_key:x,type:"anonymous",user:x]]
response	[type:"authed",params:[token:x,hashes:x]
response	[type:job,params:[job_id:x,blob:x]

First, *Mining Lock* collects WebWorker instantiating events by overriding the worker constructor. If *Mining Lock* injects the overriding scripts before the page loading, CoinHive instantiate WebWorkers with overrided constructor. The logs are delivered to the detector, so that the detector can get the number of workers.

Second, *Mining Lock* hooks socket messages by using wsHook[19]. Hooked messages are delivered to the detector. Table IV shows message types passing between mining pool and CoinHive. CoinHive sends a message which type is 'auth'. When the mining pool receives this auth message, they send back a message 'authed'. After the authentication process is done, the mining pool send a job by sending 'job' type message.

Mining Lock aggregates information from wsHook and overridden Web Worker. If the number of web worker is more than one, and the 'auth', 'authed' and 'job' type messages are hooked sequentially, the detector judge the page as a mining embedded page. It make a consent requesting notification and kill the worker if the user does not agree with the execution.

VI. EVALUATION

We generated some test sites for evaluation of extensions. These sites are based on 10 free templates that have no license issues. The test cases are grouped into two cases, doing perform the mining or not.

There are six types of mining cases.

1) CoinHive Direct : Get the mining script from CoinHive. Then, connect to CoinHive's pool proxy.
2) CoinHive Server : Get the mining script from its own server. Then, connect to CoinHive's pool proxy.
3) CoinHive Rename : Get the mining script from its own server, but the script masquerades as benign library. Then, connect to the CoinHive's pool proxy.
4) CoinHive obfuscated : Get the mining script from its own server, but the script is obfuscated. Then, connect to CoinHive's pool proxy.
5) JSECoin : Get the mining script from the JSECoin. Then, connect to JSECoin.
6) Self-hosted : Get the mining script from its own server. Then, connect to own pool proxy.

To build a mining pool proxy used by Self-hosted, we used *CoinHive-stratum-mining-proxy* [21] and connected it to *nanopool* [15].

The evaluation targets are the existing extensions that we mentioned: No-Coin, MinerBlock, Coin-Hive Blocker and NoMiner. Also, we evaluated the new extension that we implemented an extension using our proposed method, Mining Lock.

- No Coin (Ext1)
- minerBlock (Ext2)
- Coin-Hive Blocker (Ext3)
- NoMiner (Ext4)
- Mining Lock (Ext5)

To evaluate the extensions above, we wrote a test script using *Selenium* [18] which is a solution for web browser automation. Also, we modified the extensions to print the fixed keyword on the debug console when a mining process is detected. With this keyword, we can check whether the extension has mining script or not.

A. First Evaluation

We generated 10,000 sites for evaluation. There are 4,000 sites that do not perform mining and 6,000 sites that perform 6 cases of mining.

We gave three-second-delay for each site after loading. After the script is executed, each extension needs time to detect the mining action. In this experiment, we didn't care about the methodology that checks CPU usage rate. This method requires too much time to be tested because it checks the CPU rate every 10 seconds. Also, this can be easily bypassed by using implementation issue. So we didn't care about it.

Extension	True Positive	False Negative	True Negative	False Positive
Ext1	66.60%	33.40%	98.93%	1.06%
Ext2	66.33%	33.67%	99.63%	0.38%
Ext3	69.93%	30.07%	89.25%	10.75%
Ext4	16.60%	83.40%	100.00%	0.00%
Ext5	56.18%	43.82%	99.55%	0.45%

TABLE V
TEST RESULT OF FIRST EVALUATION

Table V is the result of first evaluation. In this result, our extension has bad performance compared to the existing extensions. However, we need to see the detail of each cases.

Site Type		Ext 1	Ext 2	Ext 3	Ext 4	Ext 5
CoinHive	T	99.6%	99.6%	100.0%	99.6%	67.1%
	F	0.4%	0.4%	0.0%	0.4%	32.9%
JSECoin	T	99.1%	99.1%	99.2%	0.0%	0.8%
	F	0.9%	0.9%	0.8%	100.0%	99.2%
Self-hosted	T	1.5%	0.5%	11.4%	0.%	69.5%
	F	98.5%	99.5%	88.6%	100.0%	30.5%

TABLE VI
DETAIL OF FIRST EVALUATION

We can see more details of the first experiment in Table VI. It shows that existing extensions detect CoinHive well but they cannot detect self-hosted case. In addition, the exisitings successfully detected JSECoin but the other extension didn't. Ext5 that uses our proposed methodology detected about two thirds at both mining sites, CoinHive and self-hosted. However Ext5 is failed to detect JSECoin. Because it requires user agreement to do mining at embedded javascript. So we can see that our extension has a higher detection rate in self-hosted compared to existing extensions.

We thought that the reason why our extension has lower detection rate compared to other extensions is on our approach. In our approach, the extension should wait for a web socket connection to be established. After the socket established, the extension can check the status of sending and receiving specific messages. It seems to be caused by the fact that the mining script is not enough to connect normally in 3 seconds. Also there is a routine to get user consent in JSECoin. It requires user consent and it doesn't create any web socket connection without user's consent. Because of this, other extensions detect it through blacklists, but our approach doesn't. Moreover, we execute the test script in parallel in one workstation. The parallel environment makes the required delay much longer. This is assumed to be related to the false rate.

So in our testing script, we increased the wait time per site from 3 seconds to 10 seconds. Also, we add a routine to handle user consent for JSECoin.

Extension	True Positive	False Negative	True Negative	False Positive
Ext5	95.93%	4.07%	99.85%	0.15%

TABLE VII
TEST RESULT WITH ENHANCED TEST SCRIPT

Table VII is the result of the experiment with modification our test script. Compared to the previous one, our approach has become similar *True Positive* rate with other extensions at CoinHive.

We analyzed the False-Negative cases and found some errors on the site generating process. We inserted mining scripts in random location of the templates, but some code is inserted in unreachable parts in Google Chrome. We can get the result that our extension performs well to detecting browser-based mining even self-hosted case. To get a fair evaluation result, we retested for five extensions through modified mining sites and test script.

B. Second Evaluation

In previous evaluation, we found the problem, time. So, we increased the wait time per site from 3 seconds to 10 seconds. Also we added a routine to handle user consent for JSECoin. Since we assumed that the parallel environment would be a problem, we will test sequentially in this time.

Also we found the evaluation works fine. However we need to get the fairness each extensions. So we decreased the number of sites from 10,000 to 1,000.

CoinHive server and *CoinHive Rename* are similar to *CoinHive Direct*, so they are excluded from this experiment. That means there are 400 sites that doing mining and 600 sites that doing not mining.

Extension	True Positive	False Negative	True Negative	False Positive
Ext1	75.00%	25.00%	100.00%	0.00%
Ext2	75.00%	25.00%	100.00%	0.00%
Ext3	75.00%	25.00%	100.00%	0.00%
Ext4	25.00%	75.00%	100.00%	0.00%
Ext5	94.75%	5.25%	100.00%	0.00%

TABLE VIII
TEST RESULT OF SECOND EVALUATION

In Table VIII, there is no false positive rate. It means that our script gets more reliability on the evaluation. The results of these experiments were performed on a limited site that we generated. So, further evaluation's target should be the sites from Alexa 1M to gain objectivity.

Compared to the first evaluation, the second one shows that there is much less false rate. We can see all of extensions perform detecting CoinHive based mining sites in Table IX.

Site Type		Ext 1	Ext 2	Ext 3	Ext 4	Ext 5
CoinHive Direct	T	100%	100%	100%	100%	100%
	F	0%	0%	0%	0%	0%
CoinHive Obfuscated	T	100%	100%	100%	0%	100%
	F	0%	0%	0%	100%	0%
JSECoin	T	100%	100%	100%	0%	79%
	F	0%	0%	0%	100%	21%
Self-hosted	T	0%	0%	0%	0%	100%
	F	100%	10%	100%	100%	0%

TABLE IX
DETAIL OF SECOND EVALUATION

It shows our extensions has strength that is not strengths in detecting self-hosted sites that doing mining.

However, JSECoin detection rate has little defect. The other extensions detect JSECoin with blacklist, so they works well against it. But our approach don't use blacklist. To explain this, we need to know most of coin mining process have inconstant frequency of uplink and downlink which is the mining works communication. In our implementation, it monitors the authentication part in the case of CoinHive, but it monitors the job in case of JSECoin. Table IX shows 79% of JSECoin sites give the job in 10 seconds. That is our extensions detected.

VII. CONTRIBUTION

This is the first paper which not only systemize the knowledge of web browser based mining scripts but also implemented a real extension. Our paper has two main contribution:

- Arouse people the harm of malicious web mining by systemizing the knowledge of web based mining
- Developing an extension which can protect user's computer resources more effective than existing tools.

Based in Section II, III and IV, we offer precise and specific information about web based mining to user who are not aware of it. Because unlike local mining concept, web based mining concept is no research in academic publication.

Also as you can see in Section II, we implemented an effective anti-mining extension based on the context we analyzed, so that the user can protect their computer from the malicious mining web sites.

VIII. CONCLUSION

Recently, web browser based mining attracts many web administrator as a new business model and it becomes more and more popular.5According to the research of PageFair, a lot of advertisement blocking extensions are already developed and broadly used in many countries to block the advertisements that were a major profit model of web administrators. Instead of deploying advertisements, web administrator started to insert mining scripts on their website. Inserting mining scripts has an ethical issue. It uses visitor's computer resource so this process should acquire user's consent. However, most of web administrators don't care about this issue and hide the scripts on their sites. In contrast to the advertisement blocker, there are only a few web mining blocker and the few miners have problems to detect malicious mining scripts.

Most of mining blocker uses blacklist. It is the fastest and easiest way to detect it, but also it is the easiest way that web miners can bypass. Only changing a URL that is not on the blacklist can disable the blacklist. There are some more effective way to detect mining scripts, checking DOM elements and checking CPU usage. However, they also can be bypassed. If the web administrator doesn't allocate the miner instance in DOM variable, DOM element checking cannot detect the mining script at all. CPU usage checking method has high false-positive ratio and easily bypassed by adjusting the CPU usage throttle. So, we developed powerful mining blocker : "Mining lock".

Mining lock not only has blacklist and checking DOM elements, but also checks web socket traffics by logging javascript event. Most of web mining sites use web socket to communicate with mining pool. According to Section VI, our extension has high detection ratio mining sites using CoinHive by checking the web socket traffics. In this paper, we suggested a prototype which can successfully blocks mining sites using CoinHive and we are exploring this further by testing on real-world website and investigating the patterns of another web mining library.

REFERENCES

[1] andreas0607. 2018. Coinhive-blocker. (2018). https://github.com/andreas0607/CoinHive-blocker.

[2] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the mirai botnet. In *USENIX Security Symposium*.

[3] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. 2015. Sok: research perspectives and challenges for bitcoin and cryptocurrencies. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy* (SP '15). IEEE Computer Society, Washington, DC, USA, 104–121. ISBN: 978-1-4673-6949-7. DOI: 10.1109/SP.2015.14. https://doi.org/10.1109/SP.2015.14.

[4] BrianKrebs. 2018. Who and what is coinhive? (2018). https://krebsonsecurity.com/2018/03/who-and-what-is-coinhive/.

[5] Daniel Cawrey. 2014. Are 51% attacks a real threat to bitcoin? (2014). https://www.coindesk.com/51-attacks-real-threat-bitcoin/.

[6] Cyren. 2018. 725% increase in cryptocurrency mining threatens more than just your cpu. (2018). https://www.cyren.com/blog/articles/increase-in-cryptocurrency-mining-threatens-more-than-just-your-cpu.

[7] Jacob Donnelly. 2016. What is the 'halving'? a primer to bitcoin's big mining change. (2016). https://www.coindesk.com/making-sense-bitcoins-halving/.

[8] Dr-Hack. 2018. How crypto mining ruins your machine. (2018). https://blog.drhack.net/cryptocurrency-mining-damages-laptop-d.

[9] Shayan Eskandari, Andreas Leoutsarakos, Troy Mursch, and Jeremy Clark. 2018. A first look at browser-based cryptojacking. *CoRR*, abs/1803.02887. arXiv: 1803.02887. http://arxiv.org/abs/1803.02887.

[10] Syed Fida Gillani, Ehab Al-Shaer, Sardar Ali, and Syed Ali Khayam. 2012. Monetizing spambot activity and understanding its relation with spambot traffic features. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security* (ASIACCS '12). ACM, Seoul, Korea, 51–52. ISBN: 978-1-4503-1648-4. DOI: 10.1145/2414456.2414486. http://doi.acm.org/10.1145/2414456.2414486.

[11] JSECoin. 2018. Jsecoin. (2018). https://jsecoin.com/.

[12] keraf. 2018. Nocoin. (2018). https://github.com/keraf/NoCoin.

[13] MDN. 2018. Webassembly. (2018). https://developer.mozilla.org/en-US/docs/WebAssembly.

[14] Satoshi Nakamoto. 2008. Bitcoin: a peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf. (2008).

[15] nanopool.org. 2018. Nanopool — monero. (2018). https://xmr.nanopool.org/.

[16] pagefair. 2017. The state of the blocked web. (2017). https://pagefair.com/downloads/2017/01/PageFair-2017-Adblock-Report.pdf.

[17] Nicolas van Saberhagen. 2013. Cryptonote v 2.0. (2013). https://cryptonote.org/whitepaper.pdf.

[18] SeleniumHQ. 2018. Selenium - web browser automation. (2018). https://www.seleniumhq.org/.

[19] skepticfx. 2017. Wshook. (2017). https://github.com/skepticfx/wshook.

[20] Jacob Tuwiner. 2018. Monero mining hardware comparison. (2018). https://www.easypc.io/crypto-mining/monero-hardware/.

[21] x25. 2018. Coinhive-stratum-mining-proxy. (2018). https://github.com/x25/coinhive-stratum-mining-proxy.

[22] xd4rker. 2018. Minerblock. (2018). https://github.com/xd4rker/MinerBlock.

[23] Wira Zanoramy A. Zakaria, Mohd Faizal Abdollah, Othman Mohd, and Aswami Fadillah Mohd Ariffin. 2017. The rise of ransomware. In *Proceedings of the 2017 International Conference on Software and e-Business* (ICSEB 2017). ACM, Hong Kong, Hong Kong, 66–70. ISBN: 978-1-4503-5488-2. DOI: 10.1145/3178212.3178224. http://doi.acm.org/10.1145/3178212.3178224.

[24] ZeroDot1. 2018. Coinblockerlists. (2018). https://github.com/ZeroDot1/CoinBlockerLists.

[25] Alexander Zlatkov. 2017. How javascript works: a comparison with webassembly + why in certain cases it's better to use it over javascript. (2017). https://blog.sessionstack.com/how-javascript-works-a-comparison-with-webassembly-why-in-certain-cases-its-better-to-use-it-d80945172d79.