Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.4, Aug. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.4.963

크립토재킹 사이트 탐지를 위한 동적 분석 프레임워크*

고 동 현,† 정 인 혁, 최 석 환, 최 윤 호‡
부산대학교

Dynamic Analysis Framework for Cryptojacking Site Detection*

DongHyun Ko,† InHyuk Jung, Seok-Hwan Choi, Yoon-Ho Choi‡
Pusan National University

요 약

비트코인과 같은 암호 화폐에 대한 관심이 증대됨에 따라 블록체인 기술은 뛰어난 보안성을 갖춘 분산 원장 플랫폼으로 다양한 응용분야에서 많은 주목을 받고 있다. 그러나 암호 화폐 채굴(Cryptomining) 과정에 대한 취약성으로 인해 타인에게 CPU와 같은 컴퓨터 자원을 몰래 갈취하는 공격인 Cryptojacking이 등장하였다. 그 중에서도 브라우저 기반 Cryptojacking은 사용자의 PC에 설치하는 동작 없이 단순히 웹 사이트를 방문하는 것만으로 공격이 수행된다는 점에서 그 심각성이 증대되고 있다. 현재까지의 Cryptojacking 탐지 시스템은 대부분 시그니처 기반으로 동작하기 때문에, 기존 Cryptomining 코드의 변형이나 새롭게 등장하는 Cryptomining 코드는 탐지하지 못하는 문제점이 존재한다. 이를 극복하기 위하여, 본 논문에서는 알려지지 않은 Cryptojacking 공격에 대한 탐지를 위해 Headless 브라우저를 이용하여 탐지대상 사이트의 공격 여부를 확인하는 동적 Cryptojacking 사이트 탐지 방안을 제안한다. 제안하는 동적 분석 기반 Cryptojacking 탐지 시스템은 기존 시그니처 기반 Cryptojacking 탐지 시스템에서 탐지하지 못하는 새로운 Cryptojacking 사이트를 탐지 할 수 있으며, Cryptomining 코드를 우회하여 호출하거나 난독화하더라도 이를 탐지하는 것이 가능하다.

ABSTRACT

With the growing interest in cryptocurrency such as bitcoin, the blockchain technology has attracted much attention in various applications as a distributed security platform with excellent security. However, Cryptojacking, an attack that hijack other computer resources such as CPUs, has occured due to vulnerability to the Cryptomining process. In particular, browser-based Cryptojacking is considered serious because attacks can occur only by visiting a Web site without installing it on a visitor's PC. The current Cryptojacking detection system is mostly signature-based. Signature-based detection methods have problems in that they can not detect a new Cryptomining code or a modification of existing Cryptomining code. In this paper, we propose a Cryptojacking detection solution using a dynamic analysis-based that uses a headless browser to detect unknown Cryptojacking attacks. The proposed dynamic analysis-based Cryptojacking detection system can detect new Cryptojacking site that cannot be detected in existing signature-based Cryptojacking detection system and can detect it even if it is called or obfuscated by bypassing Cryptomining code.

Keywords: Blockchain, Cryptojacking, headless browser, dynamic analysis-based

Received(07. 05. 2018), Modified(08. 03. 2018), Accepted(08. 06. 2018)
* 본 연구는 한국연구재단 논문연구과제 (NRF-2018R1D1A3B07043392) 지원으로 수행되었습니다.
† 주저자, uyt1209@pusan.ac.kr
‡ 교신저자, yhchoi@pusan.ac.kr(Corresponding author)

I. 서 론

4차 산업혁명의 도래와 더불어 블록체인 기술은 뛰어난 보안성을 갖춘 플랫폼으로 많은 주목을 받고 있다. 이를 활용한 대표적인 사례인 암호 화폐(Cryptocurrency)는 거래의 무결성 보장 및 낮은 유지보수 비용으로 지폐를 대체할 수단으로도 자주 언급되고 있다[1].

하지만, 블록체인 기반의 암호 화폐의 강력한 보안성에도 불구하고 암호 화폐 채굴(Cryptomining) 과정에 대한 취약성은 여전히 존재한다. 암호 화폐 보안업체인 시만텍에 따르면, 암호 화폐 채굴 관련 악성코드는 2017년 1월 2만건에서 12월 170만건으로 8500%증가하였다. 그 중에서도, 브라우저 기반 Cryptomining을 위해 필요한 자원을 타인에게서 몰래 갈취하는 공격인 Cryptojacking은 전년 대비 34000% 급증하였다[2]. 이러한 지표는 Cryptojacking의 심각성을 시사한다.

현재까지의 Cryptojacking 탐지 시스템은 대부분 시그니처 기반으로 동작한다[3][4][5][6]. 즉, 기존의 탐지 시스템은 사용자가 Cryptomining 코드가 심어져 있는 사이트를 방문하게 될 경우 해당 사이트가 기존에 알려진 Cryptomining 코드 제공 업체의 주소로 연결을 요청하지 못하도록 차단하거나 사용자가 입력한 사이트의 HTML 문서에서 알려진 Cryptomining 코드 제공 업체로 요청하는 부분이 있을 경우 사용자에게 알려주는 방식을 통해 Cryptojacking을 탐지한다. 이러한 시그니처 기반 탐지 방법은 탐지 속도가 빠르다는 장점이 있지만, 기존 Cryptomining 코드의 변형이나 새롭게 등장하는 Cryptomining 코드는 탐지하지 못하는 문제점이 존재한다[7].

따라서, 본 논문에서는 시그니처 기반의 탐지방법뿐만 아니라 알려지지 않은 Cryptomining 코드에 대한 탐지를 위해 터미널 환경에서 GUI 없이 브라우저를 실행할 수 있는 툴인 Headless 브라우저[8]를 이용하여 탐지대상 사이트의 공격 여부를 확인하는 동적 분석 방식을 추가한 Cryptojacking 탐지 방안을 제안한다.

제안하는 Cryptojacking 탐지 시스템은 기존의 Cryptojacking 탐지 방법과 다음과 같은 차이점이 존재한다. 첫째, 제안하는 탐지 시스템은 동적 분석 방식을 이용하기 때문에 기존의 잘 알려진 Cryptojacking 사이트뿐 아니라 새로운 Cryptojacking 사이트를 탐지할 수 있다. 둘째, 제안하는 탐지 시스템은 Cryptomining 코드의 동작 패턴을 분석하기 때문에 Cryptomining 코드를 우회하여 호출하거나 Cryptomining 코드를 난독화하더라도 효율적으로 탐지가 가능하다. 마지막으로, 하나의 플랫폼만 제공해주는 기존의 탐지 방법과 달리 크롬 확장 프로그램과 웹 사이트의 두 가지 플랫폼으로 제공하기 때문에 사용자 접근성이 높은 장점이 있다.

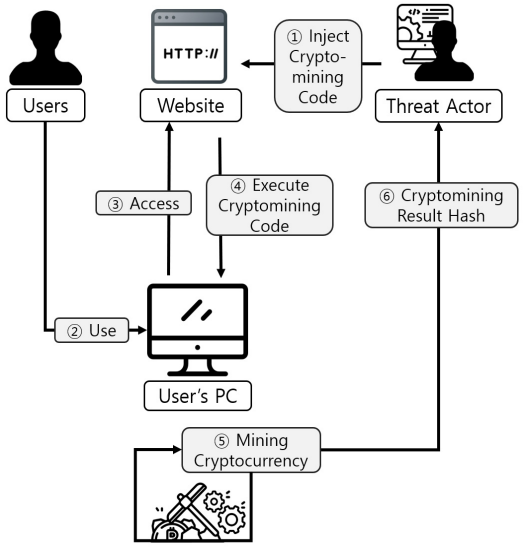

Fig. 1. Operational flow diagram of web browser-based cryptojacking

본 논문의 구성을 요약하면 다음과 같다. 2장에서는 Cryptomining 과정과 Cryptojacking에 대해 기술에 대해 소개하고, 3장에서는 Cryptojacking 탐지와 관련한 기존의 연구 결과를 요약한다. 4장에서는 제안하는 Cryptojacking 탐지 기법에 대해 상세히 기술하고, 5장에서는 시스템 구현 및 성능 검증 결과를 기술한다. 마지막으로, 6장에서는 전체적인 내용을 요약 기술한다.

II. 배경 지식

본 절에서는 암호 화폐 관점에서 Cryptomining 과정과 Cryptojacking에 대해 기술한다.

Table 1. Characteristics of various Cryptojacking detection methods

	Proposed system	Whoismining[3]	Antiminer[5]	Opera No Coin[6]
Well-known Cryptojacking site detection	Detected	Detected	Detected	Detected
Bypass-based Cryptojacking site detection	Detected	Undetected	Detected	Detected
Latest Cryptojacking site detection	Detected	Undetected	Undetected	Undetected

2.1 Cryptomining

Cryptomining은 소위 암호 화폐 채굴이라고 일컬어지며 채굴자가 채굴을 통해 보상을 얻어내는 일련의 과정을 뜻한다[9]. 채굴은 블록헤더의 Nonce 값을 바꿔가며 목표값 이하의 해시를 만들어 새로운 블록을 생성함으로써 이루어지고, 가장 먼저 블록 생성을 성공하는 채굴자에게만 보상이 주어진다. 이러한 Cryptomining의 특성상 암호 화폐 채굴자가 많아질수록 새로운 블록을 찾아낼 확률이 증가하기 때문에 일정한 시간동안 일정한 양의 블록만 생성될 수 있도록 그 목표값인 난이도를 높여 조절한다.

가장 먼저 블록을 생성한 채굴자에게만 보상이 주어지고 난이도의 상승으로 인해 필요로 하는 연산량이 증가하기 때문에, 채굴자들은 빠른 연산이 수행 가능한 GPU를 이용하거나 암호 화폐 채굴만을 위해 개발된 FPGA를 이용하여 주로 채굴을 수행한다.

2.2 Cryptojacking

Cryptocurrent와 Hijacking을 합친 용어인 Cryptojacking은 다른 사용자들의 PC를 이용해 Cryptomining을 수행하는 행위들을 통칭하는 공격을 일컫는다[10].

기존 Cryptojacking 방식은 다른 사용자들의 PC내에 채굴 프로그램을 몰래 설치하는 방식이 주를 이루고 있었다[11][12]. 하지만 최근에는 사용자들의 PC에 직접 채굴 프로그램을 설치하지 않고 웹 브라우저를 이용해 웹 사이트에 방문을 하는 것만으로도 방문자의 PC를 채굴에 이용하는 형태의 Cryptojacking 방식이 나타나고 있다. Fig.1.에서 이러한 웹 브라우저 기반의 Cryptojacking의 일반적인 동작 절차를 도식화하였다. 웹 브라우저를 통한 Cryptojacking은 공격자가 자바스크립트로 작성된 Cryptomining 코드를 웹 사이트에 삽입하고(①) 방문자가 해당 사이트를 방문하게 되면(②, ③) 브라우저의 JavaScript 엔진에서 해당 코드를 실행하게 된다(④). 이 후, 방문자의 동의 없이 방문자의 자원을 암호 화폐 채굴에 이용하며(④), 채굴에 의한 보상은 공격자에게 지급된다(⑤, ⑥).

대부분의 Cryptomining 코드는 WebSocket을 통해 CoinHive[13] 등과 같은 업체로부터 초기 hash값을 전달받아 Block을 생성할 때 필요한 Nonce값을 찾아내는 연산을 수행한다. 여기서 목표에 맞는 Nonce값을 찾게 되면 WebSocket을 통해 CoinHive등의 업체에 결과 값을 전달하게 되고, 위의 동작이 반복 수행된다. 이로 인해 사용자의 디바이스 자원이 낭비되어 본래의 서비스를 이용하지 못하는 결과가 나타날 수 있으며, 나아가 사용자의 디바이스 수명이 단축되는 문제가 발생할 수 있다[14].

III. 관련 연구

이 장에서는 기존의 시그니처 기반 Cryptojacking 탐지 방법에 대해 기술한다. 기존의 Cryptojacking 탐지 방법은 탐지 플랫폼에 따라 세 가지로 분류 가능하다: (1)웹 사이트 기반 탐지[3]; (2) 브라우저 기반 탐지 확장 프로그램 기반 탐지[4][5]; (3) 브라우저 기반 탐지[6].

Whoismining[3]과 같은 웹 사이트 기반 탐지 방법은 탐지 대상 사이트의 URL을 사용자로부터 입력받아 해당 사이트의 HTML 문서를 바탕으로 Cryptojacking을 탐지하는 방식으로 동작한다. 이를 위해 각 웹 사이트들은 잘 알려진 Cryptomining 코드 제공 업체의 Library 주소 리스트를 보유하며,

입력 HTML 문서에서 단순한 텍스트 매칭 알고리즘을 적용하여 탐지를 수행한다. 하지만, Library를 다른 서버에서 얻어오는 Cryptomining 코드 우회 호출의 경우에는 탐지하지 못한다. 또한, 텍스트 매칭 기반으로 탐지하기 때문에 실제로 Cryptojacking을 의도하지 않고 해당 텍스트를 언급한 경우에 오탐율이 증가하는 문제가 있다.

이를 해결하기 위해 No Mining[4], Anti Miner[5] 등의 크롬 확장 프로그램 기반 탐지 방법은 미리 만들어둔 blacklist.txt 파일을 기준으로 브라우저에서 외부 Resource를 요청하는 주소를 비교하여 blacklist의 Request를 차단하는 방식으로 동작한다. 해당 방법은 Resource 요청 주소를 기반으로 탐지하기 때문에 Cryptojacking을 우회하여 호출하더라도 탐지가 가능하다. 하지만, 미리 만들어둔 blacklist.txt 파일에 없는 Cryptomining 코드를 이용할 경우 탐지할 수 없는 문제점이 여전히 존재한다.

마지막으로 Opera 브라우저[6]의 No coin 기능은 50 베타 버전부터 제공되었으며 그 방식은 공개되지 않았으나 광고를 차단하는 방식과 비슷하게 동작하며, 페이지에 심어진 Cryptomining 스크립트를 차단한다. 이러한 No coin 기능은 크롬 확장 프로그램 기반 탐지 방법과 유사하게 Cryptomining 코드를 우회하여 호출하는 경우에는 탐지가 가능하지만, 등록되지 않은 Cryptomining 스크립트는 탐지하지 못한다.

상기 방법들은 모두 시그니처 기반 탐지 방법을 이용하기 때문에 기존 코드의 변형인 최신 Cryptomining 코드 및 새로운 유형의 Cryptojacking을 탐지하지 못한다. 반면에 제안하는 방법은 동적 분석 방식을 추가로 적용하기 때문에 잘 알려진 Cryptojacking 뿐만 아니라 새로운 Crytopjacking 공격도 탐지가 가능하다. Table 1.에서 기존의 Cryptojacking 탐지 방법과 본 논문에서 제안하는 시스템의 차이를 요약하여 기술하였다.

IV. 제안하는 시스템

이 장에서는 터미널 환경에서 GUI 없이 브라우저를 실행하는 Headless Chrome을 이용하여 Cryptojacking을 분석하는 방법에 대해 기술한다.

기존의 시그니처 기반 Cryptojacking 탐지 방법의 문제점을 해결하기 위해, 제안하는 시스템은

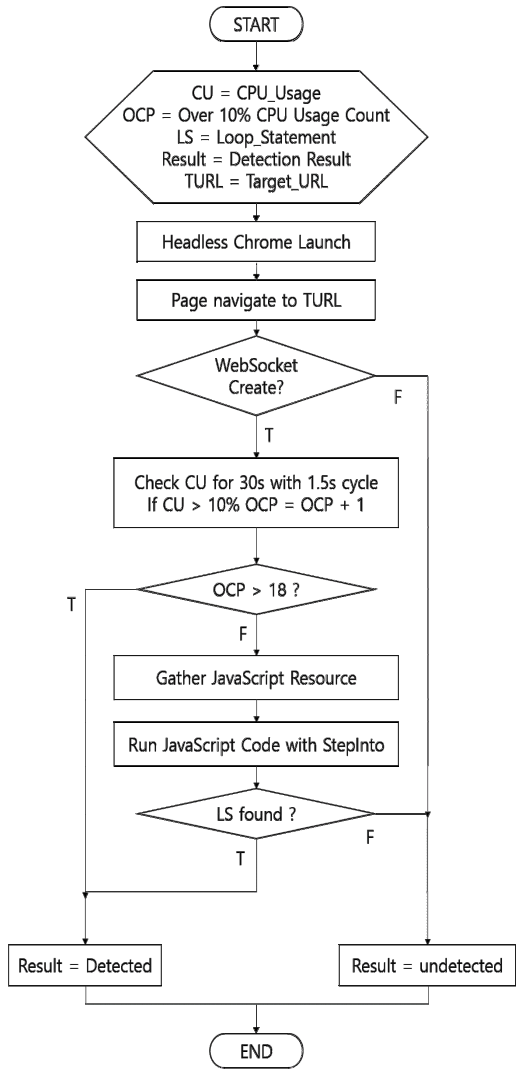

Fig. 2. Operation flow chart of the proposed analysis engine

headless chrome을 활용한 동적 분석 방법을 사용한다. 이러한 동적 분석 방법은 크게 두 절차를 통해 수행된다: (1) Cryptomining 코드 패턴 분석; (2) 분석 엔진 설계.

4.1 Cryptomining 코드 패턴 분석

Cryptomining 코드는 사용자의 추가적인 동작 없이 웹 사이트에 접속되어 페이지가 로드된 순간 CoinHive등과 같은 업체로부터 초기 hash값을 전달받아 Block을 생성할 때 필요한 Nonce값을 찾는

연산을 수행한다. 여기서 목표에 맞는 Nonce값을 찾으면 CoinHive등의 업체에 결과 값을 전달하게 되고, 위의 동작이 반복적으로 수행된다.

이러한 Cryptomining 코드는 크게 두 가지 방식으로 동작을 수행할 수 있다. 첫 번째는 브라우저에서 background로 WebWorker를 이용하는 방식이다. 이 방식은 Cryptomining 스크립트 코드에서 초기 hash값을 WebSocket을 통해 받아오고, Nonce값을 찾는 연산 작업을 WebWorker에 등록시킨다. 이후 WebWorker에서 Nonce값을 찾을 경우 WebSocket을 통해 결과 값을 전달한다. 이때 background에서 연산 작업이 수행되면서 CPU를 점유하게 된다. 이를 통해 WebSocket의 연결 여부 및 WebWorker의 사용에 따른 높은 CPU 점유율의 특징을 추출할 수 있다[15].

두 번째는 브라우저의 자바스크립트 엔진 자체를 사용하는 방식이다. 이 방식은 WebSocket을 이용하여 초기 hash값을 받아오는 점에서는 첫 번째 방식과 유사하지만, Nonce값을 찾는 과정에서 브라우저의 자바스크립트 엔진 자체를 사용하여 비동기 적으로 반복문 연산을 처리한다는 점에서 차이를 보인다. 따라서 WebSocket의 연결 여부 및 과도한 Loop Statement의 실행이라는 특징을 추출할 수 있다.

4.2 분석 엔진 설계

앞서 설명한 Cryptomining 코드 패턴 분석을 통해 추출한 특징을 바탕으로 Cryptojacking 분석 엔진을 설계한다. 본 논문에서의 Cryptojacking 분석 엔진은 다음과 같은 절차로 수행한다: (1) Headless Chrome 실행; (2) WebSocket 연결 여부 확인; (3) CPU Usage 확인; (4) TargetScript 분석. Fig.2.에서 분석 엔진의 동작 절차를 도시화 하였으며, 각 과정의 상세 동작 절차는 다음에서 설명한다.

4.2.1 Headless Chrome 실행

Headless Chrome 실행 단계에서는 Cryptojacking 분석을 위해 터미널 환경에서 Headless Chrome을 구동시킨다. 이 때, 추후 분석을 위해 Chrome의 devtools[16]에서 사용할 Page, Runtime, Network, Debugger 객체의 enable 함수를 호출한다.

Algorithm 1. Pseudo code for Loop Statement Detection(LSD) Algorithm

```
1   procedure LSD(RC, TargetScript)
2     $debugger.setScript($TargetScript);
3     for RC do
4       $exectue_lines.push(runtime.stepInto());
5     end
6     $LS_found = loopCheck($execute_lines);
7     if ($LS_found) {
8       db_update($TargetURL, 'danger')
9                 .then(() => {
10        $LS_found = false;
11        $execute_lines.clear();
12      });
13    }
14    else {
15      $execute_lines.clear();
16    }
17  end procedure
```

4.2.2 WebSocket 연결 여부 확인

4.1에서 언급했듯이, Cryptomining 코드는 Nonce값을 찾는 연산이 수행되기 전 초기값을 WebSocket으로부터 전달 받아야 한다. 이를 확인하기 위해, 제안하는 방법은 Headless Chrome의 Network 모듈에서 WebSocket을 생성하고 Handshaking 발생 관련 이벤트들을 등록하여 이벤트 발생 여부를 지속적으로 확인한다. 또한 WebSocket 주소 정보를 저장하여 이미 Cryptojacking이 이루어진다고 판단된 사이트와 동일한 주소의 WebSocket이 연결되었는지 확인한다. 그리고 이후의 분석 결과에서 Cryptojacking 사이트로 판단될 경우 WebSocket 연결 여부 확인 과정에서 저장한 WebSocket 주소를 Blacklist DB에 저장한다. 이를 통해 CPU 점유율, 마이닝 시간 등이 조정된 Cryptomining 코드가 실행되더라도 Blacklist DB에 등록된 WebSocket 주소를 참조하여 해당 Cryptomining 코드를 탐지한다. 또한 WebSocket을 통해 주고받는 데이터를 모니터링하여 동일한 길이의 패킷을 연속적으로 받았을 때 분석을 수행한다.

4.2.3 CPU Usage 확인

Cryptomining 코드를 브라우저에서 background로 실행하기 위해서는 WebWorker를 사용해야 한다. 하지만, 이는 높은 CPU 점유를 유발하기 때문에 제안하는 시스템은 Headless Chrome에서 탐지 대상 사이트에 접속해서부터 CPU의 점유율을 주기적으로 체크한다. 특정 시간동안 일정한 주기로 CPU 점유율을 확인하고 그 값이 threshold를 초과한 횟수가 90% 이상일 경우 Cryptojacking이 수행되고 있다고 판단한다. 본 논문에서는 raymond.cc에서 분석한 일반적인 Google Chome의 평균 CPU 사용량 (0%~10%)[17]를 참고하여 최저 threshold값을 10%로 설정하였다.

4.2.4 TargetScript 분석

Cryptomining 코드를 브라우저의 자바스크립트 엔진 자체를 사용해서 구동하는 경우 과도하게 반복문을 실행하게 된다. JavaScript 코드 상에서 반복문을 찾는 것은 어렵지 않지만, 일반적으로 웹 사이트에서 JavaScript 코드 배포 시에 난독화[18]를 시키기 때문에 반복문의 존재 여부를 파악하기 어렵다. 따라서 제안하는 방법은 Chrome의 devtools을 이용해서 해당 코드를 직접 실행하여 확인한다. 알고리즘 1에서 반복문 탐지를 위한 절차를 기술하였다.

먼저, 분석할 TargetScript를 Stepinto를 통해 Statement 단위로 일정 횟수만큼 코드를 실행시킨다 (line 2 - 5). 이 후, 코드가 수행하는 위치를 배열에 저장하고 배열에서 반복 여부를 파악하여 Cryptojacking 발생 여부를 판단한다 (line 6 - 7). 마지막으로, 결과를 바탕으로 Blacklist Datebase의 업데이트를 수행한다 (line 8 - 16).

V. 제안 시스템 구현 및 성능 검증

본 장에서는 제안한 Cryptojacking 분석 기법의 정상 동작을 검증하기 위하여 Cryptojacking 탐지를 위한 URL matching 서버와 동적 분석을 위한 분석 서버의 구현 방안에 대해 기술한다. 또한, 제안하는 Cryptojacking 탐지 시스템의 탐지 성능을 기존 시스템과 비교 분석한 결과를 기술한다.

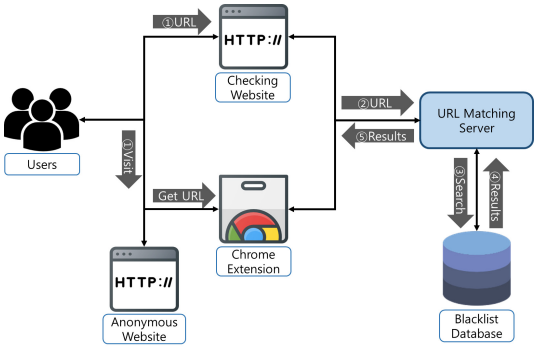

Fig. 3. Operation overview between client and URL matching server

5.1 개발 환경

5.1.1 URL matching 서버 개발 환경

Cryptojacking 탐지를 위한 matching 서버는 Ubuntu 16.04, 2.5GHz Intel Xeon Family CPU, 1GB RAM 환경에서 구현하였으며 Blacklist DB는 MySQL 5.6.39을 사용하여 구성하였다.

5.1.2 Analysis 서버 개발 환경

Cryptojacking 동적 분석을 위한 Analysis 서버는 Ubuntu 16.04, 2.5GHz Intel Xeon Family CPU, 1GB RAM 환경에서 구현하였다. 또한, chrome-launcher 0.10.2와 chrome-remote-interface 0.25.5의 Headless chrome을 이용하여 분석 엔진을 설계하였다.

5.2 URL Matching 서버 구현

URL Matching 서버는 기존 Cryptojacking 탐지 방법과 유사하게 시그니처 기반으로 탐지를 수행한다. Fig.3.은 제안하는 시스템에서 URL Matching 서버의 동작 절차를 나타낸다.

먼저, 사용자는 Matching Server에 Cryptojacking 공격 여부 확인을 요청한다(①~②). 이때, 사용자 접근성을 높이기 위해 website와 Chrome Extension의 두 가지 플랫폼을 제공한다. website의 경우, 탐지 대상 사이트의 URL을 사용자로부터 직접 입력 받는다. 반면 Chrome

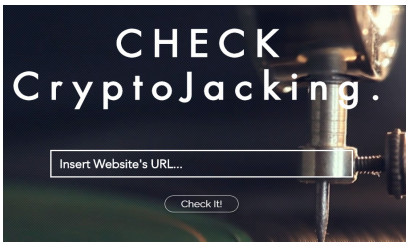

Fig. 4a. An initial page of website

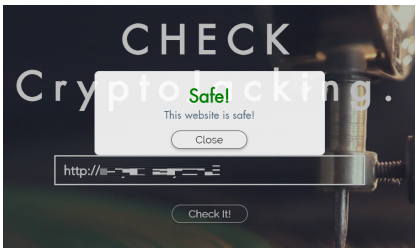

Fig. 4b. An output page of website where Cryptojacking does not occur

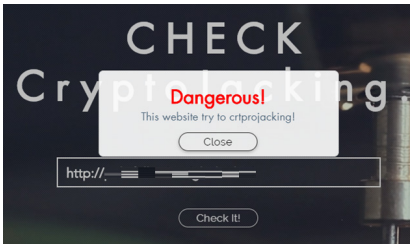

Fig. 4c. An output page of website where Cryptojacking occurs

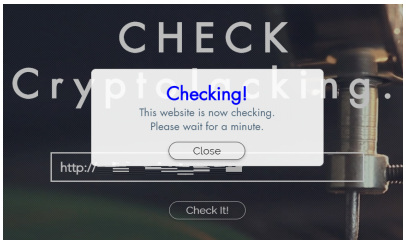

Fig. 4d. An output page of website where the requested URL does not exist in Blacklist DB

Fig. 4. Output screen of proposed system according to the requested url analysis results

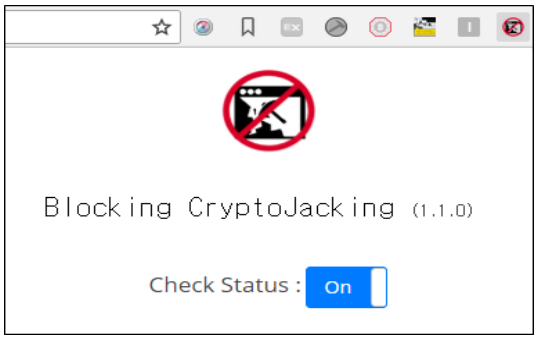

Fig. 5. Initial page of chrome extension

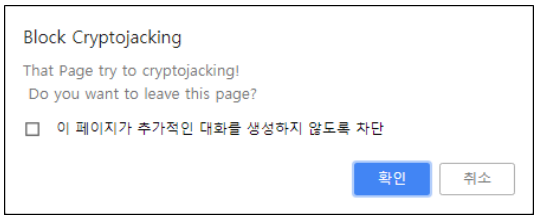

Fig. 6. Cryptojacking detection results using Chrome Extension

Extension의 경우, 사용자가 방문 중인 사이트의 URL을 자동으로 수집하여 요청을 전송한다. 사용자

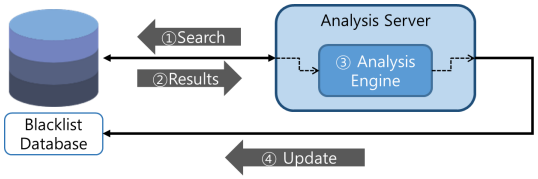

Fig. 7. Operation overview between analysis server and blacklist database

로부터 요청을 받은 Matching 서버는 요청 URL을 Blacklist DB에서 조회하고(③~④) 사용자에게 그 결과를 전달한다(⑤). 만약 요청 URL이 Blacklist DB에 존재하지 않는다면 해당 URL은 대기 상태로 Blacklist DB에 등록된다.

Fig.4a.는 Cryptojacking 탐지를 위한 website의 초기 화면을 나타낸다. Fig.4b. ~ Fig.4d.는 사용자가 website로 URL을 요청했을 때의 결과 화면을 나타낸다. website는 Cryptojacking 여부에 따라 safe (Fig.4b.) 또는 dangerous (Fig.4c.) 표시로 화면에 보여준다. 요청 URL이 Blacklist DB에 존재하지 않을 때는 checking 표시를 출력한다 (Fig.4d.).

Fig.5.는 Chrome Extension의 초기 화면을

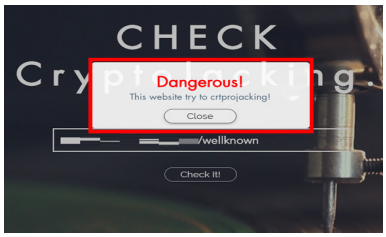

Fig. 8a. Proposed system

Fig. 8b. Whoismining

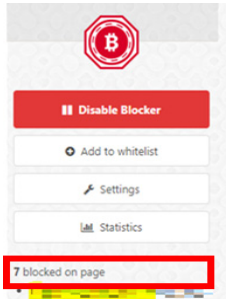

Fig. 8c. Antiminer

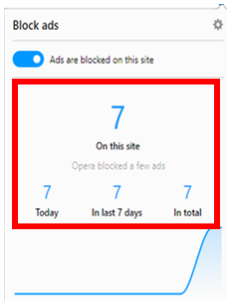

Fig. 8d. Opera No Coin

Fig. 8. Comparison of detection results against the well-known Cryptojacking site

나타낸다. Chrome Extension의 경우 Cryptojacking이 발생하지 않거나 요청 URL이 Blacklist에 존재하지 않을 때는 아무런 내용을 출력하지 않는다. 대신에, Fig.6.과 같이 요청된 URL의 결과가 danger인 경우 현재 방문 중인 사이트가 Cryptojacking 공격을 수행중임을 사용자에게 알려주고 방문 중인 사이트를 벗어날지의 여부를 사용자에게 선택하는 경고 문구(alert)를 표시한다.

5.3 Analysis 서버 구현

Analysis 서버는 Blacklist DB내의 대기 상태로 등록된 URL에 대해 동적 분석을 수행한다. Fig.7.는 제안하는 시스템에서 Analysis 서버의 동작 절차를 나타낸다.

먼저, Analysis 서버는 Blacklist DB에 등록된 대기 상태의 URL을 주기적으로 요청한다 (①). 등록된 대기 상태의 URL이 존재하면 Blacklist DB는 Analysis 서버로 URL을 전송하며 (②), URL을 수신한 Analysis 서버는 4장에서 설명한 분석 엔진을 이용해 동적으로 Cryptojacking 여부를 판단 한다 (③). 이 후, Analysis 서버는 분석 엔진의 결과에 따라 Blacklist DB의 상태 업데이트를 요청한다 (④).

5.4 Cryptojacking 탐지 결과

제안하는 Cryptojacking 탐지 시스템의 탐지 성능을 검증하기 위해 본 논문에서는 3가지 유형의 Cryptojacking에 대해 Whoismining[3], Antiminer[5], Opera No Coin[6]과 제안하는 탐지 시스템의 탐지 성능을 비교하였다.

5.4.1 잘 알려진 Cryptojacking 사이트

본 논문에서는 잘 알려진 Cryptojacking을 수행하기 위해서 브라우저 기반 Cryptomining으로 검색결과 상위에 노출되는 업체의 Cryptomining Library를 직접 이용하여 사이트를 제작해 테스트를 수행하였다[19][20].

Fig.8.은 잘 알려진 Cryptojacking 사이트에 대한 각 방법의 탐지 결과를 나타낸다. 잘 알려진 Cryptojacking 사이트의 경우, 이미 각 시스템의 시그니처로 등록이 되어있다고 판단할 수 있으며 기존 시스템과 제안하는 시스템 모두 탐지가 가능하다.

5.4.2 Cryptojacking을 우회하여 호출하는 사이트

우회를 통한 Cryptomining 코드 호출을 위해,

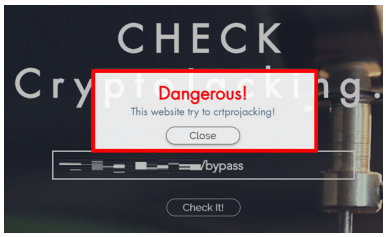

Fig. 9a. Proposed system

Fig. 9b. Whoismining

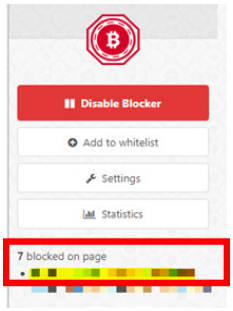

Fig. 9c. Antiminer

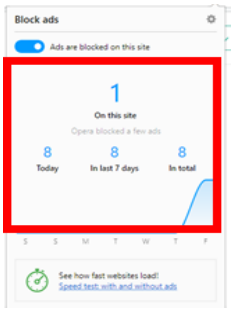

Fig. 9d. Opera No Coin

Fig. 9. Comparison of detection results against call-by-bypassing Cryptojacking site[19]

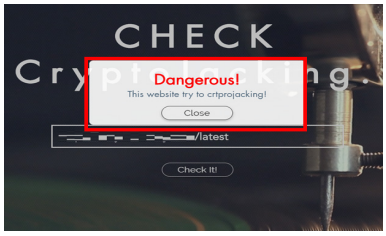

Fig. 10a. Proposed method

Fig. 10b. Whoismining

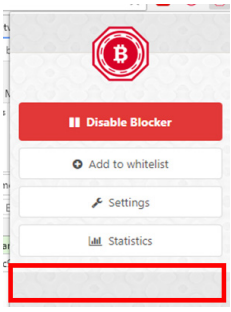

Fig. 10c. Antiminer

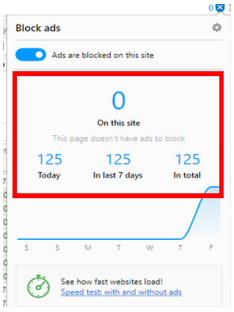

Fig. 10d. Opera No Coin

Fig. 10. Comparison of detection results against the latest Cryptojacking site[20]

본 논문에서는 이미 잘 알려진 유명 업체의 Cryptomining Library를 직접 호출하지 않고 제3의 서버를 따로 두어 서버로부터 Library를 호출하도록 테스트를 수행하였다[19].

Fig.9.는 Cryptomining 코드를 우회하여 호출하는 사이트에 대한 각 방법의 탐지 결과를 나타낸다. 제안하는 방법과 Antimier, Opera No Coin은 실질적으로 외부 Resource를 요청하는 주소를 비교

하기 때문에 Crptojacking을 우회하는 경우에도 탐지가 가능하다. 하지만, Whoismining의 경우 단순한 URL 텍스트 매칭 방식으로 동작하기 때문에 우회하여 호출하는 경우를 탐지하지 못한다.

5.4.3 최신 Cryptojacking을 이용한 사이트

최신 Cryptojacking에 대한 탐지 성능을 비교하기 위해, 기존 업체에서 Antiminer나 adBlocker에 탐지 되지 않도록 최신 업데이트된 라이브러리를 이용하여 테스트를 수행하였다[20].

Fig.10.는 최신 Cryptojacking을 이용한 사이트에 대한 각 방법의 탐지 결과를 나타낸다. 제안하는 방법의 경우 WebSocket 연결 여부, CPU 사용량, 반복문 등의 동적 분석을 통해 새로운 Cryptojacking를 효율적으로 탐지할 수 있다. 하지만 시그니처 기반으로 동작하는 기존의 탐지 방법들은 새로운 Cryptojacking에 대해 전혀 탐지하지 못한다.

5.4.4 탐지 결과 요약

브라우저 기반 Cryptomining으로 검색된 상위 업체에서 제공하는 잘 알려진 Cryptomining Library[13][19]를 이용한 Cryptojacking site에 대해 시험한 결과 제안하는 방법, Whoismining, Antiminer 및 Opera No Coin 모두에서 정상적으로 Cryptomining 코드의 존재를 탐지함을 확인하였다. 하지만, 우회하여 Cryptomining 코드를 실행한 Cryptojacking site에 대해서는 Whoismining의 경우 탐지하지 못하고, 그 외의 방법들에서는 정상적으로 탐지함을 확인하였다. 또한, 기존에 존재하지 않던 변형된 최신 Cryptomining 코드[20]를 포함한 Cryptojacking site에 대해서는 제안하는 방법을 제외한 다른 방법들에서는 해당 시그니처가 존재하지 않아 정상적으로 탐지 불가능함을 확인하였다.

VI. 결 론

본 논문에서는 터미널 환경에서 GUI 없이 브라우저를 실행하는 Headless Chrome을 이용하여 Cryptojacking의 행위를 동적으로 분석하는 방법을 제안하였고, 이를 활용한 Cryptojacking 탐지 시스템을 개발하였다. 동적 분석을 통해 제안하는 시스템은 기존의 시그니처 기반 시스템에서 탐지하지 못하는 새로운 Cryptojacking을 탐지 할 수 있으며, Cryptojacking 사이트를 우회하여 호출하거나 Cryptomining 코드를 난독화하더라도 탐지 가능한 장점을 가진다. 또한, 제안하는 Cryptojacking 탐지 시스템은 크롬 확장 프로그램과 웹 사이트의 두 가지 플랫폼을 제공하기 때문에 사용자 접근성이 우수하다.

향후 연구에서는 특정 Cryptomining 코드에서 thread 값의 조정을 통한 CPU 점유율을 낮추는 경우에 대한 threshold값의 정확도를 증가시키고 오탐율을 감소시키기 위한 연구를 수행할 것이다. 또한, WebSocket으로 전달된 데이터 중 동일한 길이의 패킷이 연속적으로 수신되지 않더라도 Cryptomining이 동작하는 예외적인 상황에 대한 연구를 수행할 것이다.

References

[1] Frank Holmes, “As Banknotes Disappear Will Bitcoin Take Its Place?,” US Global Investors, Apr. 2018.

[2] LeeYuJi, “‘Cryptojacking’ Gold Rush … 8500% increase compared with last year,” byline.network, Apr. 2018.

[3] whoismining, “who is mining” https://whoismining.com, Apr. 2018.

[4] No mining, “No mining Chrome Extension” https://chrome.google.com/webstore/detail/no-mining-block-coin-mini/hoafonbifbfcbhdconhnmcphpnplaekb?hl=ko, Apr. 2018.

[5] Anti Miner, “Anti Miner Chrome Extension” https://chrome.google.com/webstore/detail/anti-miner-no-1-coin-mine/ibhpgkhoicjhklmbhdoeikeggbeejonj?hl=ko, Apr. 2018.

[6] Opera No coin, “Opera Browser”https://www.opera.com, Apr. 2018.

[7] K. Hughes and Y. Qu, “Performance Measures of Behavior-Based Signatures: An Anti-malware Solution for Platforms with Limited Computing

Resource," 2014 Ninth International Conference on Availability, Reliability and Security, Fribourg, pp. 303-309, Dec. 2014.
[8] Headless Chrome, "Headless Chrome developer doc" https://developers.google.com/web/updates/2017/04/headless-chrome, May. 2018.
[9] BitcoinWiki, "Mining" https://en.bitcoin.it/wiki/Mining, May. 2018.
[10] "Cryptojacking - Cryptomining in the browser," Ensia, Nov. 2017.
[11] "Ransomware, infected like this," AhnLab, Dec, 2015.
[12] JungvinHwang, "Cryptomining using Wifi," zdnet, Apr. 2018.
[13] CoinHive, "CoinHive - Monero JavaScript Mining" https://coinhive.com, May. 2018
[14] hamza-ahmad, "Risks of Bitcoin mining by personal computers," steemit, 2017.
[15] M. Wenzel and C. Meinel, "Parallel network data processing in client side JavaScript applications," 2015 International Conference on Collaboration Technologies and Systems (CTS), Atlanta, GA, pp. 140-147, Aug, 2015.
[16] Chrome DevTools Protocol Viewer, "Chrome DevTools" https://chromedevtools.github.io/devtools-protocol/tot, May. 2018.
[17] HAL9000, "10 Popular Web Browsers Tested for Memory and CPU Usage," raymond.cc, 2016
[18] W. Xu, F. Zhang and S. Zhu, "The power of obfuscation techniques in malicious JavaScript code: A measurement study," 2012 7th International Conference on Malicious and Unwanted Software, Fajardo, PR, pp. 9-16, Feb. 2012.
[19] JSECoin, "JSECoin: Digital Currency - Designed for the web", https://jsecoin.com, May. 2018.
[20] CryptoLOOT, "CryptoLoot - Earn More From Your Traffic" https://crypto-loot.com, May. 2018.

〈저자소개〉

고 동 현 (DongHyun Ko) 학생회원
2012년 3월~현재: 부산대학교 정보컴퓨터공학부 학사 과정
〈관심분야〉 네트워크 보안, 블록체인, 개인정보보호

정 인 혁 (InHyuk Jung) 학생회원
2012년 3월~현재: 부산대학교 정보컴퓨터공학부 학사 과정
〈관심분야〉 모바일 Application 개발, 웹 Front-End 개발, 네트워크 보안

최 석 환 (Seok-Hwan Choi) 학생회원
2016년: 부산대학교 정보컴퓨터공학부 학사
2016년 9월~현재: 부산대학교 전자전기컴퓨터공학과 석사 과정
〈관심분야〉 모바일 보안, 무선 네트워크 보안

최 윤 호 (Yoon-Ho Choi) 종신회원
2008년: 서울대학교 전기컴퓨터공학부 박사
2010년: 펜실베니아 주립대학교 박사후 연구원
2012년: 삼성전자 네트워크사업부 책임연구원
2014년: 경기대학교 융합보안학과 조교수
2016년: 부산대학교 전기컴퓨터공학부 조교수
2016년~현재: 부산대학교 전기컴퓨터공학부 부교수
〈관심분야〉 모바일 보안, 유무선 네트워크 침입탐지, IoT 보안 프로토콜, 경량 암호, 지능형 자동차 IT 보안 등