Universiteit Leiden
The Netherlands

# Opleiding Informatica

## More than a one-time heist

Identifying the current state of
Android CPU cryptojacking

Louise van der Peet

Supervisor:
Olga Gadyatskaya

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

2020

**Abstract**

Cryptojacking is a much discussed threat among the cybersecurity community. It gained popularity after 2017, when attackers started using mining scripts, like Coinhive, to illicitly mine cryptocurrency on victim's devices. This thesis aims to evaluate the landscape of cryptojacking after the discontinuation of the Coinhive library. We establish the current state of the mining services for cryptomining and cryptojacking on Android. Focusing on finding active libraries, while evaluating their size and impact, we discovered that no competitor has taken over the market share of Coinhive. Furthermore, the thesis explores possible explanations as to why no competitor has performed as well as Coinhive.

We implement mining applications, analysing how easily the mining code can be implemented and detected, by running the mining code through anti-virus scanners and checking whether it is detected. We then further research the existing approaches mining detection. The found methods of static detection are keyword search by library, by mining credentials, by mining domains and general keywords. Dynamic detection techniques include CPU metrics that indicate mining and specifically looking for the algorithms that miners use. Moreover, we estimate the daily profit of mining and compare it to profits during the cryptocurrency boom and other ways of app monetization. We find a decreasing profit cryptomining in general, and especially on Android. We found that currently the profit of a 24 hour mining cycle on a mobile device is less than showing 1 full-screen advertisement. This indicates that in the current economic situation, there might not be as much interest in cryptojacking as there was in 2017.

# Contents

# 1 Introduction

Since the creation of the Bitcoin in 2009, its potential for anonymous monetization has attracted a lot of cybercriminals and end-users trying to make a profit. This is due to its anonymity in making and receiving payments, but also due to the practice of cryptomining. *Cryptomining* is a process to support a cryptocurrency network by performing calculations in exchange for cryptocurrency. Following the creation of Bitcoin, many other cryptocurrencies emerged, supported by cryptomining.

In 2018, the cryptocurrency market crashed. After this, the business of cryptomining has been presumed unprofitable and almost dead. Some GPU miners, which work on a circuit especially designed to rapidly process images, still manage to get a profit using their large processing power. At the same time, individual miners with commodity hardware tend to decrease in numbers.

However, since individual mining is deemed non-profitable, CPU crowd mining has taken a surge. *Crowd mining* is the practice of using a large number of devices to mine for one user account, using a CPU, which is the circuit that executes instructions to make computer programs. Since CPU's only perform enough calculations to make, at most, a few cents a day, a very large number of devices is needed for this practice to be fruitful. For miners to have access to such a number of CPU's, they will often try to use other people's devices.

Crowd mining can also be used for legitimate goals, such as monetization for websites or apps, a possible alternative to advertisements. However, when this is done without the knowledge of the user, it is referred to as *cryptojacking*, a form of hacking where cryptocurrency is mined on a user's device without their knowledge of the practice, essentially stealing a user's processing power. To perform crowd mining or cryptojacking on apps and websites, *cryptomining libraries* are used. These make cryptocurrency mining easy and accessible for a large group of developers. For a long time, the cryptomining library *Coinhive* was the most popular library among crowd miners and cryptojackers. However, as of March 2019, Coinhive was discontinued due to the small profit margin of *Monero*, the cryptocurrency they mined. Today, it is unclear what library or service cryptojackers are using, and how large operations of cryptojacking are. This paper sets out to define the state of cryptojacking after the discontinuation of Coinhive.

## 1.1 Scope

Cryptojacking spreads in several ways: via websites, and also via malicious software for all modern computing platforms, from supercomputer networks to mobile devices. These cryptojacking mechanisms all have different ways of delivering cryptomining code, which can have different impact on the end-user. For instance, cryptojacking on mobile devices can cause rapid battery drainage and possible lasting damage to the hardware. In this thesis, we focus on cryptojacking on mobile devices, and more specifically on Android apps. Furthermore the focus is on illicit apps, which mine cryptocurrency without the knowledge of the user. It is also possible that app-owners are not aware of cryptojacking on their app, for instance when an advertisement service also mines cryptocurrency without informing their customers. These apps would also be considered.

To illustrate the problem, cybersecurity company McAfee published a report in 2019 claiming overall cryptojacking had increased by 29 percent [CB19]. SingCert published a tech report which showed mobile cryptojacking specifically is also on the rise, mentioning two large mobile cryptojacking campaigns [oS19]. Cybersecurity protection companies Symantec and ESET similarly reported a significant increase in mobile cryptojacking in March 2018 [AHAMHK19].

## 1.2 Research goals

The goals this research sets out to complete are the following:

**G1:** To investigate the status of existing libraries for Android CPU mining.

**G2:** To understand how mining libraries are integrated in APK's and determine how detectable the libraries are for anti-virus scanners.

**G3:** To evaluate economic incentives for mobile mining and cryptojacking nowadays.

## 1.3 Thesis overview

The thesis is organized as follows:

In Chapter 2, we review the background information needed to understand the technical details of cryptomining on mobile devices. It shortly discusses cryptocurrency as a concept, followed by an overview on mining and pools: digging into what the concepts mean, as well as how they work and why they are used. Furthermore libraries, and specifically the Coinhive library are discussed. Importantly, we estimate the size and profit of Coinhive as a whole. Then we explain the cryptocurrency Monero, which plays a central role in this thesis. And lastly the concept of cryptojacking is handled.

In Chapter 3 the first part of the research is described, where cryptomining libraries are identified and preliminarily estimated in size.

Following in Chapter 4 is the second part of the research, where the libraries are implemented in apps, and analyzed using these implementations. Furthermore, different ways of detecting cryptomining are discussed.

We then discuss the mining profit in Chapter 5, the third part of the research, where an estimation is made on the profit of a mining operation on mobile devices. The practice is compared to the past, as well as other ways of app monetization.

In Chapter 6, we discuss papers related to the topic and establish whether these have similar results. Lastly, Chapter 7 concludes the results of this thesis and mentions the further research that could be conducted, following this research.

# 2 Background

## 2.1 Cryptocurrency

The whole practice of cryptomining and cryptojacking revolves around cryptocurrencies. A *cryptocurrency* is a currency much like the euro or the dollar, except that it does not exist physically. Instead it uses cryptography and the blockchain technology to support a *cryptocurrency network* that keeps track of transactions and creation of the currency. This currency can be virtually kept track of with a *wallet address*, a unique code to a personal online "wallet", to which the cryptocurrency gets paid. While many people only know the popular Bitcoin, it is one of many cryptocurrencies. Cryptocurrencies launched after Bitcoin, are referred to as *Altcoins*. In this paper we will mostly discuss Monero, a cryptocurrency more suitable for web and Android mining, which is discussed in Section 2.3.

## 2.2 Mining and pools

To obtain cryptocurrency, it can either be bought with a traditional currency, traded for products and services, or it can be mined. The latter is an interesting way to make money for anyone owning a PC or smartphone. *Cryptomining* is the practice of solving computationally complex puzzles to support a cryptonetwork, receiving a fixed amount of cryptocoin in return. Solving such a puzzle, also referred to as *proof of work*, is essentially coming up with a nonce, a random number. This nonce should, jointly with the other fixed data, produce a hash that is less or equal to the target hash. This fixed data contains [Alo20]:

- The block number of the currently mined block
- The content of the block: all the transactions within the block
- The hash of the previous block
- The nonce, a random number

This structure is used to make it impossible for contributors to add false information to the blockchain in exchange for a reward.

The computations to add blocks are generally made to verify transactions and issue new cryptocoins. Once the miner has verified a specific amount of transactions, referred to as a *block*, they are eligible for a reward. This block is then added to the *blockchain* of the network, keeping track of all the transactions and coins. In the Monero network, block size is variable depending on the previous size of blocks and the current necessity of a larger block, while in other cryptocurrency networks, such a Bitcoin, there is a set size for blocks that cannot change over time (1 MB in the Bitcoin network).

There is an intricate system for controlling rewards and speed of solving blocks in the cryptocurrency networks. In a way, every time a block is solved, some variables are edited so the reward and time to solve a block is steadied. The rate at which a processing unit generates hashes is the *hashrate*, measured in hashes per second. As a way of defining the reward, there is a *target* for each block, which is the hash that should be estimated by the mining software. The target is defined by the

*difficulty*: how difficult it is to find a hash below the target or more specifically, the difficulty of guessing the hash in one try. In Monero mining, the target is adjusted with each new block. The adjustment algorithm looks at 720 blocks before the current and uses the time in which the block was solved, which should be ten minutes, to calculate the new target [Alo20]. As mentioned, these changes to the target are made to steady rewards, making higher rewards per block when the previous blocks took longer to solve, and lower when they took shorter. Thus, when more miners compete for a solution, the difficulty will rise.

However, a miner is not always rewarded for verifying a block of transactions. He/she also has to be the first miner to solve the numeric problem. Once a miner has found the correct solution, it is broadcasted over the network and other miners check if the solution is correct. When it is correct and checked by the network, the new block is added to the blockchain.

This is where *mining pools* come into play. These pools are groups of miners that share the computational workload of mining and split the reward between them. Using mining pools among other things, decreases the importance of luck (i.e. the chance to stumble upon the correct solution) in mining. There are several benefits to mining pools.

- Steady income. Since mining income is partly decided by luck, due to proof of work, the income of solo mining is often not so steady. In pools, the aggregated computational capacity of the miners increases greatly, thus increasing the chances of the pool to win in mining each block, with a more constant flow of money coming in.
- No entry barriers. When starting solo mining, it often takes a while to generate profit, due to the competition between miners. When using a pool, even though rewards are split, a miner will immediately generate profit.
- Faster processing. Faster processing in pools is mostly due to the method of discovery. When a block is found, other miners have to agree to its discovery. Since the miners in a pool are already in the same network, it speeds up the discovery process, causing overall faster mining.

The protocol used by most mining pools for communication between the pool and the individual miner is Stratum. This handles the authentication between pool and miner, receiving the puzzle that needs to be solved, as well as handling the announcement of the result.

Mining of cryptocurrency is usually performed on PC's. Although it is most efficient on GPU's, by running specifically made software for mining, it can also be done on a CPU. This is done either via downloadable software, or it can be mined via the web. With *web mining*, the mining code is embedded in the website, and whenever the website is open on a PC, cryptocurrency will be mined.

## 2.3 Monero

Bitcoin is not the only, and not always the most useful cryptocurrency around. The most popular cryptocurrency for web mining is Monero [MSH+18]. Monero is a private, decentralized cryptocurrency, abbreviated by XMR. It uses the Cryptonight algorithm, like other, often web-mined cryptocurrencies, such as Electroneum and Bytecoin. There are several advantages to Monero.

- Anonymity. While other cryptocurrencies, like Bitcoin, are pseudo-anonymous – all blocks, transactions and addresses are public, just not the name of the person who makes the transactions – Monero is completely anonymous. When making a transaction, the receiver is unable to see what address the transaction came from, making the sender completely anonymous.

- Faster transactions. This is mostly due to Monero's adaptable block size. When transactions are made, they are sometimes too big for the current block (e.g. with Bitcoin's 1 MB block limit) and therefore have to wait to be processed. In the Monero network, a transaction can always fit in the current block since the size is variable.

- Suitable for CPU's. Unlike other cryptocurrencies, such a Bitcoin, Monero does not depend on application-specific integrated circuits (ASIC's) to support its network. These are chips that are specifically made for mining to save time and energy, often needed to solve the hashes that Bitcoin uses. Although CPU mining started with Bitcoin, to make mining more accessible, new cryptocurrencies like Monero are tailored for CPU's. The Monero platform does not prefer ASIC's, since it is designed to use small contributors with algorithms that are more suitable for CPU's. This native Altcoin mining has proven to be 1.5 times faster than Bitcoin CPU mining [ELMC18].

## 2.4 Libraries, services and apps

At the start of the cryptomining era, it was profitable for end users to mine on their own devices, even on smartphones. Smartphones are inherently a computing platform with a large variety of third-party software, applications. For several years, Android has been the dominant mobile platform, featuring 2.5 billion active devices as of 2019 [Bra19] and 2.56 million apps in Google Play as of 2020 [Cle20]. Given the popularity of Android devices, many end-users wanted to try mining on their devices, making the demand for individual Android mining apps high.

To implement mining on Android apps, most developers use *libraries*. These are a collection of functionalities for software or an application. There are a great number of free or fee-based options for cryptomining libraries easily downloadable and implementable. *Cryptomining services* facilitate the participation in cryptomining by providing an API or portal for end-users to easily engage in mining. Some services offer their own libraries to facilitate integration of mining into applications. They may also rely on third-party libraries that can connect to their API's. The libraries are essentially the software that implements the API of the mining service. They come with easily adjustable settings, like the amount of processing power it should use, or whether mining should happen in the background, making it easier for applications to mine unnoticed. It is possible to mine different types of cryptocurrency with a library, usually they are not tied to a single cryptocurrency network. In the scope of this thesis we consider mining services and libraries to be the same thing, as they come from the same people with the same intentions. They are not to be confused with a cryptocurrency network, since this is tied to a single cryptocurrency, or a pool, in which many different services can join.

Android apps are usually distributed via Google Play, an online market by Google. However, Google no longer allows mining apps to be distributed via the Play Store [Goo20]. Even though some mining apps are not recognized by the store and therefore still obtainable in this way, most mining

apps are distributed via alternative markets. The markets offer downloadable APK's directly. APK's are Android packages, the format used to distribute and install applications onto Android devices.

Furthermore, miner apps can be divided into types: illicit, legitimate and scam apps. A *legitimate* mining app clearly show that they are mining or specifically ask for the user's consent. *Illicit* apps attempt to hide mining processes from the user. This is done by a cryptojacker, for instance, making an app that seems like it is only running an in-app game, while mining in the background without the user's knowledge. A third kind of app is a scam miner. These only pretend to mine, but do not actually deliver. [DZG+20]

Another way to categorize mining apps is by the type of library they use: binary or JavaScript. *Binary miners* are programs where the mining code is packed into a binary; which is a piece of machine code which is compiled and ready to run. The application code invokes the binary code, which runs the mining library. *JavaScript miners* are programs where the mining code is written in JavaScript. These are originally made to mine on the web, by adding the piece of JavaScript code to the existing web code. Although, it is also possible to use specific methods which allow this web code to be embedded in applications. These methods are described in Section 4.1.

## 2.5 Cryptojacking

Using all the previously described concepts, cryptojackers try to hide cryptomining code on victims' devices, making them unknowingly perform cryptographic calculations, while the profit goes towards the hacker. This means that rather than being a one time heist, a cryptojacking attack provides a continuous income to the hacker. Cryptojackers most commonly target PC's, mostly via websites, but also through dedicated malicious executables. However, cryptojacking on mobile devices, such as smartphones, is also becoming increasingly more common. Users could download apps without knowing of its mining capabilities, possibly causing a victim's phone to mine in the background 24/7 without their knowledge. The next generation of cryptojacking might take place on IoT (internet of things) devices, such as smart household devices or video cameras, connected to WiFi. This is often done using botnets, networks of CPU's controlled as a group by one owner. In this manner all of the victim's smart household devices could be mining cryptocurrency for an unknown attacker, without anyone ever finding out.

Besides the hacking of individual devices, cryptojackers are also known to hack companies' servers and websites. For example, it was reported that Tesla's Amazon Webservers infrastructure was running mining malware. Attackers allegedly found their way into an administrative portal, which included login credentials for the broader web-servers, where mining code could be buried [New18]. Recently a cryptomining campaign targeted supercomputers across Europe. They, among other things, mined Monero in Supercomputer Centres, most likely for financial reasons [Arg20]. Websites can be hacked by finding different vulnerabilities. For example, the Make-A-Wish website, which was injected with mining code in 2018 due to their content management system having a critical vulnerability. This allowed hackers to inject malicious code into all websites which use the vulnerable software [Bar18].

However, if cryptojacking always happens unnoticed, why does it form a threat? Firstly, it may slow

down other processes on a device. If a CPU is busy cryptomining, it might have less CPU power left for other processes and might therefore be slower. Secondly, since cryptojacking is not regulated, a user cannot know if the processes performed on its CPU is actually bad for the hardware. The CPU might get damaged by continuously mining, causing the breakage of victims' devices. Thirdly, CPU mining uses energy. This may be an insignificant amount on a small scale, but if large servers are mining, the amount of extra energy needed can be large and run up to electricity bill too.

## 2.6 Coinhive

Coinhive used to be the main threat among cryptojacking services. It was primarily a mining service, but it also had its own library. Its ease of use and reliability caused a fast spread among miners and more notably, cryptojackers. Coinhive, like most web-based mining libraries, mined Monero. It was easily implementable by placing a small piece of JavaScript code into a website or an application. An example of such a piece of code can be found in listing 1. It then used a part of the processing power of any device using the service that the website or application provides, to mine Monero for the cryptojacker. The cryptojacker got 70 percent of the eventual block reward, while Coinhive kept the remaining 30 percent. Even though it seems like an illicit service, Coinhive pitched their service as a way for website owners to earn money off of their users without intrusive advertisements, with their slogan *"Monetize Your Business With Your Users' CPU Power"* [Coi18] However, Coinhive did not force permission for mining by users, therefore making it interesting for cryptojackers who want to mine as unnoticed as possible.

Listing 1: Coinhive mining script example

```
<script src = "https://coinhive.com/lib/coinhive.min.js"></script>
<script>
    var miner = new Coinhive.Anonymous('CLIENT-ID', { throttle : 0.9});
    miner.start();
</script>
```

Determining the size of Coinhive at its height is difficult. As of now, according to publicwww.com 8305 websites are still running Coinhive code [pub20]. A study from 2018 concluded that around 80 percent of cryptojacking websites used Coinhive scripts [SKM18]. A paper written when Coinhive was still online did a search where approximately 60 percent of the generated database of mining Android applications used Coinhive [DZG+20]. At any rate, it is safe to say Coinhive was the most used cryptomining library at the time of its discontinuation.

As for profitability, a 2018 study suggests that Coinhive, at that time, mined around 150.000 USD worth a month [RZWH18]. Considering Coinhive keeps 30 percent of the profit, this adds up to over half a million USD annually. The study also discovered that Coinhive contributed $\tilde{1}.18$ percent of the mining power in the Monero network, forming more evidence that Coinhive was the leader in mining libraries.

# 3 Identifying libraries

To get a basic idea of the cryptomining landscape, it is important to know which mining libraries are out there. The first step of the research is to identify which CPU cryptomining libraries are being used by cryptojackers. To find this out, a web research was conducted. Cryptomining library names were gathered from previously written papers, forums and GitHub. Some libraries, like CoinImp and CryptoLoot, have a great online presence and are recommended on a large number of forums. This is most likely due to these libraries being more commercial, i.e. having a well maintained website and business model, and therefore spending more effort on advertising and spreading the library. Some libraries are less present on forums, but more often used by mining pools, gaining users through these communities and websites. An example of such a library is XMRig, a binary library located on GitHub.

When identifying a library, it is important to evaluate its activeness. This is done by finding active users. We consider the following indicators of activity: the software repository (e.g. GitHub) of the library has been recently edited; there is an active mining pool online; forums are actively discussing the library; the library has a well maintained website. With these methods a list of active mining libraries was compiled as of March 2020. Table 1 presents our findings.

| Library | Type | URL |
|---|---|---|
| Cryptoloot | JavaScript | https://crypto-loot.org/ |
| CoinImp | JavaScript | https://www.coinimp.com/ |
| JSEcoin | JavaScript | https://jsecoin.com/ |
| Deepminer | JavaScript | https://github.com/deepwn/deepMiner |
| Moonify | JavaScript | https://moonify.io/web-miner |
| CoinNebula | JavaScript | https://coinnebula.com/ |
| Rhino Miner | JavaScript | https://monerominer.rocks/ |
| Webminerpool | Binary / JavaScript | https://github.com/notgiven688/webminerpool |
| XMRig | Binary | https://github.com/XMRig/XMRig |
| CGminer | Binary | https://github.com/ckolivas/cgminer |
| MinerD | Binary | https://github.com/pooler/cpuminer |

Table 1: Active Mining libraries in March 2020

Interestingly, by August 2020, four of the found libraries have already been (partially) discontinued. CoinImp has stopped mining Monero due to it being unprofitable [1]. JSEcoin has discontinued all its services by the 21st of April 2020. Their reason being *"Under the current economic environment it has not been possible to raise the funding required to continue the JSE project."* [2]. Moonify has stopped its mining services and continued using their cryptocurrency for a gaming platform. While CoinNebula has gone offline all together [3].

[1] https://www.coinimp.com/news/coinimp-will-no-longer-support-monero-xmr-coin-mining
[2] https://jsecoin.com
[3] https://coinnebula.com/

To get a grasp on which libraries are used most, a web research was performed, finding the included files per JS library on websites. Publicwww finds snippets and keywords in web pages' HTML, JS and CSS code and returns the amount of times the keyword has been found. Because mining libraries are often invoked using the same code snippet, they are easily countable using Publicwww. Even though Coinhive was discontinued over a year ago, it is still included in this table, to give an idea of the size of it, compared to other libraries. It is important to note that these keywords were counted over a year after the discontinuation of Coinhive and it still has the largest online presence. The snippets of Coinhive that are still found are evidently not functioning, since the server is offline. Table 2 presents our findings on the frequency of specific keywords related to JavaScript libraries as found by Publicwww. Note that we do not include frequencies of binary libraries, as they are by definition not searchable through a website code.

| library | keyword used | # keywords found |
|---|---|---|
| coinhive | "coinhive.min.js" | 8305 |
| jsecoin | "load.jsecoin.com" | 531 |
| cryptoloot | "crypta.js" | 335 |
| webminerpool | "webmr.js" | 139 |
| crypto-webminer | "EverythingIsLife(" | 95 |
| coinnebula | "coinnebula.Instance" | 38 |
| deepMiner | "deepMiner.min.js" | 33 |
| moonify | "moonify.min.js" | 10 |
| rhino miner | "monerominer.rocks" | 10 |

Table 2: Libraries found on the web as of June 2020

CoinImp hosts individual JavaScript files for each user, making it hard to track their online presence. Interestingly, as of now, over a year after the discontinuation of Coinhive, no library has taken off as largely as Coinhive has in the past, even though there has been plenty of time for another library to take over the market. This seems to be an indication of the decreasing profit of Monero, as well as the decreasing interest in cryptomining overall.

Furthermore, it is interesting to know how cryptojackers choose which mining libraries to embed in their payload. Based on our analysis of mining services websites and related discussions on the internet, we have identified the following criteria. In future work, it would be interesting to validate these criteria and investigate the security economics landscape of cryptomining services in the context of cryptojacking.

- Fee and profit:
  For a user to choose a service, the fee is often one of the most important considerations. When mining with a library, the owners of the library usually get a percentage of the mined cryptocurrency. This is called the fee. They also take into account how much profit they will make using a library. Although outside of the fee, services do not have much difference in eventual profit, the consideration to start mining can depend on the estimated profit as well. An estimation of overall profit is made in Chapter 5.

- Ease of use:
  Most users will recommend other users services due to their ease of use. This is important, since a lot of users lack the programming skills to deal with complicated instructions. Therefore, most JavaScript crypto mining services offer users a small piece of code to inject into their JavaScript, which will easily activate the mining for them. Others also offer a WordPress plugin for websites. When considering ease of use, a service using a JavaScript library might be preferred over a binary library, since these are usually harder to implement.

- Opt-in or not:
  Another thing to consider for crypto miners, especially for the illicit ones, is whether a mining library requires the user to opt-in for mining happening on their device. Some mining libraries try to avoid illicit mining by forcing the users to opt-in with a pop-up. Illicit miners will evidently not choose these libraries so they can mine in the background without the user's knowledge.

- Detectability:
  When choosing a service, cryptojackers will want the library to be as little visible as possible. Since Google Play does not allow cryptomining apps to be in the Play Store, cryptojackers have to obfuscate their code to be able to offer it in the store. In this case, binary libraries are preferred, since this mining code is harder to detect. The user also should not notice the cryptojacking, this is explained in Chapter 4. It should slow the device down as little as possible, while still using as much processing power as it can without being noticed.

The results of the web conclude that no competitor or successor has taken over the market as much as Coinhive has in the past. With respect to our research goal G1, no library has proved as prevalent on the web as Coinhive was, as is shown by the results of the keyword search. Moreover, it seems cryptomining libraries are starting give up, as more and more services discontinue their practice.

# 4 Cryptomining Libraries Analysis

To research the environment of cryptojacking on Android applications, firstly the patterns of such apps need to be found. In order to do that, the libraries were integrated in Android apps. It is important to treat JavaScript and binary application differently. This is because JavaScript code is easy to read, and make out the function of the code, while binaries are much harder to interpret for humans. After finding the patterns, methods of detection can be found. We evaluate the methods of detection and assess whether the increasing and improving detection methods contribute to the decreasing use of cryptomining libraries and services.

## 4.1 Implementing JavaScript mining

JavaScript libraries are relatively easy to implement. Most services offer a simple guide with copy-pasteable code that anyone could inject into their website or application. The more commercial services have a system where the user fills in their email and the injecteable code gets sent to them personally. These usually also offer a WordPress plugin, making it even easier to use. The JS libraries based on GitHub have the injecteable code in a README file. Usually, they also contain a large amount of parameters to change in the mining process. The implementation of JS libraries in apps is slightly harder than the implementation of websites, due to Android apps not having an option to implement JavaScript code natively.

To implement JavaScript mining in a Java-based Android application, the Android API `loadUrl(...)` is used to implement a piece of HTML code, containing the relevant injecteable JavaScript code. Note that DOM Storage (Web storage) needs to be enabled for the JavaScript code to work. An example of such a code can be seen in Listing 2.

Listing 2: Java HTML injection

```
webView.getSettings().setDomStorageEnabled(true);
webView.getSettings().setJavaScriptEnabled(true);
webView.loadUrl("file:///android_asset/index.html");
```

As for the JavaScript code snippets, these can usually be found on the website or GitHub of the mining library. In the example (Listing 3) the server needs to be given, as well as a pool to mine in (no parameter for solo mining) and a Monero wallet address consisting of 95 characters. The password, threads and user value are all optional. The throttle is set to the amount of CPU power that needs to go into mining. Setting the throttle to 20 will limit the CPU usage to approximately 80 percent.

Listing 3: JavaScript mining code example

```
<script src="https://monerominer.rocks/miner-mmr/webmnr.min.js"></script>
<script>
    server = "wss://f.xmrminingproxy.com:8181";
    var pool = "moneroocean.stream";
    var wallet_address = <wallet address>;
```

```
        var password = "";
        var threads = -1;
        var user = "x";
        startMining(pool, wallet_address, password, threads, user);
        throttleMiner = 20;
    </script>
```

To test the detectability of JavaScript miners, we implemented two of the libraries in 'Hello World'-type apps. The services used here are Coinhive and the Rhino miner. Coinhive uses its own Coinhive SDK library, and Rhino miner uses the webmnr library. These were chosen to compare the detectability of the most common library Coinhive, to that of a lesser known library. When implementing, compiling and deconstructing the 'Hello World' JavaScript apps, it was extremely easy to classify them as mining apps. Since the HTML file could be recovered completely, the mining code was immediately visible. To evaluate the detectability of this mining code, we have uploaded the test APK's to VirusTotal (virustotal.com), a website aggregating 60 virus scanners, managed by Google. Out of the 60 scanners, only one classified the test APK as malicious. This is most likely due to the used miner not being a very commonly used one; the scanners' databases do not yet contain the necessary strings. However, when implementing an app with a (now outdated) Coinhive script and running it through virus scanners, 31 out of 60 recognized it as a potential threat. The results of the VirusTotal analysis can be seen in Figure 1.

Using a database of common strings in cryptomining libraries, JavaScript libraries could be easily detected by virusscanners. The research shows that new, rather unknown libraries might not be detected with this method, since the common strings in these libraries are not yet known to the scanners. However, as soon as the library is more popular, scanners will pick up on it, making them extremely easy to detect because JavaScript files are easily readable after deconstructing an application. The ease of this detection could be a reason for hackers to quit cryptojacking, since easy detection causes an easy ban on cryptojacking.

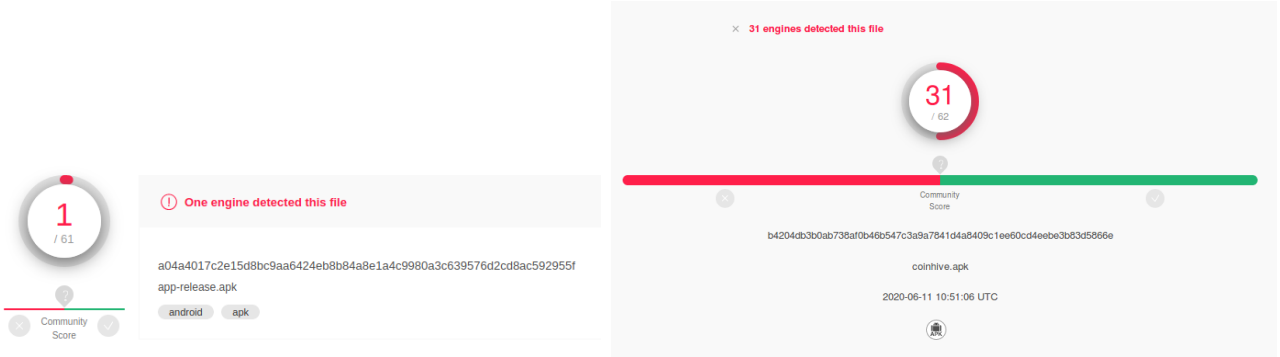


Figure 1: Rhino and Coinhive app analyzed

Listing 4: XMRig config file

```
{
    "algo": "cryptonight",
    "api": {
        (...)
    },
    (...)
    "background": false,
    (...)
    "donate-level": 5,
                (...)
    "max-cpu-usage": 75,
    "pools": [
        {
            "url": "pool.hashvault.pro:3333",
            "user":<wallet address>,
            "pass": "x",
            "keepalive": true,
            "nicehash": false,
            "variant": -1,
            "tls": false,
            "tls-fingerprint": null
        }
    ],
    (...)
}
```

## 4.2 Implementing binary mining

Binary libraries can be implemented using a native C++ library. The library code can usually be found on GitHub, where the README file contains instructions to compile a static library. To load native code from the shared or static library, the Android API `System.loadLibrary(...)` can be used.

To analyze binary mining apps, we use an existing app, which employs the XMRig library. XMRig was chosen because it came out as the most commonly used binary mining library in another research into Android cryptojacking [DZG+20]. The source code of the app can be found on GitHub [Eo18]. To verify the accuracy of the app, making sure to not analyze a scam app, it was compared to other XMRig apps on the web, which have also been used to find the patterns of XMRig code in applications. When running the app through VirusTotal, 32 of the scanners detected the malicious code as shown in Figure 2

When using XMRig, and this is the case with most binary miners, a config file is used to set the mining address and where to mine to. The algorithm is set, to CryptoNight in this case, to mine

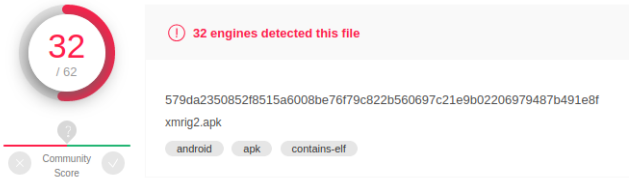


Figure 2: XMRig app analyzed

Monero. The config file can also set whether the algorithm should mine in the background (i.e., not visible to the device user), although the norm is to not do it. The amount of donation to the original owner of the program can be set manually, as well as the maximum of CPU percentage that can be used by the miner. As for the pool, a pool address and wallet address need to be given, similarly to the JS code. This can all be adjusted in the config file, an example can be seen in Listing 4.

The most important pattern in XMRig apps is the XMRig binary file. To make sure the app is not a scam app, it is important to check whether the binary is being called in the smali code. Smali code is used to interpret the binary code of an APK. A disassembler takes binary parts of the APK and translates them into human-readable smali code.

To test whether the binary is an XMRig binary, we constructed a Yara ruleset. Yara is a well-known notation and a tool to find patterns in code based on string search (reference), finding textual and binary patterns. We have applied the Yara notation and have produced a rule to detect an Android APK with an XMRig binary included. To make such a ruleset, firstly patterns in the XMRig binaries have to be found. In order to do so we compared a few known XMRig binaries and searched for patterns regarding significant attributes of the XMRig library, based on fee, the stratum protocol and the pools used by XMRig. These strings were found using IDA Pro, a disassembler which can display binary code in readable text patterns. The rule is displayed in listing 5. It was published on Koodous (koodous.com) [4], a collaborative platform for Android malware research, where contributors can upload rulesets to detect malware, as well as upload possibly malicious APK's to find out its malicious properties.

Listing 5: Yara rulset XMRig

```
import "cuckoo"

rule xmrigStrings
{
    strings:
        $fee = "fee.xmrig.com" wide ascii
        $nicehash = "nicehash.com" wide ascii
        $minergate = "minergate.com" wide ascii
        $stratum = "stratum+tcp://" wide ascii
```

[4] https://koodous.com/rulesets/6994

```
    condition:
       $fee and
       $nicehash and
       $minergate and
       $stratum
}
```

## 4.3 Other ways of library detection

The downside of detecting libraries by means of keyword search, is that legitimate mining apps are also found and labeled as malicious. This is called a *false positive*, where legitimate apps are labeled as malware. Due to obfuscation of mining code, *false negatives* can also occur. This is when an app does mine but it is not detected. Additionally, in order for this detection approach to work, all used libraries need to be known by the detector, for it needs keywords for each library separately. A new mining library might go unnoticed in this approach, causing more false negatives. Other researchers have proposed different ways for mining to be detected, removing these obstacles.

### 4.3.1 Static approach

A different way of keyword search is by searching for patterns that occur in every form of crypto-mining. This can be done through:

- Mining credentials, such as wallet addresses and site keys. These are the kind of credentials that are loaded into most mining application, making it a more universal way to detect mining.
- Mining domains and keywords, potential mining domains and keywords such as Monero, Bitcoin or Stratum can be found in a large amount of potential miners. This method does detect a lot of false positives, since scam apps, as well as just miner-related apps often contain these keywords as well [DZG+20].

### 4.3.2 Dynamic approach

A way to implement a dynamic approach is by detecting mining by looking at metrics of CPU mining. These are indicators that might change drastically once a CPU starts mining in the background. An app, when mining, can for example use a great deal more CPU power as well as battery power. A collection of metrics like this could be a very good indicator for mining. A study [DZG+20] looking into this developed a dynamic prototype called BrenntDroid, determining that the two most important features for mining detection are Maximum CPU Utilization percentage and Average Battery Power. This method has shown to be 95 percent accurate.

Another dynamic method is by detecting the hashing algorithms of mining algorithms. These are so fundamental to the mining process, that they cannot possibly be obfuscated or somehow removed. Researchers developed a method called Minesweeper. This firstly uses fingerprints to detect the

cryptographic primitives that the Cryptonight algorithm uses. This is done by fingerprinting. Fingerprints consist of a cryptographic operation count, which can be compared to possible mining programs. The outcome of this comparison (with a 70 percent or higher threshold) can define whether an app is mining. Additionally, the cryptographic operations are counted and compared to a threshold, and CPU Cache Events are monitored, since during cryptomining, a higher load/store frequency occurs [KVM+18].

If these methods would be commonly implemented, it would be a great deal harder for cryptojackers to make money off of unsuspecting users. The dynamic approach might even make it completely impossible for mining to go unnoticed, even when using code obfuscation. Additionally, the Android operating system could forbid cryptomining without consent or opt-in from the user. The advancing detection of cryptomining software could be another reason of the decreasing interest in cryptojacking in general.

With respect to our research goal G2, JS mining libraries are relatively easily implementable by injecting a small piece of JS code, found on the libraries' website / GitHub, into the application code, using `loadUrl(...)`. Binary libraries need to be fully loaded into the application, after which the library functions can be called. JS libraries can be easily detected by anti-virus software as soon as the library is known. This can be done since the JS code is easily obtainable from an APK and compared to known strings in libraries. Binary libraries can be found using known patterns in the binary code of APK's, while we also discuss ways of dynamic detection and keyword search.


Figure 3: Monero worth in USD from 2017 to 2020 [Coi20]

# 5 Mining profit

## 5.1 Calculating mining profit

To get an idea of whether Android CPU mining is worth the effort, as well as the consumed electricity, we estimate the profit of mining. In the following calculations, we consider the specifications of a Samsung Galaxy S7 device, using a Qualcomm Snapdragon 820 processor. A study in 2018 shows an average mobile CPU hash rate of 5.04 hash/s [HCBL19]. This restricts that the device has internet access and the battery is above 15 percent. Due to the difficulty changing overtime, the hash rate should be slightly different as of now. The results of the previous study were measured in May 2018. The average Monero network hashrate during May 2018 was 444M hash/s, as can be seen in figure 5 [bit20a] and the measured network hashrate on June 16th 2020 1382M hash/s, making the current hash rate on a mobile device more likely to be around 15.69 hash/s, calculated in Figure 4. As of June 16, the block reward is fixed at 1.614349644556 XMR and the euro/XMR conversion is 58.68 euro/XMR [bit20b]. Putting this together, mining on a mobile device generates about 0.00077 euro in profit daily, as is calculated in Figure 4.

A way to put this into perspective is comparing it with the profit of a mobile mining device at the height of Monero's success, which was before the cryptocurrency crash in 2018. As can be seen in Figure 3, the height of Monero's value was around January 2018. To be more specific, the top of the peak lies at the 9th of January 2018, at a value of 542.33 USD per XMR (454.39 euro at the time, roughly 7 times higher than the value of Monero today) [Coi20]. Using the Waybackmachine (https://archive.org/web/), a website that archives old versions of websites, two values of block rewards at the time can be found. The rewards were 5.06 on 29-8-2017 [bit17] and 6.56 on 12-3-2018 [bit18]. Since the Monero reward decreases somewhat linearly, the right block reward can be extrapolated from these values to 6.083 XMR, see Figure 4. The result of these calculations is a daily profit of 0.02259 euro, roughly 28 times higher than Monero profit today.

Another way to put this profit into perspective, we can compare it to a different way of making money off of Android applications, advertisements. As seen in figure 6, the revenue for full screen advertisements (interstitials) is around 3 USD RPM, revenue per thousand views. This means that one view gets around 0.003 USD (or 0.0027 euro). As a way of monetization of Android apps, it

*Mobile hashrate 16-06-2020*

$$5.04/444 * 1382 = 15.69 hash/s$$

*Mobile hashrate 09-01-2018*

$$5.04/444 * 608.88 = 6.9116 hash/s$$

*Block Reward 09-01-2018*

$$(6.56 * 133 + 5.06 * 62)/195 = 6.083 XMR$$

*Profit in euro per day*
(device's hashrate) * (current block reward) * 720 / (network hashrate) * (euro/XMR)

*Profit per day on 16-06-2020*

$$0.00001569 * 1.614349644556 * 720/1382 * 58.68 = 0.000774346 euro/day$$

*Profit per day on 09-01-2018*

$$0.0000069116 * 6.083 * 720/608.88 * 454.39 = 0.02259 euro/day$$

Figure 4: Mining profit calculation


Figure 5: Monero hash rate historical chart [bit20a]

is most likely more favorable to use advertisements rather than Monero mining. A single ad will generate more profit than a miner running in the background for a whole day.

| Average Mobile RPMs | |
|---|---|
| iOS Banners | $0.20 – $2.00 |
| iOS Interstitials | $3.00 – $5.00 |
| Android Banners | $0.15 – $1.50 |
| Android Interstitials | $2.00 – $4.00 |
| General Trend | Down |
| *Source: MonetizePros Aggregation* | |

Figure 6: Advertisement revenue [Mon19]

When mining on a cellular network, the average cost of mining is 0.000219 USD per minute [PIM18], meaning the cost is around 0.315 USD per day (about 0.2803 euro). A user mining on cellular data, even when keeping the profits, would make a loss. Mining also, on average, consumes 2.08 times more energy than an ad-supported service [PIM18]. The discharge rate of mining is about 309.3 mAh. Considering the average price per KwH in the Netherlands is 22 cents [Con19], we made a calculation to put the energy cost of mining on the user's side into perspective in Figure 7. This is again more than the calculated daily profit, about 3.28 times as much. When a cryptojacker is using a user's smartphone for mining, they are essentially stealing from the victim.

Since the crash of the cryptocurrency market has made Monero less valuable, it is interesting to know at which point individual cryptomining should be profitable again. Considering that the end-user does not mine on a mobile network, and therefore only has to take the energy cost into account, we make a calculation. This concludes, as shown in Figure 8 that in order to not make a loss in the current circumstances, the value of Monero would have to be 250.08 euro/XMR.

*Difference in discharge rate*

$$309.3 - 309.3/2.08 = 160.60mAh$$

*Power Consumption*

$$Wh = mAh * V/1000$$

$$160.60 * 3.85/1000 = 0.618Wh$$

[ifi20]

*Cost*

$$0.618/1000 * 0.22 * 24 = 0.0033euro/day$$

Figure 7: Energy cost calculation

$$0.0033/(0.00001569 * 1.614349644556 * 720/1382) = 250.0742 euro/XMR$$

Figure 8: Monero price for end-user profitability

Even though a hacker might never make a loss on cryptojacking, the profit has been decreasing. If a hacker takes over a botnet and can make every device mine at the same time, 24/7, the collective profit might be a substantial amount. However, even monetization of Android applications through advertisements has proven to be more profitable than mining. Since mining profit was larger in the past, the decrease of popularity in mining could also be caused by the reduction of mining profit. This can be a decrease in interest on the user's side, with less individual profit, but as mentioned in Chapter 3, the decrease can also occur on the library's side.

With respect to our research goal G3, the current economic incentives of cryptomining are definitely too low for individual mining, since the income of a day of mobile mining is only about 0.0007 euro and the energy costs are around 3 times higher than this. The mining profit is also considerably lower than it was, estimating the mining profit at its height to be 28 times higher than it is now. As for cryptojacking, there might not be an actual loss, but advertisements are deemed to have a larger profit in general circumstances. For individual cryptomining to be profitable under the current circumstances, the price of Monero would have to be more than 250.08 euro.

## 5.2 Discussion

The results of the calculations indicate that solo cryptocurrency mining, as well as cryptojacking are not profitable enough at the time. This result is in line with the earlier chapters, concluding that interest in cryptojacking has decreased. These results add a new layer to the research, a new reason why cryptojacking is less popular, namely the low profit. However, the reliability of the data is impacted by the reliability of the numbers used from other researches. The mobile hashrate was taken from another research, but these hashrates vary per research, as well as per device, making the actual results vary. Though, the variation of the hashrate will not change too much in the eventual profit. The comparison between advertisement rates and cryptojacking profit might also be less reliable, due to the difficulty of calculating actual advertisement rates. The results are based on phones that mine 24/7, while most smartphone users might not have their device or WiFi on at all times, making 24/7 mining virtually impossible. Additionally, mining might be an easier way to generate profit, compared to advertisements, since mining apps do not have to be opened or viewed to make a profit; they can run in the background without the user's knowledge. In order to be more certain of the profit, a real-time test might need to take place to test the actual profit of mining and possibly compare it to a real-time advertisement profit test.

# 6 Related work

By far the most relevant paper, as well as inspiration of the thesis, is *Dissecting Android Cryptocurrency Miners* Dashevskyi et al. [DZG+20], which discusses Android CPU mining, creating, dissecting and analyzing a database of 728 Android cryptomining apps. The paper also describes a set of dynamic features and proposes a prototype to detect miners at run-time. A similar paper, *MineSweeper: An In-depth Look into Drive-by Cryptocurrency Mining and Its Defense* Konoth et al. [KVM+18] describes the difficulty of detecting miners by means of blacklists, string patterns or CPU utilization and proposes a detection technique that focuses on finding the cryptographic computations that mining software makes to produce valid hashes. The paper *CryptoCurrency Mining on Mobile as an Alternative Monetization Approach* Huynh et al. [HCBL19] describes the profit mobile mining makes in a similar way this thesis does. *Is Cryptojacking Dead after Coinhive Shutdown?* Varlioglu et al. [VGOB20] concludes similarly that, following the discontinuation of Coinhive, the business of cryptojacking is not as popular as before. *Digging into Browser-based Crypto Mining* Rüth et al. [RZWH18] looked into the most popular websites and found that less than 0.08 percent of them utilize browser mining, identify Coinhive as the largest web-based mining provider (at 75 percent of mining services).

# 7 Conclusions and Further Research

In this research we have investigated the current state of cryptojacking on Android devices, mainly looking into the overall popularity of cryptojacking on Android. The amount of the cryptojacking libraries and operations on Android seem to be decreasing since the discontinuation of Coinhive. Other mining libraries, such a Cryptoloot and XMRig still seem to be active, but have yet to take over a market segment like Coinhive did in the past. We empirically show it is relatively easy to implement cryptomining in an Android app. The detection of mining currently is shown to be rather poor. Virus scanners often do not detect miners in Android applications, even less when the mining service is not well-known. This is most likely due to the absence of economic incentives. The lack of detection might become dangerous once the cryptocurrency market will start booming again. Due to the currently low worth of cryptocurrencies, the banning of cryptomining apps from Google Play and the increasing efficiency of mining detection, the process of mining on Android devices is deemed unprofitable.

Furthermore, this research has shown that monetization using cryptomining on Android does not generate large profit margins, reaching only 0.0007 euro per day on an Android device. This profit is lower than the profit of showing a single full-screen advertisement on Android, as well as that it is 3 times lower than the electricity it costs to mine for this amount of time. The mining profit per day was 28 times higher at the height of Monero, while the value of Monero at that time was actually only 7 times higher.

In the future this research could be continued by improving cryptomining detection approaches on Android devices. This is increasingly important since the current detection of mining applications has proven to be poor. Static analysis poses a good solution, as it allowed to analyze application code without first running the app, and does not take up CPU power while the app is running.

Current static detection techniques, as e.g. proposed in [DZG+20], focus on string matching but do not explore code patterns of existing libraries. This too could be explored in future research. Additionally, it may be interesting to explore automatic mining detection at software repositories, i.e. at the moment of publishing on GitHub. This could help establish a more complete list of cryptomining libraries and services, making detection easier.

# References


[AHAMHK19] Haitham Hilal Al Hajri, Badar Mohammed Al Mughairi, Mohammad Imtiaz Hossain, and Asif Mahbub Karim. Crypto jacking a technique to leverage technology to mine crypto currency. *international journal of acedemic research in bussiness and social sciences*, 9(3), 2019.

[Alo20] Kurt M Alonso. Zero to monero: First edition. a technical guide to a private digital currency; for beginners, amateurs, and experts, 2020.

[Arg20] Ionut Arghire. Crypto-mining campaign hits european supercomputers, https://www.securityweek.com/crypto-mining-campaign-hits-european-supercomputers, accessed 11-07-2020, 2020.

[Bar18] Brian Barrett. Hack brief: Criminals with no shame hit make-a-wish website. https://www.wired.com/story/make-a-wish-website-cryptojacking-hack/, accessed 06-07-2020, 2018.

[bit17] bitinfocharts. Monero info in 29-8-2017, https://web.archive.org/web/20170829025717/https://bitinfocharts.com/monero/, accessed 30-06-2020, 2017.

[bit18] bitinfocharts. Monero info in 12-3-2018, https://web.archive.org/web/20180312070028/https://bitinfocharts.com/monero/, accessed 30-06-2020, 2018.

[bit20a] bitinfocharts. Monero hashrate historical chart, https://bitinfocharts.com/comparison/monero-hashrate.html, accessed 17-06-2020, 2020.

[bit20b] bitinfocharts. Monero info, https://bitinfocharts.com/monero/, 2020.

[Bra19] Russell Brandom. There are now 2.5 billion active android devices, https://www.theverge.com/2019/5/7/18528297/google-io-2019-android-devices-play-store-total-number-statistic-keynote, accessed 30-07-2020, 2019.

[CB19] John Fokker Steve Grobman Tim Hux Tim Polzer Marc Rivero Lopez Thomas Roccia Jessica Saavedra-Morales Raj Samani Ryan Sherstobitoff Christiaan Beek, Taylor Dunton. Mcafee labs threats report. Technical report, 2019.

[Cle20] J. Clement. Number of apps available in leading app stores as of 1st quarter 2020, https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/, accessed 30-07-2020, 2020.

[Coi18] Coinhive. Coinhive 2018, https://web.archive.org/web/20180515073251/https://coinhive.com/, accessed 22-08-2020, 2018.

[Coi20] Coingecko. Monero worth from 2017 to now, https://www.coingecko.com/nl/coins/monero, accessed 30-06-2020, 2020.

[Con19] Consumentenbond. Energie vergelijken, https://www.consumentenbond.nl/energie-vergelijken/kwh-prijs, accessed 23-06-2020, 2019.

[DZG+20] Stanislav Dashevskyi, Yury Zhauniarovich, Olga Gadyatskaya, Aleksandr Pilgun, and Hamza Ouhssain. Dissecting android cryptocurrency miners. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 191–202, 2020.

[ELMC18] Shayan Eskandari, Andreas Leoutsarakos, Troy Mursch, and Jeremy Clark. A first look at browser-based cryptojacking. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 58–66. IEEE, 2018.

[Eo18] ElijaxApps-org. android-xmrig-miner, https://github.com/ElijaxApps-org/android-xmrig-miner, accessed 07-07-2020, 2018.

[Goo20] Google. Google developer policy, https://play.google.com/about/developer-content-policy-print/, accessed 11-05-2020, 2020.

[HCBL19] Sinh Huynh, Kenny Tsu Wei Choo, Rajesh Krishna Balan, and Youngki Lee. Cryptocurrency mining on mobile as an alternative monetization approach. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, pages 51–56, 2019.

[ifi20] ifixit. Galaxy s7 battery, https://nl.ifixit.com/Store/Android/Galaxy-S7-Replacement-Battery/IF329-023?o=3, accessed 30-06-2020, 2020.

[KVM+18] Radhesh Krishnan Konoth, Emanuele Vineti, Veelasha Moonsamy, Martina Lindorfer, Christopher Kruegel, Herbert Bos, and Giovanni Vigna. 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*, pages 1714–1730, 2018.

[Mon19] Monetizepros. ad cpm rates, https://monetizepros.com/cpm-rate-guide/mobile/, accessed 30-06-2020, 2019.

[MSH+18] Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, et al. An empirical analysis of traceability in the monero blockchain. *Proceedings on Privacy Enhancing Technologies*, 2018(3):143–163, 2018.

[New18] Lily Hay Newman. Hack brief: Hackers enlisted tesla's public cloud to mine cryptocurrency, https://www.wired.com/story/cryptojacking-tesla-amazon-cloud/, accessed 06-07-2020, 2018.

[oS19] Cyber Security Agency of Singapore. Mobile threats. Technical report, 2019.

[PIM18] Panagiotis Papadopoulos, Panagiotis Ilia, and Evangelos P Markatos. Truth in web mining: Measuring the profitability and cost of cryptominers as a web monetization model. *arXiv preprint arXiv:1806.01994*, 2018.

[pub20] publicwww. Amount of coinhive online, https://publicwww.com/websites/%22coinhive.min.js%22/, accessed 11-05-2020, 2020.

[RZWH18] Jan Rüth, Torsten Zimmermann, Konrad Wolsing, and Oliver Hohlfeld. Digging into browser-based crypto mining. In *Proceedings of the Internet Measurement Conference 2018*, pages 70–76, 2018.

[SKM18] Muhammad Saad, Aminollah Khormali, and Aziz Mohaisen. End-to-end analysis of in-browser cryptojacking. *arXiv preprint arXiv:1809.02152*, 2018.

[VGOB20] Said Varlioglu, Bilal Gonen, Murat Ozer, and Mehmet Bastug. Is cryptojacking dead after coinhive shutdown? In *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, pages 385–389. IEEE, 2020.