This article has been accepted for publication in IEEE Transactions on Information Forensics and Security.

CryptojackingTrap: An Evasion Resilient Nature-Inspired Algorithm to Detect Cryptojacking Malware

UNIVERSITÀ DI TRENTO

Amirkabir University of Technology (Tehran Polytechnique)

Atefeh Zareh Chahoki[1], Hamid Reza Shahriari[2] and Marco Roveri[1]
[1] University of Trento, [2]Amirkabir University of Technology, Tehran, Iran

ICT DAYS
16-18 April 2024

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^{-20}. CryptojackingTrap has an open, extensible architecture and is available to the open-source community.

Background

- **Main chain**: the longest chain (black blocks)
- **Genesis block** (1): the first block of a blockchain (the green block)
- **Orphan block** (2): a block that is not in the main chain (gray blocks)
- **Current Block** (3): the last block in the main chain
- **Previous block hash** (4): a hash pointer to provide a tamper-proof structure in blockchain

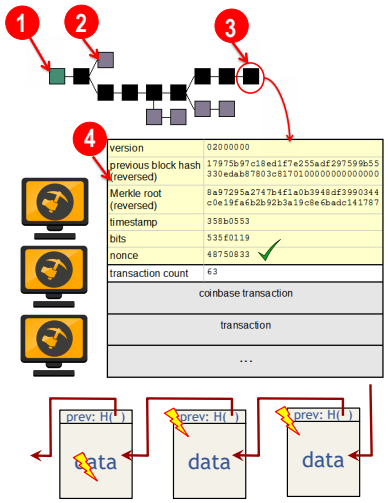

Figure 1: Blockchain data structures with hash pointers

Problem Statement

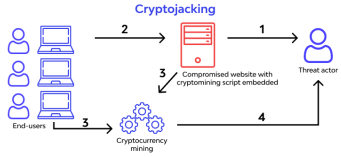

Figure 2: Monetize the stolen cycles for Bot masters

The current mechanisms for detecting Cryptojacking malware can be evaded 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**.

CryptojackingTrap

How CryptojackingTrap coupe with these evasion approaches:

- focuses on the intrinsic activity of the success phase of malware
- traces the low-level memory access of malware
- predicts the data that miners must access from the cryptocurrency network

Inspired by the Venus Flytrap :

- Used an algorithm to detect flies on its lobes
- Minimized false positives
- Implemented a mechanism using 3 trigger hairs in each lobe and a 21-second timer window after the initial hit to monitor and confirm the final detection.

Figure 3: Venus Flytrap optimized mechanism to catch flies

The Levels of CryptojackingTrap Detection Abstractions:

- Split Occurrence (SO).
- Hash Occurrence (HO).
- Mining Occurrence (MO).
- CryptojackingTrap Occurrence (CO).

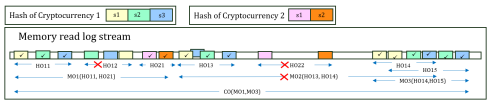

Figure 4: Illustrative examples demonstrating levels of CryptojackingTrap detection abstraction [1].

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

Level	Symbol	Meaning
SO	Ω_c	Predictable bytes count in each cryptocurrency c
	Ξ	Min size of each acceptable SO's substring
HO	Π	Min percentage of SO's hash coverage that specifies one HO
	Δ	Max window of all SOs to specify one HO
MO	Ψ	Min number of HOs to specify one MO
	Θ	Max window of all HOs to specify one MO
CO	Γ	Min number of MOs to specify the CryptojackingTrap alert
	Λ	Max window of all MOs to specify the CryptojackingTrap alert

Algorithm 1 Detector Algorithm: Is Cryptojacking [1].

```
1:  procedure isCryptojacking(setting: DetectorSetting): Boolean
    ▷ returns true if finds Γ MOs in a Λ-sized window
2:      mos ← findMiningOccurrences(setting)
3:      for i = 0; i ⩽ mos.size-1; i++ do
4:          mo_start ← mos.get(i)
5:          acceptedMOs ← new MiningOccurrence[]
6:          acceptedMOs.add(mo_start)
7:          for j = i+1; j ⩽ mos.size-1; j++ do
8:              mo_end ← mos.get(j)
9:              if mo_end.rlEndIdx-mo_start.rlStartIdx ≤ Λ then
10:                 acceptedMOs.add(mo_end)
11:                 if acceptedMOs.size ≥ Γ then
12:                     print "Positive Alarm!" + acceptedMOs
13:                     return true
14:                 else continue ▷ add the next MO to acceptedMOs
15:             else break ▷ restart accepted MO window
16:     print "Negative Alarm!"
17:     return false
```

Architecture

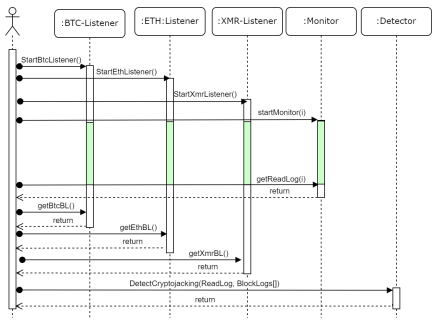

Figure 5: Venus Flytrap optimized mechanism to catch flies [1].

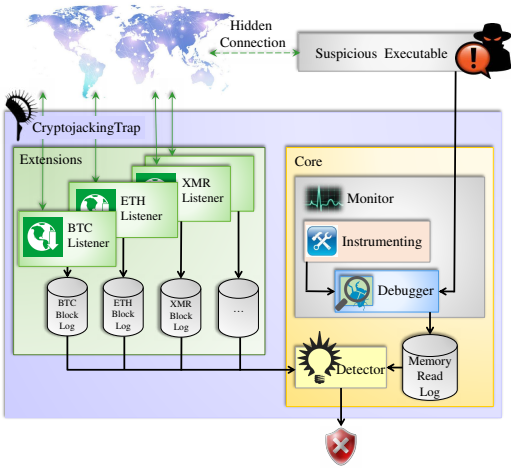

Figure 6: Asynchronous architecture of CryptojackingTrap.[1]

Evaluation

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

	Name	Time	Valid Split Occurrences Information				Discarded Split Occurrences Error Types		
			Coverage	Splits Count	Max Split Size	Avg Split Size	MISSED	TOO_SMALL	TOO_FAR
Random Miners	100% Random	35.55 s	27.91 %	17.33	5.11	3.63	0.91 %	34.54 %	37.47 %
	85% Random	36.34 s	60 %	13	18	6.0	0 %	30.77 %	23.07 %
	50% Random	49.57 s	68.5 %	8	18	12.5	0 %	37.5 %	18.75 %
	15% Random	47.92 s	75 %	8	18	12.0	25 %	12.5 %	12.5 %
Benign Non-Miners	Adobe Acrobat DC	37.03 s	21.60 %	32.33	4.15	3.00	8.76 %	62.37 %	16.13 %
	Microsoft Calculator	31.54 s	21.93	28.50	3.78	3.00	7.02 %	57.31 %	20.37 %
	Microsoft Word	30.87 s	25.17	30.17	4.00	3.00	8.29 %	57.46 %	17.68 %
	Notepad++	24.76 s	19.76	27.83	3.30	3.00	11.38 %	53.29 %	20.26 %
	Photos	26.02 s	20.12	30.00	3.79	3.10	12.22 %	57.22 %	17.04 %

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

[1] A. Z. Chahoki, H. R. Shahriari, and M. Roveri, "CryptojackingTrap: An Evasion Resilient Nature-Inspired Algorithm to Detect Cryptojacking Malware," in IEEE Transactions on Information Forensics and Security, doi: 10.1109/TIFS.2024.3353072.