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

RAJSHAKHAR PAUL

WAYNE STATE
Outlines

- Introduction
- Motivation
- Background
- Threat Model
- Data Analysis
- Conclusion
Intro

- Cryptocurrency
  - A digital asset
  - Work as a medium of exchange that uses strong cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets
  - Use blockchain method
  - Example: Bitcoin

- Drive-by mining
  - A web-based attack, in which an infected website secretly executes JavaScript code and/or a WebAssembly module in user's browser to mine cryptocurrencies without the consent of the user
  - Also known as Cryptojacking
Cryptocurrency Mining

- A process in which transactions for various forms of cryptocurrency are verified and added to the blockchain digital ledger

- Each time a cryptocurrency transaction is made, a cryptocurrency miner is responsible for ensuring the authenticity of information and updating the blockchain with the transaction

- The mining process need a lot of computational power

- The first cryptocurrency miner to crack the code is rewarded by being able to authorize the transaction
Cryptojacking Attack

- Using cryptomining service like Coinhive
- By compromising web servers
- Taking advantage of misconfiguration and installing JavaScript based miners
- Distributing miners through online advertisements
- Compromising third party libraries
- By using pop-under window
- By playing online video streaming
Outlines

- Introduction
- Motivation
- Background
- Threat Model
- Data Analysis
- Conclusion
Motivation

- Dedicated browser extensions and ad blockers use blacklists.
- Maintaining a complete blacklist is not scalable and prone to false negative.
- Easily defeated by URL randomization and domain general algorithms

- Some detection technique look for high CPU usage as an indicator of cryptocurrency mining.
- Causes both false positive and false negative as one program may take high CPU usage and on the other hand cryptocurrency miners have started to throttle their CPU usage to evade detection.
Proposed Method

- Focus on Wasm-based mining, the most efficient and widespread technique for drive-by mining attacks.
- Identify the intrinsic characteristics of the mining itself: the hashing function.
- Two level approaches.
  - First, perform static analysis on the Wasm code and identify the hashing code based on cryptographic operations it performs.
  - Second, monitor CPU cache events at run time to identify cryptominers based on their memory access patterns.
Contributions

- Perform the first in-depth assessment of drive-by mining
- Discuss why current defense based on blacklisting and CPU usage are ineffective
- Propose a novel detection approach based on the identification of cryptographic functions through static analysis and monitoring of cache events during run time
Outlines

- Introduction
- Motivation
- Background
- Threat Model
- Data Analysis
- Conclusion
Existing Defenses against Drive-by Mining

- **CoinBlockerList**
  - Maintain a blacklist of mining pools and proxy servers that is manually collected from reports on security blogs and twitter

- **Dr. Mine**
  - Attempts to block drive-by mining by means of explicitly blacklisted URLs

Both approaches suffer from high false negative
Outlines

- Introduction
- Motivation
- Background
- Threat Model
- Data Analysis
- Conclusion
Threat Model

1. User sends an HTTP request to the webserver.
2. Webserver returns the HTTP response containing orchestrator code.
3. User processes the response and fetches the mining payload.
4. User communicates with the mining pool via WebSocket proxy.
5. Mining pool sends a relay communication to the mining pool.
When a user visits a drive-by mining website, the website (1) serves the orchestrator script, which checks the host environment to find out how many CPU cores are available.
(2) downloads the highly-optimized cryptomining payload (as either Wasm or asm.js) from the website or an external server.
(3) instantiates a number of web workers i.e., spawns separate threads, with the mining payload, depending on how many CPU cores are available.
(4) sets up the connection with the mining pool server through a WebSocket proxy server.
(5) Finally, fetches work from the mining pool and submits the hashes to the mining pool through the WebSocket proxy server.
Outlines

- Introduction
- Motivation
- Background
- Threat Model
- Data Analysis
- Conclusion
Data Collection

- Build web crawler for visiting Alexa’s top 1 million websites
- The crawler stays for four seconds on each visited page
- Collect data related to drive-by mining
- Identify Orchestrator and Mining Payload with the help of key word based search and Wasm module respectively
Data Analysis

Three different artifacts produced by the data collection system

1. Cryptomining Code
   - Authors identified 13 well known cryptomining services using keywords listed below.
   - 866 websites are using these 13 services without obfuscating the orchestrator

<table>
<thead>
<tr>
<th>Mining Service</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coinhive</td>
<td>new Coinhive\Anonymous</td>
</tr>
<tr>
<td>CryptoNote</td>
<td>minercry.pt/processor.js</td>
</tr>
<tr>
<td>NFWebMiner</td>
<td>new NFMiner</td>
</tr>
<tr>
<td>JSECoin</td>
<td>load.jsseoin.pro/load</td>
</tr>
<tr>
<td>Webmine</td>
<td>webmine.cz/miner</td>
</tr>
<tr>
<td>CryptoLoot</td>
<td>CRLT\Anonymous</td>
</tr>
<tr>
<td>DeepMiner</td>
<td>new deepMiner.Anonymous</td>
</tr>
<tr>
<td>Monerise</td>
<td>apin.monerise.com</td>
</tr>
<tr>
<td>Coinhave</td>
<td>minescripts\info</td>
</tr>
<tr>
<td>Cpufun</td>
<td>snipli.com/[A-Za-z]+&quot; data-id=</td>
</tr>
<tr>
<td>Minr</td>
<td>abc\pema\cl</td>
</tr>
<tr>
<td>Mineralt</td>
<td>ecart\html?bdata=</td>
</tr>
</tbody>
</table>
Data Analysis

Three different artifacts produced by the data collection system

2. CPU Load as a Side Effect
   - Not used to detect drive-by mining
Data Analysis

Three different artifacts produced by the data collection system

3. Mining Pool Communication

- Authors identified 1,008 websites that are communicating with mining pool servers using the Stratum protocol based on the keywords

<table>
<thead>
<tr>
<th>Command</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>type:auth</td>
</tr>
<tr>
<td>Authentication accepted</td>
<td>type:authed</td>
</tr>
<tr>
<td>Fetch job</td>
<td>identifier:job</td>
</tr>
<tr>
<td>Submit solved hash</td>
<td>type:submit</td>
</tr>
<tr>
<td>Solution accepted</td>
<td>command:accepted</td>
</tr>
<tr>
<td>Set CPU limits</td>
<td>command:set_cpu_load</td>
</tr>
</tbody>
</table>
Data Correlation

Using Cryptomining Code and Mining Pool Communication authors identify:

- There are 402 websites that don’t need user consent.
- Other 464 websites wait for user’s consent.
Results

Three evasion techniques have been identified:
- Code Obfuscation
- Obfuscated Stratum Communication
- Anti-debugging Tricks
Modus Operandi of Attack

Three ways

1. Miners inject through third-party services
2. Miners inject through advertisement networks
3. Miners inject by compromising vulnerable websites
Common Drive-by Mining Characteristics

Three characteristics:

1. All services use CryptoNight-based cryptomining implementations.
2. All identified websites use a highly-optimized Wasm implementation of the CryptoNight algorithm to execute the mining code.
3. All drive-by mining websites use WebSockets to communicate with the mining pool through a WebSocket proxy server.
How MineSweeper Works?

- Takes the URL of website as input
- Detection Based on Primitive Identification
- Generic Cryptographic Function Detection
- Detection Based on CPU Cache Events
Deployment

MineSweeper Can be integrated into browsers easily.

- Will overcome the limitations of Blacklists based system.
Limitations

- It only spends 4 seconds on each webpage. Could miss websites that wait for more time.
- Not capable to capture the mining pool communication for websites that implement mining delays.
Thank You