

# MineSweeper: An In-depth Look into Drive-by Mining and its Defense

Veelasha Moonsamy  
Utrecht University, The Netherlands



28 August 2018  
University of Adelaide, Australia

# Utrecht University, The Netherlands



# Acknowledgment

- ▶ Joint collaboration:

## MineSweeper: An In-depth Look into Drive-by Cryptocurrency Mining and Its Defense

Radhesh Krishnan Konoth  
Vrije Universiteit Amsterdam  
r.k.konoth@vu.nl

Emanuele Vineti  
Vrije Universiteit Amsterdam  
emanuele.vineti@gmail.com

Veelasha Moonsamy  
Utrecht University  
email@veelasha.org

Martina Lindorfer  
TU Wien  
martina@iseclab.org

Christopher Kruegel  
UC Santa Barbara  
chris@cs.ucsb.edu

Herbert Bos  
Vrije Universiteit Amsterdam  
herbertb@cs.vu.nl

Giovanni Vigna  
UC Santa Barbara  
vigna@cs.ucsb.edu

- ▶ Paper available at: [www.veelasha.org](http://www.veelasha.org)

# Cryptocurrency: the rise of decentralized money

- ▶ A cryptocurrency:
  - is a digital asset designed to work as a **medium of exchange**

# Cryptocurrency: the rise of decentralized money

- ▶ A cryptocurrency:
  - is a digital asset designed to work as a **medium of exchange**
  - uses cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets

# Cryptocurrency: the rise of decentralized money

- ▶ A cryptocurrency:
  - is a digital asset designed to work as a **medium of exchange**
  - uses cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets
- ▶ In 2009, the first cryptocurrency, 'Bitcoin', was introduced

# Cryptocurrency: the rise of decentralized money

- ▶ A cryptocurrency:
  - is a digital asset designed to work as a **medium of exchange**
  - uses cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets
- ▶ In 2009, the first cryptocurrency, 'Bitcoin', was introduced
- ▶ Fast forward to 2018, about **1600** cryptocurrencies are in existence, out of which **more than 600** still see an active trade

# Cryptocurrency: the rise of decentralized money

- ▶ A cryptocurrency:
  - is a digital asset designed to work as a **medium of exchange**
  - uses cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets
- ▶ In 2009, the first cryptocurrency, 'Bitcoin', was introduced
- ▶ Fast forward to 2018, about **1600** cryptocurrencies are in existence, out of which **more than 600** still see an active trade
- ▶ An overall surge in market value across cryptocurrencies has renewed interest in *cryptominers*

# Cryptocurrency: the rise of decentralized money

- ▶ A cryptocurrency:
  - is a digital asset designed to work as a **medium of exchange**
  - uses cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets
- ▶ In 2009, the first cryptocurrency, 'Bitcoin', was introduced
- ▶ Fast forward to 2018, about **1600** cryptocurrencies are in existence, out of which **more than 600** still see an active trade
- ▶ An overall surge in market value across cryptocurrencies has renewed interest in *cryptominers*
- ▶ ... which in turn led to the proliferation of cryptomining services, such as **Coinhive** - introduced in September 2017

## From September 2017 onwards ...

It started with:



**UNICEF Is Mining Crypto to Raise Funds  
for Children**

'Our Cryptocurrency Mining Policy:  
Free Content, No Ads!'

# From September 2017 onwards ...

And things went downhill very quickly:

## Coinhive Is Rapidly Becoming a Favorite Tool Among Malware Devs

By [Catalin Cimpanu](#)

September 23, 2017 09:00 PM 3

## Cryptojackers Found on Starbucks WiFi Network, GitHub, Pirate Streaming Sites

By [Catalin Cimpanu](#)

December 13, 2017 09:25 AM



By [Helen Partz](#)

MAY 08, 2018

## Coinhive Code Found On 300+ Websites Worldwide In Recent Cryptojacking Campaign

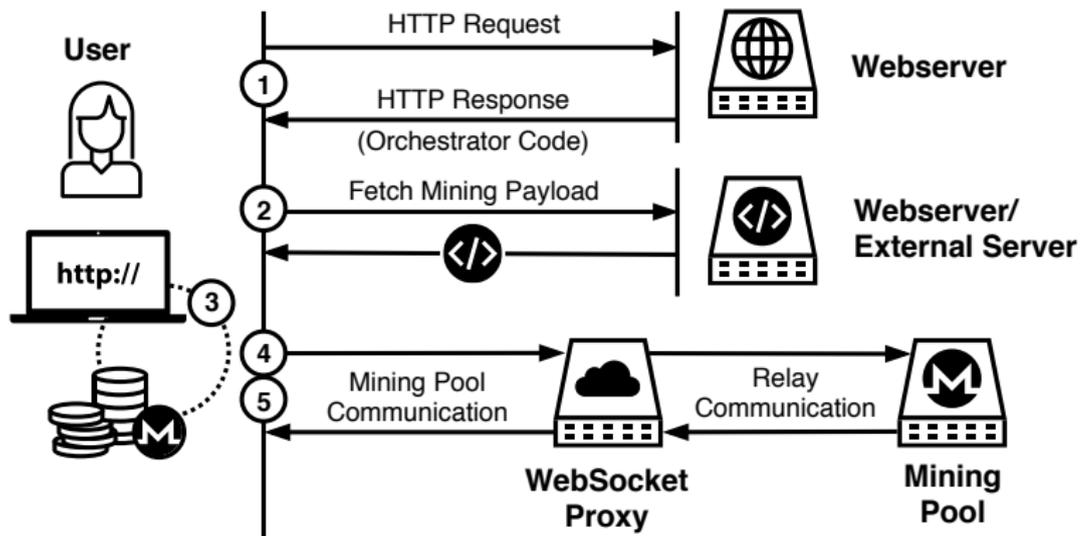
## Drive-by mining aka *Cryptojacking*

- ▶ Is a web-based attack
- ▶ An infected website **secretly** executes a mining script (Javascript code and/or WebAssembly module) in user's browser to mine cryptocurrencies
- ▶ Is considered **malicious** only when user does not explicitly give their consent

## Drive-by mining aka *Cryptojacking*

- ▶ Is a web-based attack
- ▶ An infected website **secretly** executes a mining script (Javascript code and/or WebAssembly module) in user's browser to mine cryptocurrencies
- ▶ Is considered **malicious** only when user does not explicitly give their consent
- ▶ In this work: we study the prevalence of drive-by mining attacks on Alexa's Top 1 million websites

# Threat Model



# Current detection methods

Two main approaches have been used:

1. Blacklist-based approach

# Current detection methods

Two main approaches have been used:

1. Blacklist-based approach
  - ▶ Not scalable

# Current detection methods

Two main approaches have been used:

1. Blacklist-based approach

- ▶ Not scalable
- ▶ Prone to high false negatives

# Current detection methods

Two main approaches have been used:

1. Blacklist-based approach

- ▶ Not scalable
- ▶ Prone to high false negatives
- ▶ Easily defeated by URL randomization and domain generation algorithms

# Current detection methods

Two main approaches have been used:

1. Blacklist-based approach
  - ▶ Not scalable
  - ▶ Prone to high false negatives
  - ▶ Easily defeated by URL randomization and domain generation algorithms
2. High CPU-based approach

# Current detection methods

Two main approaches have been used:

1. Blacklist-based approach

- ▶ Not scalable
- ▶ Prone to high false negatives
- ▶ Easily defeated by URL randomization and domain generation algorithms

2. High CPU-based approach

- ▶ False positives, as there might exist other CPU-intensive use cases

# Current detection methods

Two main approaches have been used:

1. Blacklist-based approach

- ▶ Not scalable
- ▶ Prone to high false negatives
- ▶ Easily defeated by URL randomization and domain generation algorithms

2. High CPU-based approach

- ▶ False positives, as there might exist other CPU-intensive use cases
- ▶ False negatives, as cryptominers have started to throttle their CPU usage to evade detection

# Contributions

- ▶ Perform first in-depth assessment of drive-by mining

# Contributions

- ▶ Perform first in-depth assessment of drive-by mining
- ▶ Discuss why current defenses based on blacklisting and CPU usage are ineffective

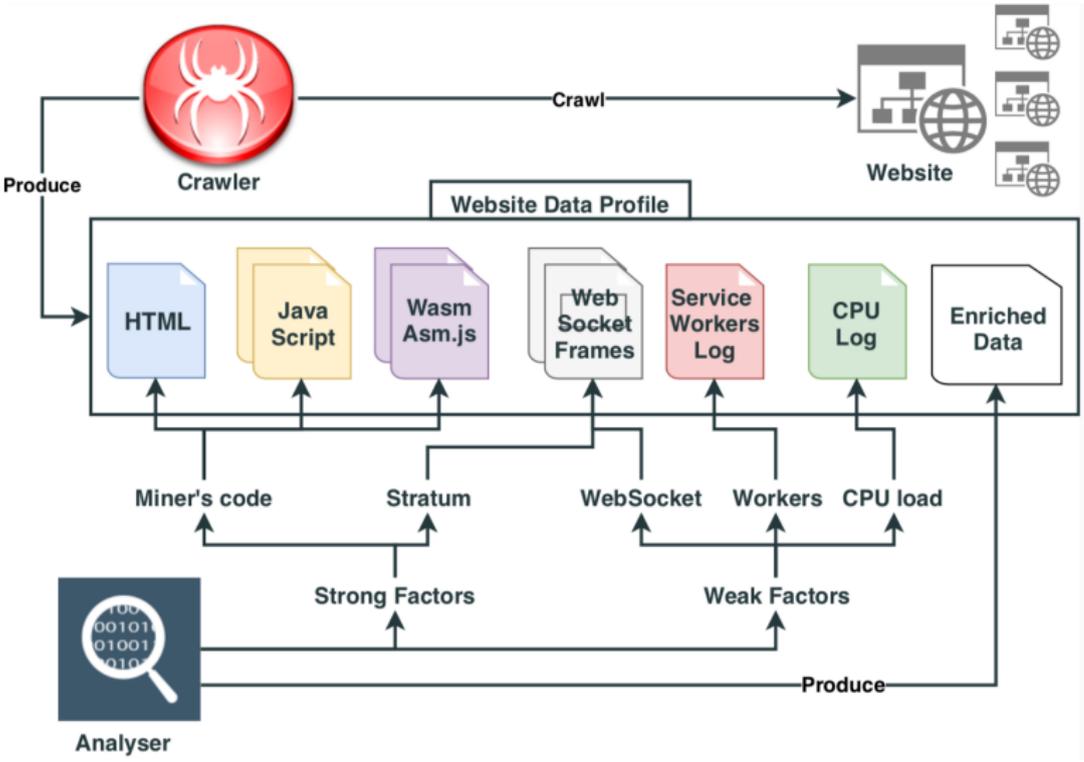
# Contributions

- ▶ Perform first in-depth assessment of drive-by mining
- ▶ Discuss why current defenses based on blacklisting and CPU usage are ineffective
- ▶ Propose **MineSweeper**, a novel detection approach based on the identification of the cryptographic functions (static analysis) and cache events (during run-time)

# Drive-by mining in the wild

- ▶ Conducted a large-scale analysis with the aim to answer the following questions:
  1. How prevalent is drive-by mining in the wild?
  2. How many different drive-by mining services exist currently?
  3. Which evasion tactics do drive-by mining services employ?
  4. What is the modus operandi of different types of campaign?
  5. How much profit do these campaigns make?
  6. What are the common characteristics across different drive-by mining services that can be used for their detection?

# Large-scale Analysis: experiment set-up



# Data collection

- ▶ Over a period of one week in mid-March 2018

# Data collection

- ▶ Over a period of one week in mid-March 2018
- ▶ Crawler
  - ▶ Crawled landing page and 3 internal pages
  - ▶ Stayed on each visited page for 4 seconds
  - ▶ No simulated interaction, i.e. the crawler did not give any consent for cryptomining

# Data collection

- ▶ Over a period of one week in mid-March 2018
- ▶ Crawler
  - ▶ Crawled landing page and 3 internal pages
  - ▶ Stayed on each visited page for 4 seconds
  - ▶ No simulated interacted, i.e. the crawler did not give any consent for cryptomining
- ▶ Crawled 991,513 websites; 4.6 TB raw data and 550 MB data profiles

## Preliminary results: Cryptomining code (1/2)

- ▶ Recall: cryptomining code consists of *orchestrator code* and *mining payload*

## Preliminary results: Cryptomining code (1/2)

- ▶ Recall: cryptomining code consists of *orchestrator code* and *mining payload*
- ▶ Identification of orchestrator code

## Preliminary results: Cryptomining code (1/2)

- ▶ Recall: cryptomining code consists of *orchestrator code* and *mining payload*
- ▶ Identification of orchestrator code
  - ▶ Websites embed the orchestrator script in the main page

## Preliminary results: Cryptomining code (1/2)

- ▶ Recall: cryptomining code consists of *orchestrator code* and *mining payload*
- ▶ Identification of orchestrator code
  - ▶ Websites embed the orchestrator script in the main page
  - ▶ Can be detected by looking for specific string patterns

## Preliminary results: Cryptomining code (1/2)

- ▶ Recall: cryptomining code consists of *orchestrator code* and *mining payload*
- ▶ Identification of orchestrator code
  - ▶ Websites embed the orchestrator script in the main page
  - ▶ Can be detected by looking for specific string patterns

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

    miner.start();
</script>
```

## Preliminary results: Cryptomining code (1/2)

- ▶ Recall: cryptomining code consists of *orchestrator code* and *mining payload*
- ▶ Identification of orchestrator code
  - ▶ Websites embed the orchestrator script in the main page
  - ▶ Can be detected by looking for specific string patterns

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

    miner.start();
</script>
```

- ▶ Keywords: `CoinHive.Anonymous` or `coinhive.min.js`

## Preliminary results: Cryptomining code (2/2)

- ▶ Identification of mining payload
  - ▶ Dump the Wasm (WebAssembly) payload
  - ▶ `-dump-wasm-` module flag in Chrome dumps the loaded Wasm modules
  - ▶ Keyword-based search: `cryptonight_hash` and `CryptonightWasmWrapper`

# Effectiveness of fingerprint-based detection

<b>Mining Service</b>	<b>Number of Websites</b>	<b>Percentage</b>
Coinhive	514	59.35%
CoinImp	94	10.85%
Mineralt	90	10.39%
JSECoin	50	5.77%
CryptoLoot	39	4.50%
CryptoNoter	31	3.58%
Coinhave	14	1.62%
Minr	13	1.50%
Webmine	8	0.92%
DeepMiner	5	0.58%
Cpufun	4	0.46%
Monerise	2	0.23%
NF WebMiner	2	0.23%
Total	866	100%

## Effectiveness of fingerprint-based detection

Mining Service	Number of Websites	Percentage
Coinhive	514	59.35%
CoinImp	94	10.85%
Mineralt	90	10.39%
JSECoin	50	5.77%
CryptoLoot	39	4.50%
CryptoNoter	31	3.58%
Coinhave	14	1.62%
Minr	13	1.50%
Webmine	8	0.92%
DeepMiner	5	0.58%
Cpufun	4	0.46%
Monerise	2	0.23%
NF WebMiner	2	0.23%
Total	866	100%

- ▶ Detected 866 websites; 59.35% used Coinhive cryptomining services

## Effectiveness of fingerprint-based detection

Mining Service	Number of Websites	Percentage
Coinhive	514	59.35%
CoinImp	94	10.85%
Mineralt	90	10.39%
JSECoin	50	5.77%
CryptoLoot	39	4.50%
CryptoNoter	31	3.58%
Coinhave	14	1.62%
Minr	13	1.50%
Webmine	8	0.92%
DeepMiner	5	0.58%
Cpufun	4	0.46%
Monerise	2	0.23%
NF WebMiner	2	0.23%
Total	866	100%

- ▶ Detected 866 websites; 59.35% used Coinhive cryptomining services
- ▶ Issues: code obfuscation and manual effort of updating signatures

## Preliminary results: Mining pool communication (1/2)

- ▶ Miners use the Stratum protocol to communicate with the mining pool
- ▶ Use of WebSockets to allow full-duplex, asynchronous communication between code running on a webpage and servers
- ▶ Search in WebSocket frames for keywords related to Stratum protocol

Command	Keywords
Authentication	type:auth   command:connect   identifier:handshake   command:info
Authentication accepted	type:authed   command:work
Fetch job	identifier:job   type:job   command:work   command:get_job   command:set_job
Submit solved hash	type:submit   command:share
Solution accepted	command:accepted
Set CPU limits	command:set_cpu_load



## Preliminary results: Mining pool communication (2/2)

- ▶ 59,319 (5.39%) websites use WebSockets
- ▶ 1,008 websites use Stratum protocol for communication
- ▶ 2,377 websites encode the data (Hex code or salted Base64)
  - more on this later

## Summary of key findings

- ▶ Identified 1,735 websites as mining cryptocurrency, out of which 1,627 (93.78%) could be identified based on keywords in the cryptomining code
- ▶ 1,008 (58.10%) use the Stratum protocol in plaintext, 174 (10.03%) obfuscate the communication protocol
- ▶ All the websites (100.00%) use Wasm for the cryptomining payload and open a WebSocket
- ▶ At least 197 (11.36%) websites throttle their CPU usage to less than 50%, while for only 12 (0.69%) mining websites we observed a CPU load of less than 25%.

## In-depth analysis: evasion techniques (1/2)

We identified three evasion techniques, which are widely used by the drive-by mining services in our dataset

- ▶ Code obfuscation
  - ▶ *Packed code*: The compressed and encoded orchestrator script is decoded using a chain of decoding functions at run time.
  - ▶ *PCharCode*: The orchestrator script is converted to charCode and embedded in the webpage. At run time, it is converted back to a string and executed using JavaScript's eval() function.
  - ▶ *Name obfuscation*: Variable names and functions names are replaced with random strings.
  - ▶ *Dead code injection*: Random blocks of code, which are never executed, are added to the script to make reverse engineering more difficult.
  - ▶ *Filename and URL randomization*: The name of the JavaScript file is randomized or the URL it is loaded from is shortened to avoid detection based on pattern matching.

## In-depth analysis: evasion techniques (1/2)

We identified three evasion techniques, which are widely used by the drive-by mining services in our dataset

- ▶ Code obfuscation
  - ▶ *Packed code*: The compressed and encoded orchestrator script is decoded using a chain of decoding functions at run time.
  - ▶ *PCharCode*: The orchestrator script is converted to charCode and embedded in the webpage. At run time, it is converted back to a string and executed using JavaScript's eval() function.
  - ▶ *Name obfuscation*: Variable names and functions names are replaced with random strings.
  - ▶ *Dead code injection*: Random blocks of code, which are never executed, are added to the script to make reverse engineering more difficult.
  - ▶ *Filename and URL randomization*: The name of the JavaScript file is randomized or the URL it is loaded from is shortened to avoid detection based on pattern matching.
- ▶ Mainly applied to orchestrator code, only obfuscation on mining payload is *name obfuscation*

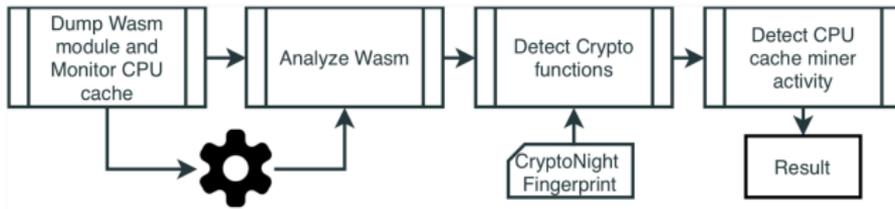
## In-depth analysis: evasion techniques (2/2)

- ▶ Identified the Stratum protocol in plaintext for 1,008 websites
- ▶ Manually analyzed the WebSocket communication for the remaining 727 websites and found the following:
- ▶ Obfuscate by encoding the request, either as Hex code, or with salted Base64 encoding before transmitting it through the WebSocket
- ▶ Could not identify any pool communication for the remaining 553 websites, either due to other encodings, or due to slow server connections

Finally, anti-debugging tricks (139 websites): code periodically checks whether the user is analyzing the code served by the webpage using developer tools. If the developer tools are open in the browser, it stops executing any further code

# MineSweeper

- ▶ MineSweeper employs multiples stages in order to detect a webminer:



## CryptoNight algorithm (1/2)

- ▶ CryptoNight was proposed in 2013 and popularly used by Monero (XMR)

## CryptoNight algorithm (1/2)

- ▶ CryptoNight was proposed in 2013 and popularly used by Monero (XMR)
- ▶ We exploit two fundamental characteristics:

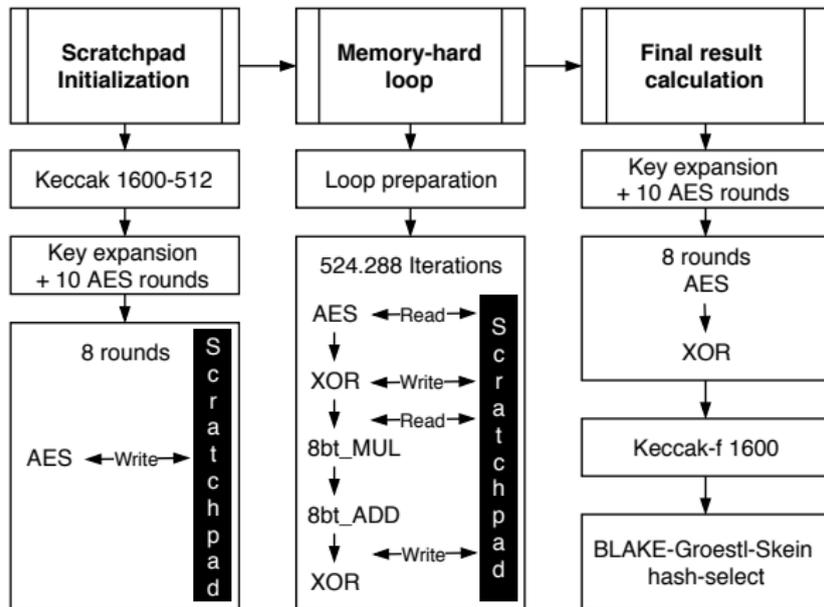
## CryptoNight algorithm (1/2)

- ▶ CryptoNight was proposed in 2013 and popularly used by Monero (XMR)
- ▶ We exploit two fundamental characteristics:
- ▶ It makes use of several cryptographic primitives
  - Keccak 1600-516, Keccak-f 1600, AES, BLAKE-256, Groestl-256, and Skein-256

## CryptoNight algorithm (1/2)

- ▶ CryptoNight was proposed in 2013 and popularly used by Monero (XMR)
- ▶ We exploit two fundamental characteristics:
- ▶ It makes use of several cryptographic primitives
  - Keccak 1600-516, Keccak-f 1600, AES, BLAKE-256, Groestl-256, and Skein-256
- ▶ A memory hard algorithm
  - ▶ High-performances on ordinary CPUs
  - ▶ Inefficient on today's special purpose devices (ASICs)
  - ▶ Internal memory-hard loop: alternate reads and writes to the Last Level Cache (LLC)

## CryptoNight algorithm (2/2)



- ▶ CryptoNight allocates a scratchpad of 2MB in memory
- ▶ On modern processors ends up in the LLC

## Wasm analysis

- ▶ Linear assembly bytecode translation using the WebAssembly Binary Toolkit (WABT) debugger
- ▶ Functions identification - to create an internal representation of the code for each function
- ▶ Cryptographic operation count - track the control flow and crypto operands
- ▶ Static call graph construction, including identification of loops

## CryptoNight detection

- ▶ MineSweeper is given as input a CryptoNight fingerprint
- ▶ We created a fingerprint for each of CryptoNight's cryptographic primitives based on operands counts and flow structure
- ▶ If 3 out of the 5 cryptographic primitives are good matches, then the miner is identified

## CryptoNight detection - example

- ▶ Assume the fingerprint for BLAKE-256 has 80 XOR, 85 left shift, and 32 right shift instructions
- ▶ Function `foo()`, which is an implementation of BLAKE-256, that we want to match against this fingerprint, contains 86 XOR, 85 left shift, and 33 right shift instructions
- ▶ In this case, the similarity score is 3 and difference score is 2
- ▶ all three types of instructions are present in `foo()`; `foo()` contains extra XOR and an extra shift instruction

## CPU cache events monitoring

- ▶ What if an attack would sacrifice part of the profits for obfuscated Wasm?
- ▶ Solution: CPU cache events monitoring
- ▶ MineSweeper monitors the L1 and L3 for load and store events caused by the CryptoNight algorithm
- ▶ Also detects a fundamental characteristic of the CryptoNight algorithm: the memory-hard loop!

## Evaluation of blacklisting approaches

- ▶ For comparison, we evaluate MineSweeper against Dr. Mine
- ▶ Dr. Mine uses CoinBlockerLists as the basis to detect mining websites
- ▶ Visited the 1,735 websites that were mining during our first crawl for the large-scale analysis with both tools
- ▶ Dr. Mine could only find 272 websites, while MineSweeper found 785 websites that were still actively mining cryptocurrency

# Evaluation of cryptofunction detection

- ▶ Identified 38 unique samples among the 748 collected Wasm samples
- ▶ Applied the cryptofunction detection routine of MineSweeper on them

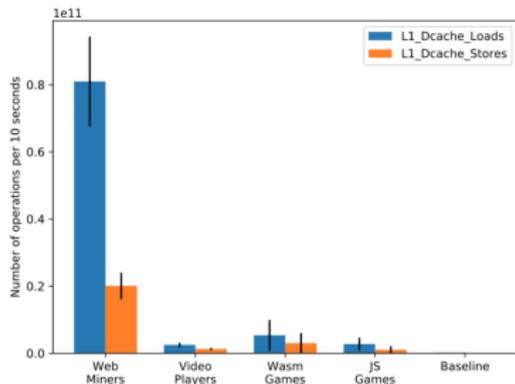
<b>Detected Primitives</b>	<b>Number of Wasm Samples</b>	<b>Number of Cryptominers</b>	<b>Missing Primitives</b>
5	30	30	-
4	3	3	AES
3	-	-	-
2	3	3	Skein, Keccak, AES
1	-	-	-
0	4	0	All

## Evaluation of CPU cache events monitoring (1/2)

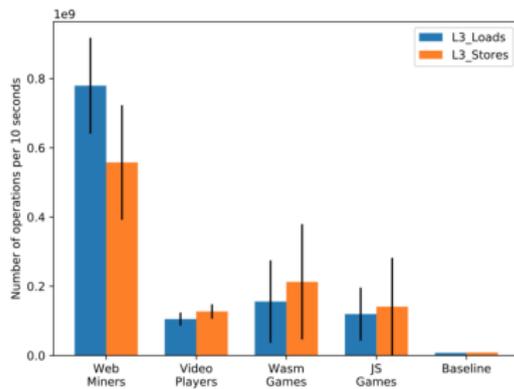
- ▶ We visited 7 pages for the following categories of applications:
  - ▶ Cryptominers
  - ▶ Videoplayers
  - ▶ Wasm-based games
  - ▶ JavaScript (JS) games

## Evaluation of CPU cache events monitoring (2/2)

Our tests confirm us the effectiveness of this detection method on CryptoNight-based algorithms



Performance counter statistics for the L1 cache for different types of web applications (logscale)



Performance counter statistics for the L3 cache for different types of web applications (logscale)

# Conclusion

Crawling period	March 12, 2018 – March 19, 2018
# of crawled websites	991,513
# of drive-by mining websites	1,735 (0.18%)
# of drive-by mining services	28
# of drive-by mining campaigns	20
# of websites in biggest campaign	139
Estimated overall profit	US\$ 188,878.84
Most profitable/biggest campaign	US\$ 31,060.80
Most profitable website	US\$ 17,166.97

- ▶ Drive-by mining is real and can be very profitable for high traffic websites
- ▶ Current defenses are not sufficient to stop malicious mining
- ▶ To severely impact their profitability, we need to aim at the core properties of the miners code: **cryptographic functions** and **memory behaviors**

Thank you for your attention!

[email@veelasha.org](mailto:email@veelasha.org)

[www.veelasha.org](http://www.veelasha.org)