Abstract

The high profitability of mining cryptocurrencies mining, a computationally intensive activity, forms a fertile ecosystem that is enticing not only legitimate investors but also cyber attackers who invest their illicit computational resources in this area. Cryptojacking refers to the surreptitious exploitation of a victim's computing resources to mine cryptocurrencies on behalf of the cyber-criminal. This malicious behavior is observed in executable files and browser executable codes, including JavaScript and Assembly modules, downloaded from websites to victims' machines and executed. Although there are numerous botnet detection techniques to stop this malicious activity, attackers can circumvent these protections using a variety of techniques. In this paper, CryptojackingTrap is presented as a novel cryptojacking detection solution designed to resist most malware defense methods. The CryptojackingTrap is armed with a debugger and extensible cryptocurrency listeners and its algorithm is based on the execution of cryptocurrency hash functions: an indispensable behavior of all cryptojacking executors. This algorithm becomes aware of this specific hash execution by correlating the memory access traces of suspicious executables with publicly available cryptocurrency P2P network data. With the advantage of this assembly-level investigation and a nature-inspired approach to triggering the detection alarm, CryptojackingTrap provides an accurate, evasion-proof technique for detecting cryptojacking. After experimental evaluation, the false negative and false positive rates are zero, and in addition, the false positive rate is mathematically calculated as $10^{-30}$. CryptojackingTrap has an open, extensible architecture and is available to the open-source community.

Background

- **Main chain**: the longest chain (black blocks)
- **Genesis block**: the first block of a blockchain (the green block)
- **Orphan block**: a block that is not in the main chain (gray blocks)
- **Current block**: the last block in the main chain
- **Previous block hash**: a hash pointer to provide a tamper-proof structure in blockchain

Figure 1: Blockchain data structures with hash pointers

Table 1: Summary of Notation for CryptojackingTrap[1].

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MO</td>
<td>Mining Occurrence</td>
</tr>
<tr>
<td>DO</td>
<td>Detection Occurrence</td>
</tr>
<tr>
<td>IO</td>
<td>infection Occurrence</td>
</tr>
<tr>
<td>LO</td>
<td>loss Occurrence</td>
</tr>
<tr>
<td>CO</td>
<td>coverage Occurrence</td>
</tr>
</tbody>
</table>

Algorithm 1 Detector Algorithm: Is CryptojackingTrap[1].

```python
def detect_cryptojacking(executable):
    if detect_hash_function_access(executable):
        return True
    else:
        return False
```

Figure 2: Monetize the stolen cycles for Bot masters

The current mechanisms for detecting cryptojacking malware can be evade in several ways:

1. Dynamic analysis of C&C network traffic: This can be bypassed via encrypted channels.
2. Static analysis of malware code and cryptographic constants: Vulnerable to code obfuscation.
3. CPU activity analysis: Evasion possible through mining rate reduction.
4. Static code analysis for mining pools keywords and URLs: Ineffective due to code obfuscation and dynamic crypto ecosystem.

Figure 3: Venus Flytrap optimized mechanism to catch flies

Table 2: randomly generated (upper) and benign non-miner applications (lower) test results[1].

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Is CryptojackingTrap</th>
<th>Is a Valid Miners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Lower</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 4: Illustrative examples demonstrating levels of CryptojackingTrap detection abstraction[1].

Figure 5: Venus Flytrap optimized mechanism to catch flies

Figure 6: Asynchronous architecture of CryptojackingTrap[1]

Figure 7: Github code

Conclusion

- A cutting-edge method for identifying cryptocurrency mining activity within suspicious applications
- Encompasses executable files, processes, and websites
- Focusing on the success phase of malware and predicting miners' low-level memory access
- By leveraging cryptocurrency network data, this technique detects mining activities without relying on traditional detection methods like function signatures or code features
- Highly resistant to obfuscation, effective across diverse botnet network protocols, resilient to ten times mining rate reduction
- Supporting Bitcoin, Ethereum, and Monero mining detection
- Modular design facilitates easy extension to include additional cryptocurrencies
- Mathematical and experimental evaluation results demonstrate its high accuracy and exceptionally low false positive rate
- An open-source release comprising 6.5K lines of code (SLOC)
- CryptojackingTrap stands as a versatile and robust solution for modern cybersecurity challenges.

References