https://doi.org/10.33778/kcsa.2024.24.2.019

악성 크립토재킹 대응을 위한 탐지 환경별 동향 분석 및 클라우드 환경에서의 탐지 프레임워크 제안★

유 지 원*, 강 서 연**, 이 수 미**, 김 성 민***

요 약

크립토재킹 공격은 암호 화폐 채굴에 필요한 컴퓨팅 자원을 탈취하여 사용자의 가용성을 침해하는 공격이다. 공격의 대상은 일반적인 데스크톱이나 서버 환경에서부터 클라우드 환경까지 점차 다변화되고 있다. 따라서 다양한 컴퓨팅 환경에 적합한 크립토 마이너 탐지 기법의 적용이 필수적이다. 하지만 기존의 탐지 방법론들은 특정 환경에서만 탐지가 시행되었기 때문에 환경별로 적용할 수 있는 방법론에 대해서 비교분석이 제대로 수행되지 않았다. 따라서 본 연구에서는 종래의 크립토 마이너 탐지 기법들에 대한 분류 기준을 수립하고, 각자 다른 실험 환경과 데이터 셋을 기반으로 한 기존의 크립토 마이너 탐지 기법에 대한 심층적인 비교분석을 통해 클라우드 환경에서 적용 가능한 복합적이고 통합적인 탐지 프레임워크를 제시한다.

Analysis of Trends in Detection Environments and Proposal of Detection Frame work for Malicious Cryptojacking in Cloud Environments

Jiwon Yoo*, Seoyeon Kang**, Sumi Lee**, Seongmin Kim***

ABSTRACT

A crypto-jacking attack is an attack that infringes on the availability of users by stealing computi ng resources required for cryptocurrency mining. The target of the attack is gradually diversifying from general desktop or server environments to cloud environments. Therefore, it is essential to ap ply a crypto-minor detection technique suitable for various computing environments. However, sinc e the existing detection methodologies have only been detected in a specific environment, comparat ive analysis has not been properly performed on the methodologies that can be applied to each env ironment. Therefore, in this study, classification criteria for conventional crypto-minor detection tec hniques are established, and a complex and integrated detection framework applicable to the cloud environment is presented through in-depth comparative analysis of existing crypto-minor detection techniques based on different experimental environments and datasets.

Key words : Cryptomining, Detection, Cryptojacking, Container

접수일(2024년 04월 18일), 수정일(1차: 2024년 05월 05일), 게재확정일(2024년 06월 10일)

★ 본 논문은 2024년도 산업통상자원부 및 한국산업기술진흥원의 산업혁신인재성장지원사업 (RS-2024-00415520)과 과학기술정보통신부 및 정보통신기획평가원의 ICT혁신인재4.0 사업의 연구결과로 수행되었음 (No. IITP-2022-RS-2022-00156310).

* 성신여자대학교/융합보안공학과(주저자)

** 성신여자대학교/융합보안공학과(공동저자)

*** 성신여자대학교/융합보안공학과(교신저자)

1. 연구 배경

분산 원장 기반의 블록체인 기술이 발전함에 따라, 중앙 집중적인 기존 형태의 금융 거래 서비스를 탈중앙화하기 위한 2세대 암호화폐의 핵심 기술인 스마트 컨트랙트 기반 서비스 및 디파이(DeFi) 시장의 규모가 증가하고 있다. 이때, 블록체인 기반 네트워크의 트랜잭션을 검증하고 안정적으로 유지하는 데에 기여한 자들에게 암호 화폐를 발급해 준다. 마이닝(mining) 또는 채굴이라고 불리는 행위를 통해 컴퓨팅 자원을 제공한 사람들은 금융자산의 가치를 갖는 암호 화폐를 획득할 수 있기에, GPU 가격의 폭등 및 암호 화폐 채굴을 위한 전용 하드웨어가 등장하기도 하였다.

이와 관련하여, 암호 화폐 채굴과 관련된 악성 행위인 크립토재킹(cryptojacking)으로 인한 피해 사례가 꾸준히 증가하였다. 크립토 재킹은 암호 화폐 거래소에 대한 해킹 시도 및 사용자 지갑 애플리케이션 내 기밀 정보를 탈취하던 과거의 공격 형태와 달리, 암호 화폐 채굴에 필요한 컴퓨팅 자원을 탈취하여 사용자의 가용성을 침해하고, 암호 화폐를 채굴하는 공격이다. 2023년 Microsoft Threat Intelligence에 따르면 기업을 대상으로 한 크립토재킹 공격 피해 규모는 약 300,000달러로 조사되었다[1].

공격의 대상 또한 일반적인 데스크톱이나 서버 환경에서부터 클라우드 환경까지 다변화되고 있으며,암호화폐 채굴의 난도 상승으로 더 많은 CPU 및 GPU 자원이 필요하게 되었다. 이에, 공격자들이 클라우드 환경에서 다량의 인스턴스를 통해 암호 화폐를 탈취하는 시도가 증가하고 있다. 대표적인 사례로는, 2018년 해커들이 미국의 기업인 테슬라의 클라우드 시스템으로 침투하여 암호 화폐 채굴에 클라우드 컴퓨팅 자원을 무단 탈취한 사건이 있다.[2] 이러한 크립토재킹 공격은 등장 이래로 꾸준하게 증가 추세를 보이고 있으며, 글로벌 보안 전문기업인 Sonic Wall 측에서 발행한 보고서에서는 2023년 기준 전년 대비 크립토재킹 공격이 399% 증가하였다고 밝혔다. 더욱이, 컨테이너 환경에서 구동되는 도커 컨테이너 이미지에 크립토재킹 악성코드가 내포되어 배포되는 경우도 다수 증가하고 있다. 국내 클라우드 전문 보안 기업인 아스트론 시큐리티에서 발행한 보고서에서도, 클라우드 환경에서 가장 빈번하게 발생하는 공격 유형이 불법적인 암호 화폐 채굴이라고 언급되었다[3].

기존의 연구에서는 CPU 및 리소스 사용량을 모니터링하거나 시스템 호출 트레이스 및 관련 시스템 로그를 분석하는 등의 기법을 통해 정상 애플리케이션과 악성 마이너를 구분하기 위한 방법론을 탐구하였다[4]. 그러나 제안된 방법론들의 경우 특정 운영 환경에서만 적용이 가능하다는 한계점을 가지거나, 환경별로 사용할 수 있는 기법들에 대한 비교 분석은 수행되지 않았다. 크립토 재킹은 시간이 지남에 따라 환경 및 대상을 바꾸어 공격 방식을 진화시켜 왔기 때문에, 다양한 컴퓨팅 환경에서 지속해서 모니터링하고 예방해야 하는 취약점으로 간주되며, 리소스의 비효율적인 사용, 재정적 손실, 서비스 성능 저하 등 다양한 피해를 예방하기 위해서는 각 환경에 맞는 적절한 모니터링 및 탐지 방법론의 적용이 필수적이다.

따라서, 본 연구에서는 각자 다른 실험 환경과 데이터 셋을 기반으로 한 기존의 크립토 마이너 탐지 기법에 대한 심층적인 비교 분석을 통해, 다변화된 컴퓨팅 환경에 따른 분류 체계를 제안하고, 적절한 방법론의 통합 및 복합적 사용에 대해서 논의하며, 각 환경에 가장 적합한 크립토재킹 탐지 방법론 수립에 기여하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 악성 크립토 마이너와 이에 대한 특징과 함께 크립토마이너 탐지 기법에 대한 연구 동향에 대해서 살펴본 후 환경별 및 발생 메트릭의 차이에 따른 방법론의 복합적 사용이 필요한 이유에 관해서 서술하였다. 3장과 4장에서는, 각 방법론을 탐지 근거와 탐지 환경에 따라서 분류하였으며 이를 통해, 각 방법론에서 사용된 크립토 마이너 탐지 기법에서 수집한 요소들을 통합하고, 구별되는 특징들을 나열하여 각 요소 및 환경에서 주안점을 두고 살펴보아야 하는 것에 관하여 서술하였다. 마지막으로, 범용 데스크톱 또는 서버 환경부터 가상 머신 기반 인스턴스, 컨테이너를 대상으로 탐지 환경별 모니터링 가능한 정보를 구분하고 분류화하여 정적 분석 및 동적 분석 방법론을 통합한 클라우드 환경에서의 복합적인 방법론 사용 프레임워크를 제시하였다.

2. 관련 연구 및 배경지식

2.1 암호 화폐 채굴

크립토 마이닝은 암호 화폐를 채굴하는 행위를 말하며 이는 검증료를 지불하거나, 새로운 코인이 주기적으로 생성될 때 해당 거래의 검증 절차를 돕는 대가로 전자화폐를 획득하는 형태로 동작한다. 구체적인 거래의 검증 절차는 블록 헤더의 Nonce 값을 변경하여 목표 값의 해시를 생성하고 새로운 블록을 형성하는 과정이다. 전자화폐, 즉 코인을 백그라운드에서 채굴하려면 전용 애플리케이션을 설치하여 수행할 수 있으며 웹 브라우저상에서도 자체적으로 크립토 마이닝을 실행할 수 있다.

이때, 보상은 가장 먼저 블록 생성을 성공한 채굴자에게 주어진다. 암호 화폐 채굴자가 증가할수록 블록 생성률이 높아지기 때문에 내부에선 적절한 난이도 조정을 통해 채굴 속도를 조절한다. 난도가 상승하면 그에 수반하는 연산량도 증가하므로 채굴자들은 빠른 연산 수행을 위해 고성능 CPU 및 GPU를 이용한다.

2.2 크립토재킹 공격 기법

앞서 언급한 암호 화폐 채굴 행위, 즉 크립토 마이닝을 악의적으로 수행하는 행위를 크립토재킹(Cryptojacking)공격이라고 하며 이는 크립토커런시(cryptocurrency)와 '납치'를 뜻하는 하이재킹(hijacking)을 합친 단어이다.[5] Cryptojacking 중 가장 큰 비중을 차지하고 있는 공격 유형은 자바스크립트로 작성된 Cryptomining 코드를 이용하는 것이다. 공격자가 특정 웹 사이트에 크립토재킹을 목적으로 하는 자바스크립트 코드를 미리 삽입한 뒤, 해당 사이트를 사용자가 방문하게 되면 브라우저의 자바스크립트 엔진에서 해당 코드를 실행하게 된다. 이를 실행하면 공격자는 사용자의 동의 없이 CPU 자원을 이용할 수 있으며, 채굴 보상은 공격자에게 지급된다.[6] 일차적으로 사용자가 모르게 채굴 과정을 반복하기 때문에 자원이 낭비되는 문제점이 발생하며, 사용자는 자신이 하지 않은 행위에 대한 비용을 지불해야 할 수도 있다. 결국, 지속적인 작업으로 인해 사용자 기기 자체의 수명 또한 단축될 수 있다.

초기의 크립토재킹 공격은 주로 악성 링크 또는 악성 첨부 파일을 포함한 이메일을 전송하여 이를 클릭하면 악성코드가 실행되도록 설계되었다. 이후에는 불특정 다수의 사용자를 대상으로 웹 페이지에 악성 스크립트를 삽입하여 공격을 확산시켰다. 이 중 가장 잘 알려진 악성코드는 암호 화폐 모네로를 채굴하기 위한 "32Kilences.exe"이다. 초기에 개발된 크립토재킹 악성코드는 사용자의 CPU 자원을 완전히 독점적으로 활용한다는 특징을 가졌으며 이에 따라 사용자는 비교적 쉽게 감염 여부를 확인할 수 있었다. 그러나 공격자들은 이를 개선하여 사용자가 쉽게 알아채지 못하도록 CPU 자원 사용률이나, 실행 시간을 유동적으로 조절하는 경우가 빈번해졌다. 그 예로 사용자가 기기를 사용 중일 때에는 눈치채지 못하도록 미세한 자원만을 소모하도록 설계하거나, 사용자가 감지하기 어려운 시간대에 자원을 많이 소모하는 방식 등이 있다.[5] 이로 인해 단순 자원 사용량을 모니터링 하는 방식이 아닌, 복합적인 모니터링에 기반을 둔 크립토 마이너 판별법이 필요하다.

기존의 크립토 마이너 탐지 방식에 대해서 복합적으로 살펴본 연구는 최원석 외 2인이 크립토 재킹 연구 동향에 대해서 살펴본 연구가 있으며 크립토 재킹에 대한 난독화 기법과 웹브라우저를 이용하는 크립토 재킹에 대해서 분석하였다.[7]
다만, 최근에 널리 사용되고 있는 클라우드 환경에서 시행되는 크립토마이닝에 대한 분석은 이루어지지 않았고, 환경별 공격기법에 대한 차이점 및 발생 메트릭의 차이점은 분석되지 않았다. 본 논문에서는 각 환경별 차이점을 고려하여 크립토 재킹 공격에 대한 분석을 시행하고, 통합적이고 복합적인 방법론 사용에 대해서 논의하고자 한다.

3. 방법론 Ⅰ: 탐지 근거를 중심으로

본 장에서는 악성 크립토 마이너 탐지를 위한 기존의 탐지 방법론을 종합적으로 살펴봄과 동시에 마이너 탐지 근거를 기준으로 시스템 자원 사용량 기반탐지, 네트워크 레벨 블랙리스트 기반 탐지, 시스템 로그 추출 기반 탐지 기법으로 세분화하여 각 방법론의

요구사항 및 장단점을 비교 분석한다. 선행연구를 통해 알려진 악성 크립토재킹 탐지를 위해 활용할 수 있는 시스템 로그 및 이벤트, 네트워크 레벨 정보 등의 원시 데이터 및 메트릭을 식별하고, 각 탐지 근거를 활용했을 시의 실효성에 대해서 고찰한다.

3.1 시스템 자원 사용량 기반 탐지

악성 크립토 마이너는 기기 CPU의 연산 처리 능력과 전기를 사용자의 동의 없이 사용한다. 마이너가 채굴용 악성코드나 스크립트를 사용하면 공격자 프로그램들이 피해 기기 백그라운드에서 자동으로 실행되게 되면서 자원 사용량이 늘어나게 된다. 따라서 자원소모량의 변화를 모니터링 하는 방식으로 크립토 마이너를 탐지할 수 있다. 자원소모량의 증가는 피해 기기의 성능 저하, 전기 요금 상승 등의 결과를 불러온다. 이러한 마이닝이 클라우드 환경 및 기업과 같은 대규모 컴퓨팅 자원을 사용하는 환경에서 시행되면 피해 규모는 더욱 커지고 막대한 복구비용이 발생할 수 있다.

시스템 자원 사용량 기반 탐지 요소로는 CPU 트래픽, CPU 시간당 사용률이 있다. 고동현 외 3인은 Headless Chrome를 이용해 특정 시간 동안 일정한 주기로 브라우저의 CPU 점유율을 확인하고 그 값이 threshold를 초과한 횟수가 90% 이상일 경우 Cryptojacking이 수행되고 있을 가능성이 있다고 판단하였다.[6] 김이수 외 2인은 모의 크립토재킹 악성 컨테이너를 제작하여 악성 컨테이너의 60초간 자원 사용량을 측정하였으며 각 컨테이너의 CPU 자원 사용률을 유저 시간과 시스템 시간으로 나누어 분석하였다.[8]

<Table 1> System-based detection

Detection factors	Criteria
Browser CPU usage	Detected when the number of times the lowest threshold is exceeded is 90%
CPU usage of mining container	Total CPU usage : running a mining container averages 55.58 seconds.

선행연구에서 제안된 시스템 기반 탐지 요소 및 기준을 요약하면 <Table 1>과 같다. 다만 이러한 시스템 자원 사용량을 단독 근거로 하여 마이너를 탐지하게 될 경우, CPU 점유율이 높은 정상 프로그램들에 대한 오탐 가능성이 증가하기에, 대부분은 다른 탐지 근거들과 함께 사용한다. 실제로, 선행연구에서는 CPU 점유율 및 사용량과 함께 웹 소켓 연결 여부, 반복문 시행 여부를 추가로 확인하였다 [6].

3.2 네트워크 레벨 블랙리스트 기반 탐지

마이너가 자바스크립트를 사용할 때는 사용자가 특정 URL이나 브라우저에 접속하는 경우가 대다수이다. 이를 이용해 스크립트 내에 존재하는 특정 키워드 또는 해당 URL 자체를 탐지하여 차단하는 기법을 사용할 수 있다. 또한, 마이닝 된 암호화폐를 전자지갑으로 전송하는 과정이 반복적으로 필요하므로 코드 내 특정 IP로 트래픽을 전송하는 반복문이 존재하는지 살펴볼 수 있다.[6]

키워드 기반 탐지 요소로는 이미 크립토재킹을 시행한 이력이 있는 IP 주소, URL, 헤더/페이로드 정보, 웹 소켓 연결 여부 등을 살펴볼 수 있다. 임은지 외 2인은 크립토재킹 스크립트에서 공통으로 발견되는 코드를 필터링 블랙리스트에 추가하여 접근을 방지하는 기법을 적용하였다. 예를 들어 'miner','anonymous', 'coinhive' 등 크립토재킹 스크립트에 자주 등장하는 단어의 빈도를 측정하여 악성 스크립트 여부를 판단하였다. [9]

선행연구에서 제안된 방법론을 기준으로 살펴본 탐지 요소와 탐지 기준에 대한 예시는 <Table 2>와 같다 해당 방법 또한 시스템 자원 사용량 기반 탐지와 마찬가지로, 단독 기준을 가지고 마이너를 판단하게 되면 Black list 기반으로 작동하기 때문에 업데이트가 되지 못한 주소에 대한 미탐 가능성이 존재하며, 공격자가 URL 또는 IP 주소를 우회하거나 변경했을 경우에도 미탐 가능성이 높아지며, 스크립트에 난독화 기법을 적용할 시 오탐 및 미탐의 가능성이 증가한다.[7]

<Table 2> Network-level blacklist-based detection

Detection factors	Criteria	Features
URL	Including blacklist	dark website
script	Including specific keywords	key word : 'miner', 'anonymous', 'coinhive' etc.
	Loop statement	Using the Browser's JavaScript engine itself
IP address	Including blacklist	DPI, DNS traffic, etc.
websocket	including websocket connection	Including websocket address in the blacklist

3.3 시스템 로그 추출 기반 탐지

채굴을 시작하게 되면, 채굴된 암호화폐를 마이너의 전자지갑으로 전송하는 과정이 필요하다. 이 과정에서 응용 프로그램의 요청에 따라 커널에 접근하기 위해 system call을 호출하는 과정이 필요하다. 따라서 해당 시스템 로그를 분석한다면 마이너의 활동을 탐지 및 추적할 수 있다. 시스템 로그 추출 기반 탐지 요소로는 system call, API 시퀀스 등이 있다. 송지현 외 4인은 정상 컨테이너와 크립토재킹 컨테이너에서의 system call을 비교하고 그 결과, 크립토재킹 컨테이너의 경우 소켓과 관련된 함수가 상위에 위치한다는 것을 알아냈다. 또한 해당 컨테이너에서 소켓 연결, 메시지 송수신, 프로세스 관리 함수 등 시스템 로그 탐지 요소들을 확인할 수 있었다.[10]

김이수 외 2인은 리눅스의 시스템 트레이스 도구인 Extended Berkley Packet Filter(eBPF)를 사용하여 악성 컨테이너의 가상화폐 채굴 프로세스의 시스템 콜 60초간 관측하였고, sys_sched_yield, sys_epoll_wait과 같은 CPU 자원과 관련된 시스템 함수가 자주 호출되는 것을 확인하였다.[8]

앞선 두 방법론을 종합해 보았을 때, cpu 및 소켓 통신 관련 함수들이 다른 프로세스에 비해서 자주 호출 되는 경우 마이닝 행위를 의심할 수 있다.

<Table 3> System log-based detection

Detection factors	Detection tool	Features
System call	Linux perf	futex,write,lseek,recvmsg, epoll_wait ...
	eBPF	sys_sched_yield, sys_epoll_wait ...

앞서 언급한 방법론에서 살펴본 탐지 요소는 <Table 3>와 같다. 다만, 이렇게 VM이나 컨테이너 환경에서 수집되는 마이너 정보들의 경우 컨테이너의 개수 또는 VM의 노드 개수가 늘어나고 규모가 확장됨에 따라서 가시적으로 정보를 확인하기 어려운 경우가 많으며, 각기 다른 역할을 하는 컨테이너들이 늘어나고 세분화되면서, 위와 같은 함수호출 빈도수에 따라 마이너를 구분하기는 현실적으로 어렵다는 한계점이 있다. 또한 이러한 System call 이나 API Sequence 들이 수집되는 위치에 따라서 여러 부가 정보가 추가로 수집되기 때문에 각 정보가 노이즈가 되어 마이너 탐지 정확도를 저하시키는 경우가 많다. 따라서 악성 마이너에 대한 확실한 근거 또는 행위가 직접적으로 시작되는 가상 머신 또는 컨테이너 인스턴스를 식별하고 특정하는 기술을 통해 탐지 정확도를 향상해야 한다.

4. 방법론 II: 탐지 환경을 중심으로

높아진 암호 화폐 채굴 난도로 인해 공격자들은 클라우드 환경을 포함하여 고성능 컴퓨팅 자원을 활용할 수 있는 환경이라면 가리지 않고 크립토재킹 공격을 수행하는 상황에 이르렀다. 이와 같은

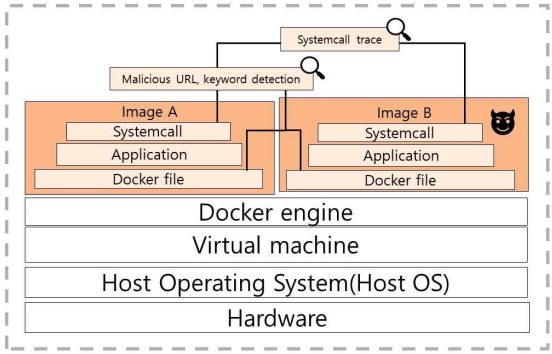
다양한 채굴 환경으로 인해, 각 환경에 적합한 마이너 탐지 방법론을 적용해야 한다. 따라서 본 장에서는 크립토 마이너 탐지가 필요한 잠재적 공격 대상 환경을 3가지 형태로 분류하고, 환경별로 어떠한 특성과 고려 사항이 존재하는지 탐지 근거를 마이닝 시행 환경 관점에서 분석하고자 한다.

4.1 컨테이너 기반 클라우드 네이티브 환경

컨테이너 환경에서 악성 이미지를 탐지하기 위해서는 마이크로서비스를 구동하는 형태의 클라우드 네이티브 환경의 특성을 고려해야 한다. 선행연구에서는 아래와 같은 환경에서 탐지 및 분석을 시도하였다. 송지현 외 4인의 경우 Ubuntu Virtual Box 환경에서 Docker를 설치하고 컨테이너를 구성하여, 정상 컨테이너와 크립토재킹 컨테이너의 System call 시퀀스를 비교하여 악성 컨테이너와 정상 컨테이너의 차이점을 구별하고자 하였다.[10] 또한 CPU 사용 시간을 스케줄러에 반납하는 시스템 콜인 sys_sched_yield를 관측하여 악성 컨테이너의 시스템 자원 사용량을 탐지하는 동적 분석 기법도 존재한다. 동적 분석 방법론은 취약점 분석 대상이 실행될 수 있는 실험 환경을 직접 구성하고 환경 내에서 악성이나 정상 워크로드를 실제로 인스턴스화 하여 관련 메트릭을 수집한 뒤 결과를 분석하는 방식이다.

이처럼 악성 크립토마이너를 탐지하기 위해 동적 분석을 통해 수집한 데이터를 바탕으로 기계학습을 이용하는 경우가 다수이며, 이 과정에서 데이터 전처리 과정과 분류 과정이 필요하다. 각 마이너가 동작할 때 발생하는 로그 데이터를 수집하여 분석에 이용하기 때문에 공격자의 행위 추적에 용이하다는 장점이 있다.

한편, 분석 대상의 소스 코드를 중심으로 코드 스크립트 속 취약점을 미리 탐지하거나 및 배포 환경 관련 설정을 프로파일링 하여 크립토 마이닝 여부를 탐지하는 방식인, 정적 분석 방법론을 이용하면 컨테이너 생성 이전 이미지에 대한 도커파일이나 악성 API 코드에 대한 블랙리스트 기반 보안 감사를 수행할 수 있다.



(Figure 1) Diagram of Container-Based Environment

<Table4> Detection Based on Container Layer

Detection environment (layer)	Data collection tools	Detection factors data
Container	perf	Systemcall

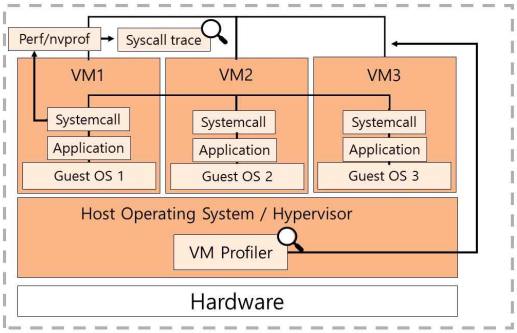
후술할 가상 머신 기반의 환경과 달리 컨테이너 기반의 네이티브 환경은 Repository를 통해 사용자가 도커 이미지 파일을 확인하고 악성 이미지 여부를 미리 탐지할 수 있다. 따라서 정적 분석 방법론을 적용하여 악성 이미지 생성 이전에 마이너 탐지가 가능하다.

그러나 이미지 실행 이후에는 프로세스가 컨테이너 내부에서 가상화되어 실행되기에 특정 프로세스의 악성 행위를 탐지하는 것이 복잡해진다는 단점이 있다.[11] 또한, 악성 컨테이너 이미지 탐지를 우회하기 위해 악성 컨테이너들은 자원 사용량을 스스로 제어 및 제한하는 시도를 하기도 하여, 단순히 자원 사용량으로 판단하는 것이 아닌 정상 컨테이너와 악성 컨테이너의 기술적 특성 및 통신 패턴을 비교, 분석해야 한다.

4.2 가상 머신 기반 클라우드 인프라 환경

일반적인 IaaS(Infrastructure as a Service) 형태로 운영되는 퍼블릭 클라우드 인프라 환경 내 독립적인 게스트 운영체제에서 마이너를 탐지하는 경우 가상 머신 기반의 클라우드 환경을 사용한다, Saide Manuel Said 외 2인은 Ubuntu 18.04 LTS와 함께 VM ware

Workstation 16 Pro 가상 머신(VM)을 설치하고 Docker 20.10.7 버전을 사용하여 Docker image와 함께 가상 머신을 사용하여 마이닝을 시행했다. 이후 데이터를 수집한 뒤, 머신러닝을 이용하여 수집된 결과들을 비교 및 분석했다.[12] Rashid Tahir 외 3인의 경우, HPC(Hardware Performance Counter) 기반의 서명을 사용해 실시간으로 가상 머신을 모니터링하며 마이너를 감지하는 자체 VMprofiler를 개발하였으며, 시스템 로그 등 데이터 수집에는 Perf와 nvprof를 함께 사용하였다.[13]



(Figure 2) Diagram of Virtual Machine-Based Environment

<Table 5> Detection Based on Virtual Machine Layer

Detection environment (layer)	Data collection tools	Detection factors data
Virtual Machine	perf/nvprof	Systemcall
		variation of HPC signature values

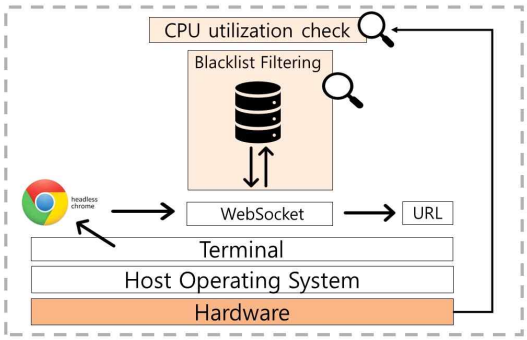
앞선 방법론을 통해 살펴본 가상 머신 환경에서 마이닝을 시도하고 이를 통해 데이터를 수집하는 경우는 (Figure 2)과 같은 방법론을 따른다. 동작 과정은 ① 가상 머신들이 시행이 되면 HPC를 통해 실행 중인 프로그램의 런타임 동작과 특성을 기록한 뒤 ② 이를 하이퍼 바이저 또는 운영체제 위에서 작동하는 VMprofiler를 이용해 분석하고 HPC에 기록된 값을 비교하여 ③ 마이너로 의심이 가는 가상 머신의 행동을 제어하는 단계로 구성되어 있다.[13]

그러나 가상 머신 기반의 인프라 환경은 게스트 운영체제에서 사용된 CPU의 자원 사용량과 실제 호스트 환경에서 모니터링 하는 자원 사용량이 정확히 일치한다고 보기 힘들다. 따라서 각각의 가상 머신 내에서 자원 사용량 분석이 이루어져야만 정확한 결과를 얻을 수 있는데, 이는 분석하고자 하는 가상 머신의 개수가 많아지는 경우 비효율적이고, 관리가 어렵고 종합적으로 판단하기에 어렵다는 단점이 있다.

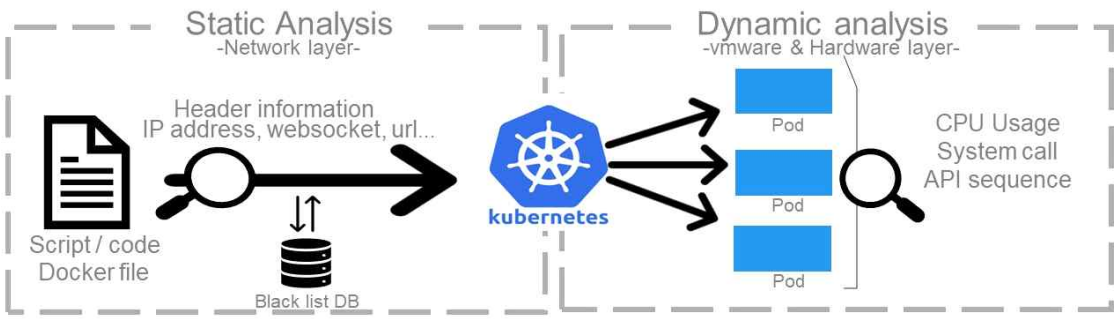
또한, 가상 머신 내부에서 크립토재킹을 시행하는 공격자가 탐지 프로그램을 인식하게 되면, 정상 컨테이너로 위장하기 위해 마이닝을 정지하는 문제점이 있을 수 있는데, 해당 방법론은 이를 방지해 주고, 실시간으로 VM에서 활동하는 마이너의 활동을 감시할 수 있다는 장점이 있다.

4.3 일반적인 서버/데스크톱 호스트 환경

가상화 환경이 아닌 범용 서버 및 데스크톱 환경에서 크립토재킹을 탐지하기 위한 연구 또한 수행되었다. 고동현 외 3인의 경우, 코인 하이브를 통한 일반 Ubuntu 16.04 호스트 환경에서 Headless 브라우저를 이용하여 탐지 대상 사이트의 공격 여부를 확인하는 동적 Cryptojacking 사이트 탐지 방안을 제안하였다.[5]



(Figure 3) Diagram of Server/Desktop Host Environment-Based Configuration



(Figure 4) Comprehensive analysis methodology

<Table 6> Detection Based on Network and Hardware Layer

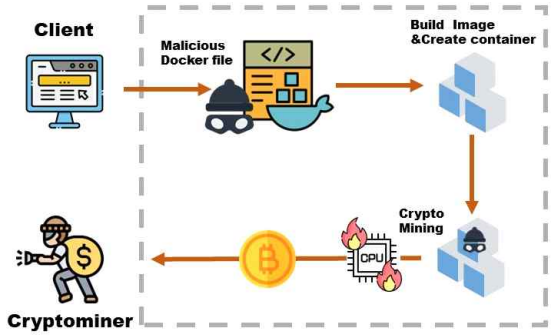
Detection environment (layer)	Execution tools	Detection factors data
Network level	mining site	URL, IP adress
	Blacklist using DB : stored IP addresses	
Hardware level	top(Linux CPU usage measuring tool)	CPU usage

앞서 살펴본 방법론에서는 Chrome의 headless 브라우저를 이용하는 방법을 사용하였으며, 자세한 과정은 (Figure 3)와 같다. ① 터미널에서 Chrome의 headless 브라우저로 접속한다. ② 이후 Websoket을 생성하고 연결하는 과정에서 블랙리스트 필터링 기법을 적용하여 마이닝이 이루어지고 있는 브라우저의 접속을 차단할 수 있다. ③ 만약 블랙리스트에 없다면 URL에 정상적으로 연결한 뒤에 ④ CPU 사용률을 주기적으로 확인하여 의심되는 사이트 및 감염사이트의 주소나 소켓 주소를 블랙리스트 데이터베이스에 추가해 의심스러운 사용자의 접속을 차단할 수 있다.[6]

이처럼, 제안된 탐지 방법론은 정적 분석 방법론과 동적 분석 방법론을 적절히 섞어 기존의 공격과 함께, 알려지지 않은 공격 또한 대비할 수 있다는 장점이 있다. 다만, 단순히 CPU 사용률이 높다고 해서 해당 프로그램이 마이닝과 관련이 있는지 여부가 불확실하고, 오탐의 가능성이 높기 때문에, 논문에서 제시한 것과 같이 오탐을 줄이기 위한 정적 분석 방법 및 동적 분석 방법론을 적절히 혼합하여 사용해야 한다.

5. 논의 및 고찰

본 고에서 살펴본 탐지 근거 및 세부 탐지 요소에 대한 요약은 <Table 7>과 같으며, 이를 종합하여 본 논문에서 제안하는 프레임워크는 (Figure 4)와 같다. 네트워크 레벨에서 먼저 시행할 수 있는 정적 분석 기법을 이용하여 Header 정보, URL, 스크립트 내 의심스러운 정보들에 대해서 판별한 후, 실제로 컨테이너가 구성된 이후에는 동적 분석 기법을 이용해 자원사용률과 System Call 호출 시퀀스를 근거로 하여 마이너가 활동 중인지 판단 할 수 있다. 이와 같은 방법이 적용 될 수 있는 환경은 도커 파일 및 쿠버네티스를 기반으로 다양한 컨테이너가 구동되는 (Figure 5)와 같은 시나리오에서 적용 될 수 있다.



(Figure 5) Malicious Docker File-Based Crypto Jacking Mechanism

<Table 7> Summary Table

Detection factors	Features	Description
System-based : CPU power consumption	CPU usage	Increased possibility of false positives when judging usage alone on a single basis
Network-level : white list, blacklist	IP address, header/payload information, malicious URL string	Fewer false positives for known addresses
		Attack judgment of difficult new techniques
System log	System call, API call sequence	Easy attacker path tracking
		Data preprocessing process required for analysis

시스템 기반의 CPU 사용률을 근거로 마이너를 탐지하는 방법론의 경우에는 실시간으로 마이너를 관찰할 수 있다는 장점이 있다. 그러나 단독 기준으로 마이너를 판단할 경우, CPU 자원을 많이 사용하는 일반 프로그램을 마이너로 오인할 수 있다. 또한 독립적인 게스트 운영체제에서는 가상화 레이어로 인한 시스템 오버헤드로 인해, 프로세스 단위로 사용량 변화를 관찰할 수 있는 컨테이너 기반과는 달리 자원 사용량 측면에서의 탐지가 어렵다. [14] 따라서 단독으로 사용하기보다는 다른 방법론과 함께 부수적인 탐지 판단 기준으로 사용하는 것이 이러한 단점을 보완할 수 있다. 본고에서는 이를 적용하여 클라우드 환경에서 적용할 수 있는, 정적분석 및 동적 분석 기법을 함께 사용하는 프레임워크를 제안하였다.

네트워크 레벨에서 블랙리스트 기법을 이용하여 마이너를 탐지하는 경우, 기존에 알려진 주소들을 탐지하는 데에는 오탐 가능성이 거의 없지만 알려지지 않은 주소는 탐지가 어렵고, 우회 또는 변경 및 코드 난독화에 취약하다. 이 또한 여러 탐지 근거와 복합적으로 사용하는 것이 오탐 가능성을 줄이는 것에 도움이 된다.

가상 머신 기반의 인프라 환경에서 마이닝 탐지가 이루어지는 경우, 게스트 운영체제에서 사용된 CPU의 자원 사용량과 실제 호스트 환경에서 모니터링 하는 자원 사용량이 정확히 일치한다고 보기 힘들기 때문에 네트워크 레벨 탐지기법과 동적 탐지기법이 혼합적으로 사용되어야 한다. 가상 머신은 다른 게스트 운영체제 혹은 타 소프트웨어 스택으로 인한 오버헤드 역시 실제 호스트 환경 안에서 발생하기 때문에 자원 사용량을 측정한다 해도, 부정확한 결과가 나올 가능성이 크다는 단점이 있다. 따라서 각각의 가상 머신 내에서 자원 사용량 분석이 이루어져야만 정확한 결과를 얻을 수 있는데, 이는 분석하고자 하는 가상 머신의 개수가 많아지는 경우 비효율적이고, 관리가 어렵고 종합적으로 판단하기에 어렵다는 단점이 있다.

시스템 로그를 기준으로 판단하는 경우, 실제 마이너가 각각의 컨테이너 또는 VM 안에서 이용한 시스템들에 대한 정보를 담고 있기에 마이너의 행위 추적에 용이하다는 장점이 있다. 다만, 정보 수집 Layer 에 따라서 의미 없는 정보들이 추가로 수집되면서, 노이즈가 발생 할 수 있으며 컨테이너 및 VM의 개수가 늘어나게 되면 로그들을 일일이 분석하기 어렵다는 한계점이 있다.[14]

6. 결론 및 추후 연구 과제

가상화 시스템 및 클라우드 시스템의 발전과 함께 크립토재킹을 이용한 암호 화폐 탈취 공격양상도 일반 데스크탑 환경에서 클라우드와 같은 대용량 컴퓨팅 자원을 사용하는 환경으로 변화하였다. 이로 인해 컨테이너와 같이 호스트 OS를 공유하는 가상화 환경이나 독립적으로 운영되는 가상 머신을 이용하는 경우 등

공격이 발생할 수 있는 상황과 환경도 다양해졌다. 이러한 환경별로 적용 가능한 크립토 마이너 탐지기법에 대한 연구가 필요한 시점이다. 따라서 본 연구에서는 기존의 크립토마이닝 탐지 연구에서 제시된 방법론에 대해 복합적으로 살펴보고, 탐지근거에 따른 분류 기준을 시스템 자원 사용량 기반 탐지, 네트워크레벨 블랙리스트 기반탐지, 시스템 로그 추출 기반 탐지로 식별하고, 각 근거에 따른 탐지 메트릭을 확인하였다.

탐지 환경에 따른 분류기준의 경우, 컨테이너기반 클라우드 네이티브 환경, 가상머신 기반 클라우드 인프라 환경, 일반적인 서버/데스크톱 환경 총 3가지 환경으로 나누어 각 환경별 탐지 근거가 수집되는 레이어를 분류 및 분석하였다. 또한 각 레이어별로 크립토 마이너 탐지 기법을 적용하였을 때의 장점, 단점에 대해서 분석하였다. 이와 같은 종합적인 비교분석을 바탕으로 클라우드 환경에서 적용 가능한 크립토 마이너 악성코드에 대한 정적분석 기법과 동적 분석 기법을 통합적으로 사용하는 프레임워크를 제시하였다.

추후 연구과제로서는, 클라우드 네이티브 환경에서 필수적으로 사용하는 컨테이너에서 호스트 운영체제를 공유함으로서 발생할 수 있는 격리 및 보안상의 이슈에 대해서 살펴보고, 악성 마이너에 대한 확실한 탐지 근거를 바탕으로 크립토재킹 행위의 직접적인 원인이 되는 가상머신 또는 컨테이너 인스턴스를 식별하는 프레임워크를 구상한다. 또한 실제 실험을 통해 제안한 프레임워크의 탐지 정확도를 높이고 탐지 과정 및 결과의 가시성을 향상시키는 것을 목표로 한다.

참고문헌

[1] Microsoft. (2023). Microsoft Threat Intelligence. https://www.microsoft.com/en-us/security/business/siem-and-xdr/microsoft-defender-threat-intelligence.

[2] Son, J (2018, February 25) . (2018, Fe 12). "'Malicious Cryptojacking'... Trials of Hacking with the Rise in Digital Currency Values." Maeil Business Newspaper, Retrieved from: https://www.mk.co.kr/news/it/8206706.

[3] Kim, K. (2022, November 15). "Astron Security: 'Cryptojacking Tops the List of Cyber Attacks Targeting Clouds... Cryptocurrency Mining'." Boan News. Retrieved from https://www.boannews.com/media/view.asp?idx=111330.

[4] Karn, Rupesh Raj, Kudva, Prabhakar, Huang, Hai, Suneja, Sahil, & Elfadel, Ibrahim (Abe) M. (2021, March). "Cryptomining Detection in Container Clouds Using System Calls and Explainable Machine Learning." IEEE Transactions on Parallel and Distributed Systems, 32(3).

[5] IGLOO. (2020, May 6). The Ever-Present Threat: Cryptojacking. Retrieved from: https://www.igloo.co.kr/security-information/%EC%82%AC%EB%9D%BC%EC%A7%80%EC%A7%80-%EC%95%8A%EC%9D%80-%EC%9C%84%ED%98%91-%ED%81%AC%EB%A6%BD%ED%86%A0%EC%9E%AC%ED%82%B9/.

[6] Goh, D., Jung, I., Choi, S. H., & Choi, Y. H. (2018). Dynamic Analysis Framework for Cryptojacking Site Detection. Journal of the Korea Institute of Information Security and Cryptology, 28(4), 963-974.

[7] Choi, W.-S., Kim, H.-S., & Lee, D.-H. (2018). Trends in Cryptojacking Research. Journal of Information Security, 28(3). 33-36.

[8] Kim, I., Yoo, S., & Nam, J. (2023). Workload Analysis of CryptoJacking Containers. In Proceedings of the 2023 Korean Computer Conference, pp.1,911 - 1,913.

[9] Lim, E. J., Lee, E. Y., & Lee, I. G. (2021). Behavior and Script Similarity-Based Cryptojacking Detection Framework Using Machine Learning. Journal of the Korea Institute of Information Security and Cryptology, 31(6), 1105-1114.

[10] Song, J. H., Park, K. M., Park, C. H., Kim, J. H., & Kim, I. K. (2023). Performance Analysis of Cryptojacking Container Detection with Machine Learning. Proceedings of the Korea Information Science Society Conference, 1294-1296.

[11] J. Burgess, D. Carlin, P. O'Kane, and S. Seze

r, "MANiC: Multi-step Assessment for Crypto-miners," 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, 2019, pp. 1-8, doi:10.1109/CyberSecPODS.2019.8885503.

[12] Saide, M., Sarmento, E. L. A., & Ali, F. D. M. A. (2022). Cryptojacking Malware Detection in Docker Images Using Supervised Machine Learning. In Proceedings of the 2022 International Conference on Intelligent and Innovative Computing Applications. https://doi.org/10.59200/ICONIC.2022.006

[13] Tahir, R., Huzaifa, M., Das, A., Ahmad, M., Gunter, C., Zaffar, F., Caesar, M., & Borisov, N. (2017). "Mining on Someone Else's Dime: Mitigating Covert Mining Operations in Clouds and Enterprises." In Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses (RAID), Lecture Notes in Computer Science.

[14] S. Sultan, I. Ahmad and T. Dimitriou, "Container Security: Issues, Challenges, and the Road Ahead," inIEEE Access, vol. 7, pp. 52976-52996, 2019, doi: 10.1109/ACCESS.2019.2911732

〔저자소개〕

유 지 원 (Ji-won Yoo)
2024년 2월 성신여자대학교 글로벌비즈니스 학과/융합보안공학과 학사
2024년 3월~현재 성신여자대학교 석사 재학 중
email : 220246055@sungshin.ac.kr

강 서 연 (Seo-yeon Kang)
2021년 3월~현재 성신여자대학교 융합보안공학과 공학사 재학 중
email : 20211036@sungshin.ac.kr

이 수 미 (Su-mi Lee)
2021년 3월~현재 성신여자대학교 융합보안공학과 공학사 재학 중
email : dltnal339@gmail.com

김성민(Seongmin Kim)
2012년 2월 : 한국과학기술원 전기 및 전자공학과 졸업
2014년 2월 : 한국과학기술원 전기 및 전자공학과 석사
2019년 2월 : 한국과학기술원 정보보호대학원 박사
2019년 9월 ~ 2020년 8월 : 삼성전자 삼성리서치 Staff Engineer
2020년 9월 ~ 현재 : 성신여자대학교 융합보안공학과 조교수
※관심분야 : 신뢰 실행 환경, 클라우드 컴퓨팅, 시스템 보안