Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.3, Jun. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.3.471

브라우저 익스텐션 기반 암호화폐 지갑의 디지털 포렌식 아티팩트 수집 및 분석 연구*

김 주 은,[1†] 서 승 희,[1] 석 병 진,[2] 변 현 수,[1] 이 창 훈[3‡]
[1,3]서울과학기술대학교 (대학원생, 교수), [2]서울과학기술대학교 전기정보기술연구소 (연구원)

A Study on the Digital Forensics Artifacts Collection and Analysis of Browser Extension-Based Crypto Wallet*

Ju-eun Kim,[1†] Seung-hee Seo,[1] Beong-jin Seok,[2] Heoyn-su Byun,[1] Chang-hoon Lee[3‡]
[1,3]Seoul National University of Science and Technology (Graduate student, Professor),
[2]Seoul National University of Science and Technology Research Center of Electrical and Information Technology (Researcher)

요 약

최근 사용자의 익명성이 보장되는 블록체인의 특성으로 인해 블록체인 기반 기술인 암호화폐가 불법 거래 등의 범죄에 악용되는 사례가 증가하고 있다. 하지만 암호화폐는 암호화폐 지갑에서 보호되어 범죄 자금 환수에 어려움이 있는 실정이다. 따라서 본 연구는 범죄에 사용된 암호화폐를 추적·환수하기 위해 브라우저 익스텐션 월렛 4종(Metamask, Binance, Phantom, Kaikas)을 대상으로 사용자 행위에 기반하여 로컬 PC의 데이터와 메모리 영역에서 아티팩트를 획득하고, 디지털 포렌식 관점에서의 활용 방안을 분석한다. 분석 결과로 브라우저의 캐시 데이터에서 획득한 API명을 통해 피의자가 사용한 지갑과 암호화폐의 종류를 확인했으며 송금 거래에 사용된 URL과 지갑 주소를 획득했다. 또한 쿠키 데이터에서 사용된 디바이스를 식별할 수 있는 Client ID를 확인하고, 메모리에서 니모닉 코드를 획득 가능함을 확인했다. 추가적으로, 획득가능한 니모닉 코드의 지속성을 측정하고 획득을 자동화하기 위한 알고리즘을 제안한다.

ABSTRACT

Recently, due to the nature of blockchain that guarantees users' anonymity, more and more cases are being exploited for crimes such as illegal transactions. However, cryptocurrency is protected in cryptocurrency wallets, making it difficult to recover criminal funds. Therefore, this study acquires artifacts from the data and memory area of a local PC based on user behavior from four browser extension wallets (Metamask, Binance, Phantom, and Kaikas) to track and retrieve cryptocurrencies used in crime, and analyzes how to use them from a digital forensics perspective. As a result of the analysis, the type of wallet and cryptocurrency used by the suspect was confirmed through the API name obtained from the browser's cache data, and the URL and wallet address used for the remittance transaction were obtained. We also identified Client IDs that could identify devices used in cookie data, and confirmed that mnemonic code could be obtained from memory. Additionally, we propose an algorithm to measure the persistence of obtainable mnemonic code and automate acquisition.

Keywords: Metamask, Binance, Phantom, Kaikas, Digital Forensics

Received(03. 13. 2023), Modified(05. 09. 2023), Accepted(05. 09. 2023)

* 본 논문은 2022년도 한국정보보호학회 동계학술대회에 발표한 우수논문을 개선 및 확장한 것임.

* 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임. (No.2021-0-00540, GPU/ASIC 기반 암호알고리즘 고속화 설계 및 구현 기술개발)

† 주저자, jek0104@seoultech.ac.kr

‡ 교신저자, chlee@seoultech.ac.kr(Corresponding author)

I. 서 론

최근 블록체인 기술이 가치를 인정받고 다양한 분야에 적용됨에 따라 암호화폐 시장(Cryptocurrency Market) 역시 성장해 암호화폐를 통한 거래가 활발하게 수행되고 있다. 암호화폐 시장에서의 거래는 P2P 방식으로 구성되는 블록체인 네트워크의 특성상 신분 인증 과정 없이 수행되어 거래자의 익명성이 보장된다는 특성이 있다. 이와 같은 거래의 익명성은 거래자들이 프라이버시 관점에서 더욱 안전한 거래를 수행할 수 있도록 하지만, 최근 거래의 익명성을 악용해 불법 거래, 자금 세탁을 목적으로 암호화폐를 활용하는 범죄가 증가하고 있어 이에 대한 대응 방안이 필요하다. 실사례로 2021년 미국의 콜로니얼 파이프라인 랜섬웨어 공격 사건에서는 랜섬웨어 복호화를 위한 몸값을 암호화폐로 요구하여 범죄수익을 창출했다. 사건 당시, FBI가 공격자의 비트코인 지갑의 권한을 가질 수 있는 개인 키를 확보하여 불법으로 입수한 몸값의 일부를 회수할 수 있었다[1]. 또한, 2019년 국내에서 발생한 N번방 성 착취물 제작 및 유포 사건에서도 범죄 수익 지불 수단으로 암호화폐가 사용되었다[2]. 상기 언급한 사례와 같은 암호화폐의 범죄 악용에 대응하기 위해서 암호화폐의 거래 내역 추적을 위해 범죄에 사용된 지갑 주소를 반드시 입수해야 한다. 지갑 주소 정보가 있다면 블록체인 원장에서 투명하게 관리되고 있는 암호화폐 거래내역을 추적 가능하다. 하지만 불법적으로 악용된 지갑 주소는 암호화폐 자산의 보호를 위해 개발된 암호화폐 지갑의 사용으로 인해 수집에 어려움이 있으며 이에 암호화폐를 통한 범죄수익 환수는 더욱 어려운 실정이다.

암호화폐 지갑은 탈중앙화된 블록체인 네트워크에서 암호화폐의 거래 및 관리를 위해 사용된다. 암호화폐 지갑과 거래의 보호를 위해 사용자별로 공개 키와 개인 키, 지갑 주소 및 니모닉 코드를 사용한다. 일반적인 은행 앱에서 공인인증서를 사용해 통장을 만드는 것과 유사하게 암호화폐 지갑에니모닉 코드로부터 개인 키와 공개 키, 지갑 주소를 생성하여 거래를 수행할 수 있도록 한다. 이에 따라, 개인 키 및 니모닉 코드가 은행 거래의 공인인증서와 같이 자산의 소유권 증명 또는 지갑 복구 수단으로 사용되는 것을 알 수 있다. 지갑 주소와 니모닉 코드 정보는 암호화폐 지갑의 불법자금 환수를 위한 핵심 정보로, 자금 환수를 목적으로 디지털 포렌식 수사 시 원활한 조사를 위해 암호화폐 지갑 서비스별 아티팩트 수집 및 분석 연구가 선행되어야 한다. 이에 본 논문에서는 브라우저 익스텐션 월렛(Browser Extension Wallet) 4종(Metamask, Binance, Phantom, Kaikas)을 대상으로 암호화폐 지갑에 대한 디지털 포렌식 아티팩트 획득 및 분석 연구를 수행한다. 브라우저 익스텐션 월렛은 인터넷 브라우저 상에서 확장 프로그램으로 동작하는 암호화폐 지갑으로, 설치 과정이 간단하고 계정 생성이 쉬워 많은 사용자를 보유하고 있다. 데스크톱 앱 형태로 동작하는 지갑 또한 브라우저 익스텐션 버전의 월렛 서비스를 추가로 도입하고 있는 추세로, 암호화폐 수사에 있어 본 연구의 결과가 매우 유의미하게 활용될 것으로 기대된다.

연구 결과 암호화폐 지갑이 동작하는 구글 크롬 브라우저의 캐시 및 쿠키 데이터에서 지갑의 세부 정보와 지갑 주소를 수집하였고, 메모리 상에서 니모닉 코드를 수집할 수 있었다.

2장은 기존 암호화폐 지갑 서비스의 아티팩트 수집 및 분석 관점의 관련 연구를 제시한다. 3장은 연구 환경 및 방법에 대해 설명하고, 4장에서 크롬 브라우저의 캐시·쿠키 데이터에서 획득할 수 있는 암호화폐 관련 아티팩트 수집 및 분석 결과를 기술한다. 그리고 5장에서 메모리로부터 니모닉 코드 아티팩트를 추출한 결과를 기술한 후 용량이 큰 메모리 덤프 파일로부터 니모닉 코드 검색을 위한 알고리즘을 제안한다. 6장에서는 실험에 따른 암호화폐 지갑별 아티팩트의 지속성을 그래프로 나타낸다. 7장에서 디지털 포렌식 관점에서의 아티팩트 활용 방안을 설명한다. 마지막으로 8장에서 결론 및 한계점을 도출한다.

II. 관련 연구

개인 암호화폐 지갑은 크게 하드웨어 형식의 지갑과 소프트웨어 형식의 지갑으로 나뉜다. 콜드 월렛이라고도 불리는 하드웨어 지갑은 네트워크 연결을 차단하여 오프라인으로 암호화폐 자금을 보관할 수 있는 지갑이다. 소프트웨어 지갑에 비해 상대적으로 안전하게 거래를 할 수 있다는 특징을 갖는다. 하드웨어 지갑을 대상으로 연구를 수행한 Tomas 외 3명은 Ledger Live, Trezor에 대해 메모리 영역에서 유의미한 아티팩트를 수집 및 분석하고 그에 대한 지속성과 무결성을 측정하여 가시화했다. 또한 대상 암호화폐 지갑에 대한 디지털 포렌식 관점의 프레임워크를 제안하였다[3]. 메모리 데이터의 잔존 여부를

시나리오에 따른 사용자 행위 기준의 그래프로 나타냈고, 메모리에서 아티팩트가 저장된 물리적 위치 또한 그림으로 정리했다. 해당 연구의 시간 경과에 따른 메모리 잔존 여부 및 지속성 가시화 아이디어는 본 연구에서도 참고해 활용했다. Lim 외 5명은 하드웨어 지갑 Ledger, Trezor에 대해 디지털 포렌식 관점에서의 아티팩트 수집 연구를 수행했다. 로컬 레지스트리를 통해 지갑 연결 기록을 획득했으며, 메모리에서 비트코인, 리플, 이더리움 지갑의 주소와 니모닉 코드를 정규표현식을 통해 추출할 수 있음을 보였다[4].

디지털 포렌식 관점의 소프트웨어 암호화폐 지갑 아티팩트 분석 연구 또한 활발히 수행되고 있다. Zollner외 3명은 비트코인 거래에 주로 사용되는 데스크탑 애플리케이션, 모바일 앱, 브라우저 익스텐션, 하드웨어 월렛에서 서비스되는 암호화폐를 각각 하나씩 선정하여 디지털 포렌식 관점에서 각 지갑별 파일 시스템과 레지스트리, 브라우저 아티팩트, 메모리를 분석했다. 추가적으로 암호화폐별로 정규표현식을 통해 아티팩트 수집을 자동화한 툴(WinBAS)을 제안했다[5]. Park외 2인은 Atomic, Bitcoin Core, Bither 등 총 10개의 애플리케이션 형태의 암호화폐 지갑을 대상으로 디지털 포렌식 관점에서 지갑 주소, 트랜잭션 데이터 등의 아티팩트 수집 연구를 수행하고, 자동 정보 수집 도구를 제안했다[6]. 본 논문의 연구 대상인 브라우저 익스텐션 월렛 Metamask 및 Phantom에 대한 아티팩트 수집 및 분석 연구 또한 진행되었다. Son 외 1명은 Metamask의 levelDB를 획득하고 소스 코드를 분석하여 사용자의 패스워드를 통한 니모닉 코드 복호화 매커니즘을 확인했으며 levelDB 복호화와 정규표현식을 통한 아티팩트 추출을 수행하는 도구를 개발했다[7]. Kwon외 4명은 Metamask와 BitPay, Exodus와 Phantom 지갑 사이의 거래를 수행하고 지갑을 복원했을 경우 파일 시스템과 메모리에서 획득 가능한 암호화폐 복원정보 아티팩트들을 정리했다[8]. 또한 이후 수행된 Kwon의 연구에서 지갑 대상을 추가하여 Monero, BitPay, MetaMask, Phantom, Kaikas를 대상으로 암호화폐 복원정보 획득 실험과 잔존 실험을 수행한 결과를 발표했다[9]. 본 논문에서는 브라우저 익스텐션 월렛을 대상으로 파일 시스템과 메모리 영역에서 암호화폐 관련 아티팩트들을 수집하고, 수집 과정을 자동화하기 위한 알고리즘을 제시한다. 또한 브라우저 익스텐션 월렛 대상을 특정화하여 수집한 데이터 분석을 통해 유의미한 아티팩트를 추출하여 기존 논문들과 차별점을 둔다.

또한 최근 사용의 편리성으로 인해 브라우저 익스텐션 월렛의 사용자는 계속해서 증가하고 있지만 디지털 포렌식 관점에서의 암호화폐 지갑 분석 연구는 하드웨어 지갑이나 애플리케이션 형태의 지갑 위주로 수행되고 있으며, 브라우저 익스텐션 월렛을 대상으로 하더라도 비트코인이나 이더리움 화폐만을 다루고 있는 실정이다. 하지만 최근 NFT 거래나 스테이킹(Staking) 등을 통한 수익 창출에 다양한 암호화폐를 지원하는 암호 지갑이 사용됨에 따라 다양한 지갑 서비스를 대상으로 하는 디지털 포렌식 관점의 아티팩트 수집 및 분석 연구가 필요하다.

본 연구에서 대상으로 하는 브라우저 익스텐션 월렛 4종은 최소 수십만에서 수천만의 다운로드 수를 가지며, 각각 ETH(이더리움), BNB(바이낸스), SOL(솔라나), KLAY(클레이튼)를 주거래 화폐로 서비스한다. 따라서 다양한 암호화폐와 지갑에서 브라우저 아티팩트가 생성되는 결과와 형태, 지속성을 정리함으로써 유의미한 연구결과를 보인다. 또한 단순 검색 기반 니모닉 코드 탐색의 오탐률을 줄이기 위해 슬라이딩 윈도우 방식을 활용한 새로운 검색 방법 알고리즘을 제시한다.

III. 연구 대상 및 연구 방법

본 장에서는 연구 대상인 브라우저 익스텐션 월렛을 소개하고 연구에 사용된 도구와 연구 방법에 대해 기술한다.

3.1 브라우저 익스텐션 월렛

브라우저 익스텐션 월렛은 웹 브라우저의 확장 프로그램 형식으로 제공되는 암호화폐 지갑이다. 별도의 애플리케이션을 설치하거나 웹사이트로 이동할 필요 없이 웹 브라우저 상에서 손쉽게 암호화폐 자산을 확인하거나 거래를 수행할 수 있고, 하나의 지갑에서 여러 종류의 암호화폐를 지원한다는 장점이 있다.

본 연구에서 대상으로 하는 암호화폐 지갑인 Metamask[10], Binance[11], Phantom[12], Kaikas[13]는 모두 브라우저 익스텐션 월렛 형태의 지갑으로, 현재 많은 이용자를 보유 중이다. Tabel 1.은 연구 대상 지갑 4종에서 지원하는 암호화폐 종류와 버전 정보를 나타낸 표로, 대상 지갑 모두 공통적

Table 1. Version and Coin Type by Wallet

Wallet	Coin Type	Version
Metamask	ETH	10.24.2
Binance	BNB	2.13.7
Phantom	SOL	23.2.4
Kaikas	KLAY	2.3.0

으로 계정 생성, 암호화폐 구매, 송금, 교환 등의 기능을 제공한다.

3.2 연구 환경

실험과 분석은 Windows 10 운영체제를 사용하는 가상환경과 로컬 PC 상에서 수행했으며, 데이터 수집 시 데이터 오염을 방지하기 위해 가상머신을 사용했다. 가상환경의 운영체제는 Winodws 10 버전 22H2, 메모리 용량 16GB 이다.

데이터 수집 및 분석은 Metamask, Binance, Phantom, Kaikas가 공통으로 지원하는 브라우저인 구글 크롬 브라우저를 대상으로 선정하여 수행했다. 크롬 브라우저로부터 생성된 캐시 및 쿠키 데이터를 로컬 상에서 수집 후 분석했는데, 이때 사용한 도구는 Table 2.와 같다. 디지털 포렌식 조사의 목적은 용의자의 범죄 행위 입증을 위한 증거 획득으로, 본 연구에서는 브라우저 익스텐션 월렛을 사용해 실제 거래를 수행했을 때 획득 가능한 디지털 포렌식 아티팩트를 분석하기 위해 실제 사용자 행위를 모방하는 실험을 수행했다. 대상 지갑 4종에 대해 모두 코인 거래 실험을 수행했는데, Metamask의 경우 ETH 코인 거래 테스트를 위해 이더리움 네트워크의 테스트 네트워크인 Goerli Testnet 네트워크를 사용했다[14]. 그리고 Binance의 경우 Binance Smart Chain(BSC) 네트워크의 BNB 코인 거래를 테스트 하기 위해 Binance Smart Chian Test(BNBT)[15] 네트워크를 사용했다. 또한 Phantom의 경우 Solana 네크워크의 SOL 코인의 거래 테스트를 위해, Solana Devnet[16]을 사용했으며, Kaikas의 KLAY 코인 거래 테스트를 위해 Baobab 테스트넷[17]을 사용했다.

Table 2. Tools for Analyzing Browser Extension Wallet Artifacts

Tool	Version
Google Chrome	110.05481.178
ChromeCacheView	2.41
Magnet RAM Capture	1.02
VMware Workstation Pro	15.5.6 build-16341506
HxD	2.5.0

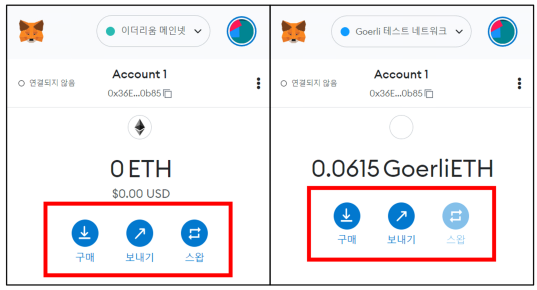

Fig. 1. Browser Extension Wallet Metamask Main Network and Goerli Testnet Network Functions

Fig. 1.은 각각 이더리움 메인넷 환경과 이더리움 테스트넷인 Goerli 환경의 Metamask 실행화면으로, 테스트넷에서는 메인넷과 다르게 구매, 송금 이외의 기능에서 제한이 발생하는 것을 확인할 수 있다. 이더리움 이외의 네트워크를 사용한 지갑에서도 마찬가지로 기능 제한이 존재하기 때문에 송금 거래 기능만 사용하여 실험을 수행했다.

3.3 연구 방법

디지털 포렌식 아티팩트 수집 및 분석을 위해 브라우저 익스텐션 월렛이 사용된 브라우저의 캐시 파일과 메모리 영역의 데이터를 분석했다. 전체 실험 과정은 Fig. 2.와 같다.

가상 환경에서 대상 지갑 4종 각각에 대해 로그

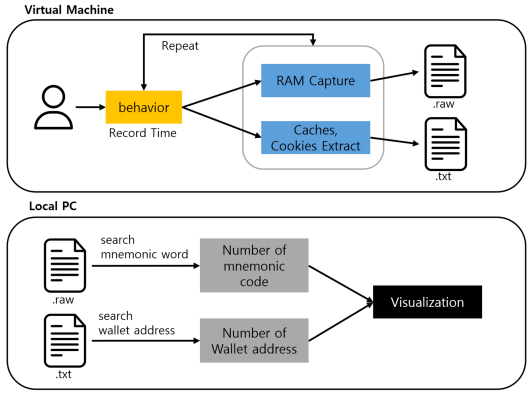

Fig. 2. Extract Artifacts and Visualization Process

인, 암호화폐 송금, 지갑 잠금, 브라우저 창 종료, 프로세스 종료 순서로 사용자 정상 행위를 수행하고, 각 행위마다 메모리 캡처와 브라우저 캐시 파일 추출을 반복했다. 이후 추출한 데이터를 로컬 PC로 옮겨 유의미한 아티팩트를 추출한 뒤, 지속성 및 무결성 측정을 위해 행위별로 수집된 아티팩트의 개수를 확인하고 그래프로 가시화했다. 실험은 크게 수집 과정과 데이터 추출 및 가시화 과정으로 나뉜다.

3.3.1 데이터 수집

데이터 수집은 Fig. 2. 그림의 상단 부분에 해당하는 과정으로 대상 지갑들이 공통적으로 제공하는 기능을 기반으로 실험을 수행했다. 지갑 4종 각각에 대해 로그인, 암호화폐 송금, 지갑 잠금, 브라우저 창 종료, 그리고 프로세스 종료 순서로 사용자 행위를 수행하고 아티팩트를 수집했다. 이러한 사용자 행위와 데이터 수집은 가상환경 상에서 수행했다. 각 행위의 전·후 시점에 크롬 캐시 및 쿠키 데이터를 .txt 파일로 수집하여 행위에 따라 생성된 데이터를 비교했고, 메모리 데이터는 행위 수행 후에만 수집했다. 메모리는 이미지 캡처를 통해 .raw 파일 형식으로 수집했다. 이때 메모리 이미지를 정상적으로 캡처하기 위해서는 가상환경 시스템이 종료되지 않도록 유의해야 한다. 또한 실험의 각 진행 과정 사이에 딜레이가 발생할 경우 데이터 손실이 발생할 수 있기 때문에, 모든 과정 수행에 2~3분 사이의 일정한 시간 간격을 두고 데이터를 수집했다. 총 5개 행위를 수행함에 따라 실험 종료 후에는 메모리 덤프 파일 5개, 캐시 및 쿠키 데이터 파일이 각 10개씩 생성된다.

3.3.2 데이터 추출 및 가시화

데이터 추출 및 가시화는 Fig. 2. 그림의 하단 Local PC에서 수행되는 과정이다. 그림의 .txt는 행위를 수행하기 전 시점과 후 시점에 PC에 기록된 모든 데이터이고, .raw는 행위 수행 후 캡처된 메모리 이미지의 덤프이다.

데이터 추출 과정에서는 캐시의 Last accessed 시간을 기준으로 가장 최근의 데이터 중, 행위 수행 전과 후의 데이터를 비교하여 각 지갑·행위와 관련된 API 호스트명, 데이터명, 리소스명을 가진 캐시를 추출했다. 쿠키 데이터 또한 캐시 데이터와 동일한 방법으로 추출했다. 메모리 덤프 파일의 경우 각 행위별로 수집한 파일에서 니모닉 코드를 검색하여 존재하는지 여부를 확인했다.

가시화 단계에서는 해당 시점 캐시 파일에서 추출한 지갑 주소 아티팩트의 개수를 확인하고, 마찬가지로 해당 시점의 메모리 덤프 파일에서 추출 할 수 있는 니모닉 코드 아티팩드의 개수를 확인하여 비교했다. 각 데이터가 수집된 시점을 기준으로 아티팩트의 개수를 꺾은선 그래프로 시각화하여 데이터의 지속성을 확인했다.

IV. 파일 시스템 아티팩트 수집 및 분석 결과

브라우저 익스텐션 월렛은 브라우저의 확장 프로그램으로, 행위 관련 정보가 브라우저 캐시 파일과 쿠키 데이터에 기록된다. 구글 크롬 브라우저의 캐시 파일과 쿠키 파일은 Table 3.에 작성된 로컬 경로에서 확인할 수 있다.

캐시 파일에는 각 지갑 프로그램에서 보내는 다양한 API Request URL이 기록되는데, 이 URL과 함께 전송되는 파라미터에 지갑 주소나 기타 메타데이터가 함께 기록되어 있다. 마찬가지로 전송되는 URL 정보를 포함한 쿠키 데이터에서도 지갑 주소가 함께 남는 경우가 있었다.

쿠키 파일에서는 디바이스 사용자 식별을 위한 _ga 쿠키 아티팩트를 수집할 수 있다. _ga 쿠키는 Google Analytics라는 구글 내의 플랫폼에서 생성하는 쿠키로, 해당 쿠키를 생성한 기기 및 브라우저 사용자에 따라 Fig. 3.과 같은 고유한 Client ID 값을 가진다. 웹사이트의 트래픽을 추적하기 위해 사용되는 값으로, 사용자가 동일한 기기로 동일한 웹사이트에 접속 시 같은 값의 _ga 쿠키를 가지기 때문에 이를 통해 동일 사용자를 식별할 수 있다.

캐시에서 확인할 수 있는 지갑 주소는 사용자의 거래 내역 추적, 암호화폐 자산 규모 확인에 활용할 수 있는 증거이며, 쿠키 데이터에서 추출 가능한 _g

Table 3. Path of Google Chrome Caches and Cookies files

Type	Path
Caches	%LOCALAPPDATA%\Google\Chrome\User Data\%PROFILE%\Cache\Cache_Data\data_1
Cookies	%LOCALAPPDATA%\Google\Chrome\User Data\%PROFILE%\Network\Cookies

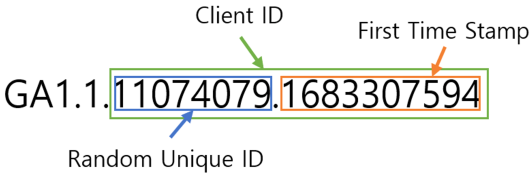

Fig. 3. _ga Cookie Data Form in Google Chrome

a 쿠키는 해당 쿠키가 생성된 기기와 사용자를 식별할 수 있는 정보이다. 크롬 브라우저에서 수집 가능한 암호 화폐 관련 캐시 및 쿠키 아티팩트는 4.1장부터 4.4장까지 지갑별로 분석한 결과를 정리하고 설명한다.

4.1 Metamask

Table 4.는 Metamask에서 기능 수행 시 생성되는 행위 관련 캐시 데이터를 정리한 표이다. 'metaswap' Gas API request URL 은 송금 거래 시 발생하는 gas 요금의 값을 반환해주는 API 요청 주소이다[18]. 해당 데이터를 통해 Metamask에서 수행된 거래 발생 시간, 거래 당시 ETH 거래 수수료 등을 확인할 수 있고, 용의자 기기에서의 ETH 거래 여부를 확인할 수 있다. 'etherscan.io' API request URL은 ETH 트랜잭션 탐색 API 요청 주소이다[19]. 해당 API 호출 시, URL에 노출되는 address 파라미터에 사용자의 계좌 주소가 함께 전달되기 때문에 온전한 지갑 주소를 획득할 수 있다. 'cryptocompare.com' API request URL은 ETH 코인의 현 시세를 세계 통화로 변환해 보여주는 API이다[20]. ETH 통화 정보가 생성된 파일을 json 형식으로 반환하여 파일이 남기 때문에 ETH 거래가 수행된 것을 확인할 수 있다. Table 5.는 Metamask 행위 관련 쿠키 데이터를 정리한 표이다. 크롬 브라우저에서 Metamask를 실행하면 ETH 탐색기인 'etherscan.io' 를 호스트로 가지는 웹사이트와 Web3 환경 블록체인 개발 관련 웹사이트의 호스트인 'infura.io' 관련 쿠키가 생성되는데[21], 해당 웹사이트에서 생성된 _ga 쿠키는 거래에 사용된 기기와 사용자 식별을 위한 유의미한 아티팩트로 사용할 수 있다. 그 근거로 Fig. 4. 그림은 'infura.io' 호스트의 _ga 쿠키 데이터를 예시로 보인다. 다른 시간에 접근하더라도 같은 크롬 사용자와 기기를 사용하면 같은 Client ID를 가지는 _ga 쿠키가 남는 것을 확인할 수 있다.

Table 4. Artifacts of Cache files for Metamask

Artifact	Discription	URL Format
'metaswap' Gas API request URL	Fee per gas	https://gas-api.metaswap.codefi.network/networks/{ChainID}/suggestedGasFees
'etherscan.io' API request URL	Transaction Information	https://api-goerli.etherscan.io/api?module=account&action=txlist&address={address}&tag=latest&page={pageNumber}&startBlock={BlockID}
'cryptocompare.com' request API URL	Price of ETH	https://min-api.cryptocompare.com/data/price?fsym=ETH&tsyms=USD

Table 5. Artifacts of Cookies for Metamask

HostName	Name	Value Foramt
etherscan.io	_ga Cookie	GA1.{domainComponent}.{clientID}.{timeStamp}
infura.io	_ga Cookie	GA1.{domainComponent}.{clientID}.{timeStamp}

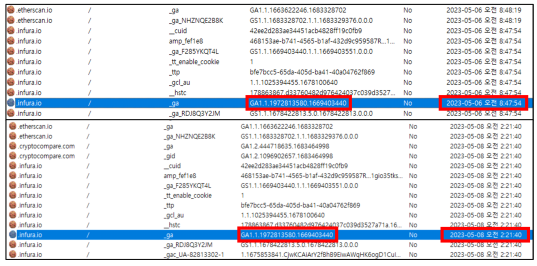

Fig. 4. Compare _ga Cookie at Different Times of 'infura.io' Host by Metamask

4.2 Binance

Table 6.는 Binance 지갑 실행 후 동작 시 생성되는 캐시 데이터를 정리한 표이다. Binance Logo Resource는 github 경로에 존재하는 Binanc

Table 6. Artifacts of Cache files for Binance

Artifact	Discription	URL Format
Binance Logo Resource	Binance logo	https://raw.githubusercontent.com/binance-chain/wallet-assets/master/assets/BNB/logo.svg
'binance.org' Gas API Request URL	Network information	https://wallet.binance.org/api/v1/bsc/gas?networkId={NetworkID}
'binance.org' Fiat API Request URL	Price of BNB	https://wallet.binance.org/api/v1/fiat
'binance.org' Balance API Request URL	Asset state of address	https://wallet.binance.org/api/v1/{NetworkID}/{address}/balance
'binance.org' Transaction API Request URL	All transaction information	https://wallet.binance.org/api/v1/{NetworkID}/address/{address}/transactions
'binance.org' Auth API Request URL	Response of Auth	https://wallet.binance.org/api/v1/venly/auth
'binance.org' Cryptocurrency API Request URL	Wallet infomation	https://wallet.binance.org/api/v1/cryptocurrency

e 지갑의 로고를 호출한 URL 정보이다. 그리고 'binance.org'로의 API Request 캐시가 다수 생성되는 것을 확인할 수 있다. 'binance.org'는 공식 BNB Chain 공식 홈페이지로 여러 API를 제공한다[22]. 그 중, 'binance.org' Balance API Request URL과 'binance.org' Transaction API Request URL 데이터에는 지갑 주소가 함께 전달된 URL이 존재하여 지갑 주소를 확인할 수 있다. 공식 API의 호스트명과 지갑 주소를 통해 Binance 지갑과 BNB 코인이 사용된 것을 파악할 수 있다.

Table 7.은 브라우저 쿠키 데이터에서 획득 가능

Table 7. Artifacts of Cookies for Binance

Host Name	Name	Value Format
.moonpay.com	ld_device_id Cookie	{device_i(f[0-9]{7}-[0-9]{4}-[a-z0-9{4}-[0-9]{4}-[a-z0-9{4}-[0-9]{4}-[a-z0-9]{12}]]
.moonpay.com	_ga Cookie	GA1.{domainComponent}.{clientID}.{timeStamp}
.moonpay.com	OptanonConsent Cookie	sGpcEnabled=0&datestamp=S{time} (?i'"ëŽ… ?i' - ???&version={versionInfo}&isIABGlobal={isIABGlobal}&hosts=&consentId={consentId}&interactionCount=1&landingPath={landingPath}apiKey={apiKey}&defaultCurrencyCode=SOL&enabledPayments={payment}&walletAddresses={address}&colorCode={colorCode}&baseCurrencyAmount={baseCurrencyAmount}&quoteCurrencyAmount=&baseCurrencyCode={baseCurrencyCode}&signature={signature}%3D&groups={groups}
.coingecko.com	_ga Cookie	GA1.{domainComponent}.{clientID}.{timeStamp}
.bnbchain.org	_ga Cookie	GA1.{domainComponent}.{clientID}.{timeStamp

한 Binance 지갑 관련 아티팩트를 정리한 표이다. 각 웹사이트 '.moonpay.com', '.coingecko.com'와 '.bnbchain.org'에서 _ga 쿠키가 생성되어 쿠키가 생성된 기기와 사용자 식별이 가능하다. '.moonpay.com'는 Web3 환경에서 결제 서비스를 제공하는 수단의 웹사이트이다[23]. '.coingecko.com'는 암호화폐 블록체인의 트랜잭션 탐색 기능과 코인 정보들을 제공하는 웹사이트이다[24]. 그리고 '.bnbchain.org'는 BNB Chain 네트워크 관련 서비스를 제공하는 웹사이트이다[25]. '.moonpay.com' 관련 쿠키는 _ga 쿠키 이외에 디바이스 식별을 위한 ld_device_id 쿠키, 그리고 지갑 주소가 포함된 OptanonConsent 쿠키가 남는 것을 확인할 수 있다. Binance 지갑은 캐시와 쿠키 모두에서 지갑 주소 획득이 가능하다.

4.3 Phantom

Phantom 지갑 실행 후, 동작 수행 시 캐시 파일에는 생성되는 데이터는 Table 8.과 같이 정리했다. 캐시에는 'phantom.app' API 호출 데이터 URL들이 생성되는데, 이는 Phantom 애플리케이션의 호스트명으로 각종 API를 자체 제공한다[26]. 이 중 'phantom.app' Token API requst URL들과 'phantom.app' History API requst URL의 파라미터에서 지갑 주소를 획득할 수 있다. 모든 URL이 공통적으로 가지는 데이터인 'api.phantom.app/solana/'에는 API 호스트명과 네트워크 정보가 남아있으므로, Phantom 애플리케이션 사용 여부와 SOL 코인 사용 여부를 확인할 수 있다.

쿠키 데이터에서는 'phantom.app'을 호스트 이름으로 가진 'ph_ns5ZjyZlgS_mGfthS7nZ33FaP1mFcbBvon3LPDNahmQ_posthog' 쿠키가 생성되는데, 해당 쿠키에는 device_id 값이 포함되어 있다. device_id는 기기 식별 정보로 항상 동일한 값을 가지기 때문에 해당 기기에서의 Phantom 지갑 사용 여부를 확인할 수 있다. device_id와 관련된 구체적인 형식은 Table 9.에서 확인할 수 있다.

Table 8. Artifacts of Cache files for Phantom

Artifact	Discription	URL Format
'phantom.app' Token API requst URL	Wallet ChainID Information	https://api.phantom.app/solana/spl_token_accounts?ownerPubKey={address}&chainId=102
		https://api.phantom.app/solana/spl_token_accounts?ownerPubKey={address}&chainId=101
		https://api.phantom.app/solana/spl_token_accounts?chainId=102&ownerPubKey={address}
		https://api.phantom.app/solana/nft/v1/pubkey/{address}/listings?chainId=102
'phantom.app' History API requst URL	Request Wallet History	https://api.phantom.app/solana/history/v1/pubkey/{address}
'phantom.app' Health API requst URL	Locale and TPS Information	https://api.phantom.app/solana/health/v1?locale=ko-KR

Table 9. Artifacts of Cookies for Phantom

Host Name	Name	Value Format
phantom.app	ph_ns5ZjyZlgS_mGfthS7nZ33FaP1mFcbBvon3LPDNahmQ_posthog	{"distinct_id":"{distinct_id}","$device_id":"{device_id}","$referrer":"$direct","$referring_domain":"$direct","$sesid":[sesid] ... }

4.4 Kaikas

Kaikas의 경우, 행위 수행 시 다른 대상 지갑과는 달리 캐시 및 쿠키 데이터에서 지갑 주소를 획득할 수 없었다. 또한 쿠키 데이터에서 유의미한 아티팩트를 확인할 수 없었다.

Table 10.은 크롬 캐시 파일에서 획득 가능한 Kaikas 관련 아티팩트를 정리한 표이다. 'kaikas.io' API request URL은 한글로 작성된 Kaikas 이용 지연 안내 및 주의사항이 작성된 json 파일을 반환한다[27]. Phantom 지갑과 동일하게 API 호스트명이 'kaikas.io'이므로 해당 아티팩트로부터 Kaik

Table 10. Artifacts of Cache files for Kaikas

Artifact	Discription	URL Format
'kaikas.io' API request URL	Guidelines for delay in use and precautions	https://static.kaikas.io/extension/urgent.json?1677836763010
		https://static.kaikas.io/extension/token-list.json?_t=1677836774824
		https://static.kaikas.io/extension/notice.json?_t=1677836786644

as 지갑의 사용 여부를 알 수 있다.

4.5 캐시 및 쿠키 파일 지갑 주소 아티팩트 검색 방안

캐시 및 쿠키 파일에서 지갑 주소 추출 시, 지갑 주소를 획득할 수 있었던 URL의 API 호스트명과 주소를 전달하는 파라미터명을 이용해 문자열로 검색하면 해당하는 캐시 데이터의 위치를 찾을 수 있다. 또한 해당하는 캐시 데이터를 추출한 뒤 최소한의 오탐을 방지하기 위해 암호화폐별로 각각 정규표현식을 매치하여 주소 형태가 맞는지 확인할 수 있다. 연구 대상 지갑 중 ETH, BNB, KLAY 코인의 지갑은 지갑 주소의 길이는 40이고 16진수 문자열로 구성되어 있어 '0x[a-fA-F0-9]{40}' 형태의 정규표현식을 적용할 수 있다. SOL 코인의 지갑은 길이 44의 알파벳과 숫자가 혼합된 문자열로 '[a-zA-Z0-9]{44}' 형태의 정규표현식이 적용된다.

V. 메모리 아티팩트 수집 및 분석 결과

본 장에서는 행위 실험 과정에서 캡쳐한 메모리 덤프 파일에서 획득 가능한 아티팩트에 대해 기술하고, 슬라이딩 윈도우 방식을 통해 니모닉 코드 추출을 자동화하는 방식을 제안한다.

5.1 메모리 아티팩트 수집 및 분석 결과

니모닉(Mnemonic) 코드란 암호화폐 지갑의 비밀 키를 복구하기 위해 사용되는 일련의 단어 패턴으로, 'able', 'baby' 등 일상적이고 쉬운 단어로 이뤄진 12개 또는 24개의 단어로 구성된다. 니모닉 코드 자체를 이루고 있는 2048개의 단어는 BIP 표준에 따라 정의되고 공개되어 있으며, 순서를 포함한 단어들의 특수한 조합이 지갑의 비밀 키 복구 구문으로 사용된다. 또한 니모닉 코드로부터 마스터 시드를 생성하고, 마스터 시드로부터 개인 키와 공개 키를 유도해 지갑 주소를 생성하기 때문에 니모닉 코드는 지갑의 모든 정보와 권한을 얻기 위한 핵심 정보로 사용된다. 마스터 시드로부터 개인 키와 공개 키를 유도하는 방식은 암호화폐별, 암호화폐 지갑별로 다르며 BIP-32, BIP-39, BIP-44 등의 규칙을 따른다. BIP(Bitcoin Improvement Proposal)란 비트코인 기능개선을 위해 제안된 기술 문서로[28] 그 중 BIP-39는 마스터 시드 및 키 유도 과정에서 니모닉 코드를 사용하는 형식이다. 본 연구에서 대상으로 하는 지갑 4종 모두 BIP-39 형식에 따라 니모닉 코드를 사용한다. 캡쳐한 메모리 덤프 파일에서 검색 기능을 통해 Fig. 5.와 같이 니모닉 코드를 획득할 수 있다.

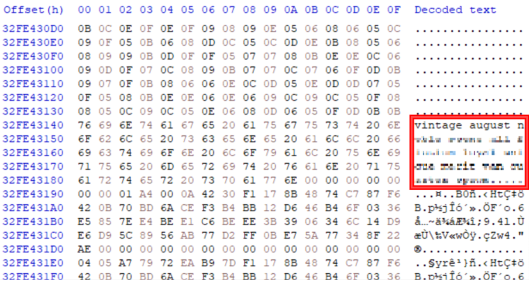

Fig. 5. Cryptocurrency Secret words in Memory raw file

5.2 슬라이딩 윈도우 방식 기반 니모닉 코드 아티팩트 자동 추출 알고리즘

연구에서 대상으로 하는 브라우저 익스텐션 월렛 4종은 모두 BIP-39 표준을 따르며, 각 지갑의 니모닉 코드는 공식 BIP-39 Wordlist에 따라 2048개의 단어 중 12단어 혹은 24단어로 구성된다. 본 절에서는 윈도우 슬라이드 기법을 활용해 방대한 크기의 메모리 데이터에서의 니모닉 코드 추출을 자동화하기 위한 알고리즘을 제안한다.

메모리 영역에서 확인 가능한 암호화폐 지갑의 니모닉 코드는 Fig. 5.와 같이 단어들이 공백으로 구분되어 나열된 형태로 존재한다. 단순 검색을 통해 BIP-39 Wordlist에 속하는 단어를 찾을 수 있지만, 매우 일상적이고 쉬운 단어로 이루어져 있어 단순 검색에 의한 탐색은 오탐 확률이 높다. 단어 검색 시 단어 간 공백을 확인하고, 전체 단어 탐색을 위해 슬라이딩 윈도우 방식을 적용할 경우 오탐 확률을 낮추며 니모닉 코드를 자동 추출할 수 있다. Fig. 6.은 제안하는 알고리즘에 대한 수도 코드이다. 우선 메모리 데이터에서 BIP-39 Wordlist와 일치하는 단어 패턴의 아스키코드가 있는지 확인한다. 일치하는 단어가 검색되면, 바이트 단위로 슬라이딩 윈도우를 수행하여 단어의 이전 문자와 이후 문자를 비교해 0x20(공백) 값을 확인하고 다시 다음 단어를 검색하는 과정을 반복하여 니모닉 코드를 추출할 수 있다.

```
Algorithm 1. Get Mmnemonic code in memory
-----------------------------------------------
Input : memory.raw M, BIP39 Wordlist W
Output : Mmnemonic code N
f1 = open('memory.raw')
f2 = open('BIP39.txt')
M = f1.readlines()
W = f2.readlines()
cnt = 0
N = []
for line in W:
    //search first word
    idx = M.find(line)
    if idx True:
        N.append(line)
        while(cnt==11):
            //if ' ' exist back
            if M[idx-1] == 0x20:

                for line in W:
                    idx = M[:idx-1].find(line)
            //if ' ' exist afeter
            elif M[idx+line.length+1] == 0x20:
                for line in W:
                    idx = M[idx+line.length+1:].
find(line)
            else:
                continue
            N.append(line)
            cnt++
        return N
```

Fig. 6. Psuedo code for Get Mmnemonic code in Memory raw file

VI. 사용자 행위별 아티팩트 지속성

브라우저 익스텐션 월렛은 브라우저의 확장 프로그램으로 동작하기 때문에 브라우저 프로세스에 종속적이다. 프로세스 메모리는 프로세스의 동작을 실시간으로 처리하기 위한 임시 저장 공간으로, 프로세스의 실행 동작에 따라 저장되는 데이터가 크게 달라진다. 따라서 크롬 브라우저 종료, 지갑 로그아웃 및 잠금 등 사용자의 가변적 행위에 따른 아티팩트의 수집의 변화와 지속성을 정리할 필요가 있다. 아티팩트의 지속성이란 프로세스 종료 혹은 지갑 로그아웃 등의 행위가 발생한 후 대상 암호 지갑 4종과 관련한 아티팩트가 유지되어 추출이 가능한지 여부를 뜻한다.

행위에 따른 아티팩트의 변화와 지속성을 한눈에 확인하기 위해 사용자 정상 행위에 해당하는 로그인, 송금, 지갑 잠금, 브라우저 탭 닫기, 프로세스 종료를 수행했다. 각 행위 수행 후 지갑 주소와 니모닉 코드 획득을 위한 수집 절차를 거치고, 시간대별로 획득 가능한 아티팩트의 개수를 그래프로 나타냈다.

Fig. 7., Fig. 8., Fig. 9., Fig. 10.은 각각 Metamask, Binance, Phantom, Kaikas에서 시간 변화에 따라 추출 가능한 지갑 주소·니모닉 코드 관련 아티팩트를 나타낸 그래프이다. 다만, 캐시 파일에서 지갑 주소를 확인할 수 없었던 Kaikas는 메모리에서 추출 가능한 니모닉 코드 관련 아티팩트만 표기했다. 아티팩트 개수는 본 논문의 3장에 기술된 과정과 동일하게 캐시 및 쿠키 데이터에서 추출한 파일과 메모리 캡처를 통해 추출한 메모리 덤프 파일에서 확인 가능한 아티팩트의 결과이다.

분석을 통해 대상 지갑 4종 모두 로그인, 송금 행위 수행 시 브라우저에서 수집 가능한 캐시 데이터 관련 아티팩트의 개수가 증가하며, 브라우저 아티팩트가 메모리 아티팩트에 비해 상대적으로 지속성이 높다는 것을 확인했다. 메모리 아티팩트 역시 로그인, 송금 행위 수행 시 아티팩트 개수가 증가하지만, 지갑 잠금, 탭 종료 행위가 수행될 경우 아티팩트의 지속 시간이 감소하는 것을 확인할 수 있다.

Fig. 7.는 Metamask의 아티팩트 지속성을 나타낸 그래프이다. 캐시 데이터의 지갑 주소 관련 아티팩트는 로그인 행위 이후 동작이 수행될 때마다 기록되는 데이터의 수가 증가하여, 크롬 프로세스가 종료된 이후에도 데이터의 수가 그대로 유지되는 것을 확인했다. 메모리 덤프 파일에서 획득한 니모닉 코드 관련 아티팩트는 로그인 행위 이후부터 생성되며, 지갑 잠금 동작 수행 시부터 획득 가능한 니모닉 코드의 개수가 감소한다. 하지만 크롬 프로세스가 종료된 이후에도 획득 가능한 아티팩트가 존재함을 확인했다.

Fig. 8.는 Binance 아티팩트의 지속성을 나타낸 그래프이다. Binance의 경우 Binance는 Metamask와 마찬가지로 로그인 직후 캐시 및 쿠키 데이터

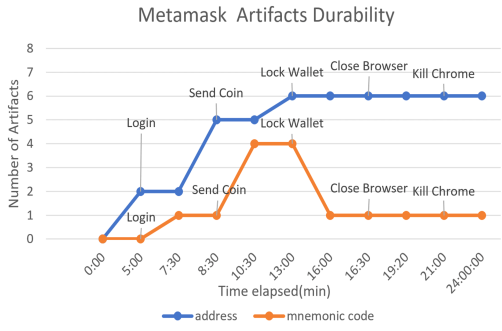

Fig. 7. Metamask Artifacts Count for Persistence

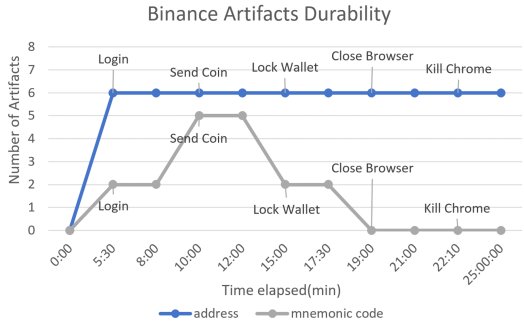

Fig. 8. Binance Artifacts Count for Persistence

의 지갑 주소를 확인할 수 있다. 또한 지갑 잠금, 브라우저 종료 등의 행위를 하더라도 지갑 주소 관련 아티팩트의 개수가 유지된다는 특징이 있다. 로그인 이후부터 지갑이 실행되는 동안은 메모리에서 온전한 니모닉 코드를 획득할 수 있었지만, 크롬 프로세스를 종료할 경우 메모리에서 니모닉 코드 데이터가 사라지는 것을 확인했다.

Fig. 9.은 Phantom의 아티팩트 지속성을 나타낸 그래프이다. Phantom 지갑은 로그인 시부터 캐시 데이터의 지갑 주소 관련 아티팩트가 생성되며, 크롬 프로세스가 종료된 후에도 아티팩트가 지속된다. 로그인 이후 송금 거래 기능을 사용하면 니모닉 코드를 획득할 수 있다. 하지만 Metamask, Binance와는 다르게 브라우저 창을 종료하지 않더라도 지갑에서 잠금 기능을 실행하면 메모리에서 니모닉 코드가 사라져 획득할 수 없다는 특징이 있다.

마지막으로 Fig. 10.은 Kaikas의 아티팩트 지속성을 나타낸 그래프이다. Kaikas 지갑의 경우 캐시 및 쿠키 데이터에서 지갑 주소 관련 아티팩트를 획득할 수 없다. 메모리 덤프 파일에서는 로그인 이후부터 니모닉 코드 획득이 가능하나, Phantom과 마찬

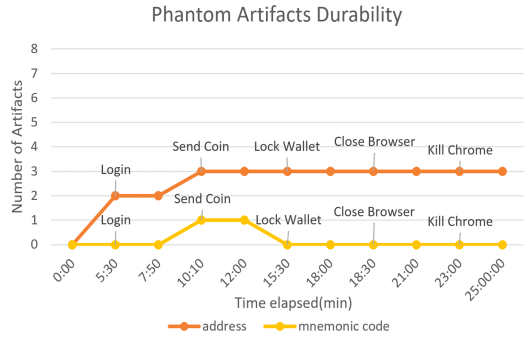

Fig. 9. Phantom Artifacts Count for Persistence

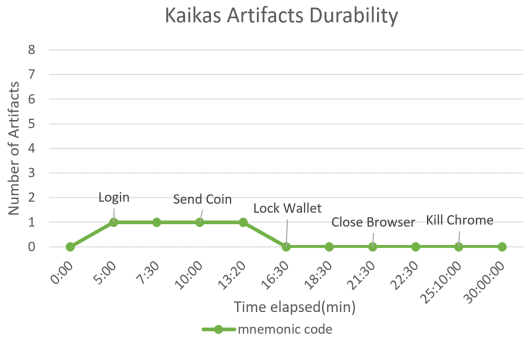

Fig. 10. Kaikas Artifacts Count for Persistence

가지로 브라우저 창을 종료하지 않아도 지갑이 잠기면 메모리에서 니모닉 코드가 사라지는 것을 확인했다.

브라우저 익스텐션 월렛 4종은 타겟 아티팩트인 지갑 주소와 니모닉 코드에 대해 공통적으로 두 가지 특징을 보인다.

첫 번째로 캐시 및 쿠키 파일에서 수집한 지갑 주소의 경우 지갑 잠금, 브라우저 종료 행위가 수행되더라도 데이터가 일정 시간 유지된다. 파일 시스템에서 추출한 정보로, 비휘발성 데이터이기 때문에 지속성이 길다는 특징을 가진다.

두 번째로 사용자 행위에 따라 메모리 데이터에서 수집 가능한 니모닉 코드의 개수가 달라지는 것을 확인할 수 있다. 이는 지갑별로, 수행되는 행위별로 메모리에 저장되는 변수의 수가 달라지기 때문이다. 또한 대부분의 경우 지갑이 잠기거나 브라우저가 종료되면 메모리의 프로세스가 종료되어 니모닉 코드 데이터 또한 지워진다. 즉, 니모닉 코드의 경우 Metamask를 제외한 지갑 3종에서는 프로그램이 실행될 때만 관련 데이터를 획득할 수 있다는 한계가 있다.

VII. 디지털 포렌식 관점에서 아티팩트 활용

디지털 포렌식 관점에서 브라우저 익스텐션 월렛 4종을 분석한 결과 브라우저 캐시 파일과 쿠키 데이터, 그리고 메모리에서 월렛의 사용에 따른 사용자 행위 관련 아티팩트를 수집할 수 있음을 확인했다.

각 지갑 별로 캐시 및 쿠키 데이터에서 수집 할 수 있는 지갑 주소 아티팩트는 해당 지갑이 사용된 트랜잭션과 거래 내역을 조사할 수 있는 정보로, 자금 세탁 추적 기술 등을 통해 범죄 수익금 환수에 도움을 줄 수 있다.

쿠키 데이터에서 획득할 수 있는 _ga쿠키의 Client ID는 용의자의 암호화폐 거래 행위 입증에 도움을 줄 수 있는 증거로, 디지털 포렌식 수사 과정에서 유의미하게 사용될 것으로 기대된다.

또한 메모리 덤프를 통해 수집할 수 있는 니모닉 코드는 그 자체로 암호화폐 지갑의 비밀 복구 구문이므로, 니모닉 코드를 안다면 해당 사용자의 지갑을 바로 복구 가능하다. 지갑을 복구함으로써 해당 암호화폐 지갑의 사용자 권한을 가지게 되면 범죄 수익금 환수 가능성이 비약적으로 증가한다. 따라서 니모닉 코드 아티팩트는 디지털 포렌식 수사에 있어 중요 증거로 활용될 수 있다. Table 11.은 분석 대상 지갑 4종에서 주요 아티팩트 획득 여부를 정리한 표이다.

실험을 통해 Metamask와 Binance의 경우 지갑 주소 정보, Client ID, 니모닉 코드를 모두 획득할 수 있었다.

Phantom의 경우 Client ID 데이터를 제외한 지갑 주소와 니모닉 코드를 획득할 수 있었고, Kaikas의 경우에는 메모리에서 니모닉 코드만 획득할 수 있었다. 대상 지갑 모두 라이브 상태일 때 수사가 이뤄진다면 지갑 복구 구문인 니모닉 코드를 메모리에서 획득할 수 있어 지갑 복원 및 자금 환수가 가능할 것이다.

Table 11. Artifacts for Browser Extension Wallet

Wallet	Address	Client ID	Mnemonic Code
Metamask	O	O	O
Binance	O	O	O
Phantom	O	X	O
Kaikas	X	X	O

VIII. 결 론

본 논문에서는 개인 암호화폐 지갑 서비스인 브라우저 익스텐션 월렛의 사용자가 증가함에 따라, 그 중 다운로드 수가 높은 서비스 4종(Metamask, Binance, Phantom, Kaikas)을 대상으로 아티팩트 수집 및 분석 연구를 수행했다. 각 지갑들의 로그인, 암호화폐 송금, 지갑 잠금, 브라우저 창 종료, 프로세스 종료 행위 시점에 따라 각 파일시스템 및 메모리 데이터를 수집하고, 유의미한 아티팩트를 추출하고 지속성을 확인했다. 그 결과 크롬 브라우저의 캐시 및 쿠키 파일에서 지갑의 주소와 장치 및 사용자 식별자를 확인했으며, 메모리 데이터에서 니모닉 코드를 수집할 수 있음을 확인했다. 수집한 니모닉 코드와 사용자 계정 정보 관련 아티팩트는 암호화폐를 이용한 범죄 사건을 수사하는 과정에서 관련 거래 추적 및 사실 확인, 범죄수익 환수를 위한 중요 단서 및 증거로 활용될 수 있다. 암호화폐를 사용한 범죄 디지털 포렌식 수사 과정에서 거래 내역 수집 및 불법 자금의 환수를 위한 방안으로 활용될 것으로 기대된다.

다만 파일 시스템과 달리 메모리에서 수집한 아티팩트는 휘발성 데이터로, 크롬이나 지갑 프로세스가 종료되면 니모닉 코드 아티팩트를 수집하기 어렵다. 수집을 위해서는 지갑이 활성화되어 있는 상태여야 하며, 라이브 상태에서의 수사가 필수적이라는 한계가 있다. 이에 따라 프로세스 사용량이 많거나, 메모리 용량이 본 연구의 실험 환경보다 작아진다면 니모닉 코드 잔존 여부가 달라질 수도 있다. 또한 지갑에 따라 로그인이나 지갑 잠금 행위에 의해 니모닉 코드 잔존 여부가 달라지는 것을 확인했다. 본 연구의 대상 지갑 4종은 오픈 소스이므로, 추후 각 지갑의 소스 코드를 분석하여 프로세스 메모리 구조적인 관점에서 입력된 계정 정보 및 니모닉 코드의 플로우를 확인할 예정이다. 또한 현재는 연구 대상이 지갑 4종으로 한정되어 있으나, 추후 대상 지갑을 추가하고 모바일 앱으로 상용되고 있는 서비스까지 범위를 넓혀 실험을 통해 다양한 지갑의 복구 방안을 도출할 계획이다. 추가적으로 암호화된 개인 키의 복호화 메커니즘과 니모닉 코드를 활용한 개인 키 유도 메커니즘을 찾아 암호지갑 복호화 도구를 개발할 예정이다.

References

[1] Understanding the blockchain investigation method of preventing money laundering using cryptocurrency., “money laundering”, IT Word, 2022.03.07, 1

[2] “Room N ′Devil′s Deal′...Cryptocurrency grows and cryptocurrency catch”, “cryptocurrency crime”, News1, 2020.03.27, 1

[3] T. Thomas, M. Piscitelli, I. Shavrov and I. Baggili, “Memory FORESHADO

W: Memory FOREnSics of HArDware CryptOcurrency wallets - A Tool and Visualization Framework," Forensic Science International: Digital Investigation, vol. 33, pp.301002-301012, Jul. 2020.

[4] Min-taek Lim, Jeong-yoon Kang, Jun-sung Park, Moon-gyu Lee and Hyeon-deok Jeoung, "A Study on the Extraction of Cryptocurrency Transaction Information Based on Memory Analysis," Journal of Digital Forensics, 16(1), pp.99-117, Mar. 2022.

[5] S. Zollner, K.K.R. Choo and N.A. Le-Khac, "An Automated Live Forensic and Postmortem Analysis Tool for Bitcoin on Windows Systems," IEEE Access, vol. 7, pp. 158250-158263, Oct. 2019.

[6] Hyun-jae Park, So-jeong Kim and Jung-heum Kim, "A study on techniques to assist cryptocurrency-related investigationsthrough collecting and analyzing OS and application artifacts on," Journal of Digital Forensics, 16(4), pp. 138-150, Dec. 2022.

[7] Ji-hun Son and Jung-heum Park, "Forensic analysis of MetaMask cryptocurrency wallet artifacts," Journal of Digital Forensics, 16(4), pp.151-165, Dec. 2022.

[8] Hyeon Kwon, Kyung-ju Lee, Ha-young Kim, Yeong-woong Kim and Gi-bum Kim. "Seizure of Ransomwar e Group's Virtual Assets Using Network Investigative Technique," Journal of Digital Forensics, 16(3), pp.130-142, Sep. 2022.

[9] Hyeon Kwon. "Acquisition Method of Virtual Asset Restoration Information in Memory," Master's thesis, Sungkyunkwan University, Feb. 2023.

[10] Chrome Web Store, "Metamask", valibale on : https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn?hl=ko, accessed 2023.05.08.

[11] Chrome Web Store, "Binance chrome extension", https://chrome.google.com/webstore/detail/binance-wallet/fhbohimaelbohpjbbldcngcnapndodjp?hl=ko, accessed 2023.05.08.

[12] Chrome Web Store, "Phantom chrome extension", https://chrome.google.com/webstore/detail/phantom/bfnaelmomeimhlpmgjnjophhpkkoljpa?hl=ko, accessed 2023.05.08

[13] Chrome Web Store, "Kaikas chrome extension", https://chrome.google.com/webstore/detail/kaikas/jblndlipeogpafnldhgmapagcccfchpi?hl=ko, accessed 2023.05.08.

[14] GOERLI FAUCET, "goerlifaucet", https://goerlifaucet.com/, accessed 2023.03.10.

[15] Binance Smart Chian Test(BNBT) "Binance testnet", https://testnet.binance.org/faucet-smart/, accessed 2023.05.08.

[16] Solana Devnet, "solana testnet", https://solfaucet.com/, accessed 2023.05.08.

[17] KLAY Faucet, "klaytn testnet", https://baobab.wallet.klaytn.foundation/faucet/, accessed 2023.05.08.

[18] MetaSwap, "metaswap", https://support.metamask.io/hc/en-us/articles/4405093054363-User-Guide-Swaps, accessed 2023.05.08.

[19] Etherscan, "etherscan api", https://docs.etherscan.io/api-endpoints/accounts, accessed 2023.03.10.

[20] cryptocompare, "cryptocompare", https://min-api.cryptocompare.com/documentation?key=Price&cat=SingleSymbolPriceEndpoint, accessed 2023.05.08.

[21] INFURA, "infura api", hhttps://docs.infura.io/infura/networks/ethereum/how

-to/make-requests, accessed 2023.05.08.
〔22〕 Binance, "binance", https://docs.bnbchain.org/docs/overview, accessed 2023.03.10.
〔23〕 Moonpay, "moonpay", https://www.moonpay.com/ko/about-us, accessed 2023.03.10.
〔24〕 CoinGecko, "coingecko", https://www.coingecko.com/en/api/documentation, accessed 2023.05.08.
〔25〕 BNB Chain, "bnb chain", https://forum.bnbchain.org/, accessed 2023.05.08.
〔26〕 Phantom, "phantom", https://help.phantom.app/hc/en-us/articles/13515761228051-Security-Tips-for-Phantom-users, accessed 2023.03.10.
〔27〕 Kaiks, "kaias", https://docs.kaikas.io/02_api_reference/01_klaytn_provider, accessed 2023,05.08
〔28〕 Github, "bitcoin bip", https://github.com/bitcoin/bips, accessed 2023.03.10.

〈저자소개〉

김 주 은 (Ju-eun Kim) 학생회원
2022년 2월: 동의대학교 컴퓨터공학과 졸업
2022년 3월~현재: 서울과학기술대학교 대학원 컴퓨터공학과 석사과정
〈관심분야〉 디지털 포렌식, 암호화폐, 정보보호

서 승 희 (Seung-hee Seo) 학생회원
2017년 2월: 서울과학기술대학교 컴퓨터공학 졸업
2019년 2월: 서울과학기술대학교 컴퓨터공학 석사
2020년 3월~현재: 서울과학기술대학교 대학원 컴퓨터공학과(박사과정)
〈관심분야〉 모바일 포렌식, 메모리 포렌식, 디지털 포렌식

석 병 진 (Beong-jin Seok) 종신회원
2012년 3월~2017년 8월: 서울과학기술대학교 컴퓨터공학과 졸업
2017년 9월~2019년 2월: 서울과학기술대학교 일반대학원 컴퓨터공학과 석사
2019년 3월~2023년 2월: 서울과학기술대학교 컴퓨터공학과 박사
2023년 3월~현재: 서울과학기술대학교 전기정보기술연구소 연구원
〈관심분야〉 정보보호, 암호학, 디지털 포렌식

변 현 수 (Heoyn-su Byun) 학생회원
2023년 2월: 서울과학기술대학교 산업공학과, 컴퓨터공학과 졸업
2023년 3월~현재: 서울과학기술대학교 대학원 컴퓨터공학과 석사과정
〈관심분야〉 디지털 포렌식, 사이버 보안, 정보보호

이 창 훈 (Chang-hoon Lee) 종신회원
2001년 2월: 한양대학교 자연과학부 수학전공 졸업
2003년 2월: 고려대학교 정보보호대학원 석사
2008년 2월: 고려대학교 정보경영전문대학원 정보보호전공 박사
2008년 3월~2008년 2월: 고려대학교 정보보호연구원 연구교수
2009년 3월~2012년 2월: 한신대학교 컴퓨터공학부 조교수
2012년 3월~2015년 2월: 서울과학기술대학교 컴퓨터공학과 조교수
2015년 3월~2020년 2월: 서울과학기술대학교 컴퓨터공학과 부교수
2020년 3월~현재: 서울과학기술대학교 컴퓨터공학과 교수
〈관심분야〉 암호, 디지털 포렌식, 사이버보안