논문 19-44-08-23 The Journal of Korean Institute of Communications and Information Sciences '19-08 Vol.44 No.08
https://doi.org/10.7840/kics.2019.44.8.1623

페이스트재킹을 이용한 크립토재킹 공격 및 방어 기법

고 동 현◆, 최 석 환*, 황 선 진*, 최 윤 호°

Cryptojacking Attack Using Pastejacking and Defense Method

DongHyun Ko◆, Seok-Hwan Choi*, Seonjin Hwang*, Yoon-Ho Choi°

요 약

비트코인과 같은 암호화폐의 열풍으로 타인의 컴퓨터를 이용해 암호화폐를 불법적으로 채굴하는 크립토재킹(Cryptojacking) 공격이 등장했다. 크립토재킹 공격은 공격 효율 향상을 위해 다양한 방식으로 고도화 되고 있으며, 새로운 유형의 크립토재킹 공격들은 시그니처 기반의 기존 탐지 시스템에서는 탐지가 불가능하다. 본 논문에서는 최신 크립토재킹 공격 동향을 바탕으로 앞으로 발생할 수 있는 새로운 공격 유형인 페이스트재킹(Pastejacking)을 이용한 크립토재킹 공격 기법을 제안하고 또한, 이를 효과적으로 탐지하기 위한 방어 기법들을 제안한다. 제안하는 공격 기법은 웹 페이지의 '복사하기' 이벤트 취약점을 노린 페이스트재킹을 이용하여 기존 크립토재킹 공격에 비해 더 많은 기기를 대상으로 더 효율적인 크립토재킹 공격을 가능하게 한다. 이에 대한 제안하는 방어 기법들인 규칙 기반 탐지 방법과 동적 탐지 방법은 크립토재킹 공격을 일으키는 페이스트재킹만을 탐지하여 기존 범용 페이스트재킹 탐지 방법의 오탐을 제거하면서 크립토재킹 공격 탐지 방법의 정탐율을 향상시킨다.

Key Words : Cryptojacking, Pastejacking, Blockchain, Cryptocurreny, Cryptomining

ABSTRACT

A Cryptojacking attack has emerged that illegally cryptomined the crypto-currency using victim's computer. Cryptojacking attacks are being advanced in various ways to improve attack efficiency. These new type of Cryptojacking attacks are not detectable in existing signature-based detection systems. In this paper, we propose a new Cryptojacking attack method using Pastejacking by analyzing trends of Cryptojacking attack and propose defense method that can effectively defend it. The proposed attack method uses Pastejacking which is aimed at vulnerability of 'copying' event of web page. This proposed attack method can perform more efficient Cryptojacking attack on more devices than conventional Cryptojacking attacks. Rule-based detection and dynamic detection, which are proposed defense methods, can improve the detection rate of the Cryptojacking attack by detecting only Pastejacking which causes a Cryptojacking attack and eliminating false positive of the conventional general Pastejacking detection method.

※ 본 논문은 한국연구재단 - 논문연구과제(NRF-2018R1D1A3B07043392)와 한국연구재단 - 과학문화 전시서비스 역량강화 지원 사업(NRF-2018X1A3A1069642) 및 BK21플러스, IT기반 융합산업 창의인력양성사업단의 연구결과로 수행되었습니다.
◆ First Author : Pusan National University, uyt1209@pusan.ac.kr, 학생회원
° Corresponding Author : Pusan National University, yhchoi@pusan.ac.kr, 종신회원
* Pusan National University, daniailsh@pusan.ac.kr, 학생회원; unlockable@pusan.ac.kr, 학생회원
논문번호 : 201905-056-B-RE, Received April 29, 2019; Revised July 3, 2019; Accepted July 5, 2019

Ⅰ. 서 론

블록체인 기반의 암호화폐인 비트코인[1]의 등장과 열풍으로 이더리움[2], 모네로[3] 등의 다양한 암호화폐가 등장하였다[4]. 이 후, 암호화폐에 대한 관심이 커져 가격이 폭등하자 사이버 범죄자들은 피해자의 컴퓨터를 이용해 불법적으로 암호화폐를 채굴하는 크립토재킹 공격을 시작했다[5,6]. 초기 크립토재킹 공격은 기업 서버를 해킹하여 크립토마이닝 툴을 설치하는 설치기반 크립토재킹 공격이 위주였으나 2017년 9월 Monero 코인 채굴을 위한 CoinHive Miner[7]의 등장으로 인해 피해자의 PC에 별도의 설치 없이 손쉽게 공격이 가능한 브라우저 기반의 크립토재킹 공격이 크게 증가하였다[8,9]. 이 후, 브라우저 기반의 크립토재킹 공격이 널리 알려지면서 JSECoin[10], Crypto-Loot[11] 등의 다양한 브라우저 기반 크립토마이닝 툴들이 등장하여 크립토재킹 공격이 다양화 되었다.

2017년 12월 이후 암호화폐 가치가 하락세를 지속하고 있으나 크립토재킹 공격은 낮은 진입 장벽과 최소한의 오버헤드로 악성코드 위협이 여전히 유지되고 있다. 최신 크립토재킹 공격은 공격 효율 향상을 위해 코드 난독화, 연산 자원 소비량 제한, 코드 최적화 등 다양한 방식으로 고도화가 이루어지고 있다[12].

따라서, 본 논문에서는 크립토재킹 공격의 변화에 미리 대응하기 위해 웹 페이지의 '복사하기' 이벤트 취약점을 노린 페이스트재킹을 이용한 새로운 방식의 크립토재킹 공격에 대한 시나리오를 분석하고, 이에 대응하기 위한 방법을 제안한다.

페이스트재킹을 이용한 크립토재킹 공격은 개발환경 설정을 위해 웹 페이지에서 명령어를 복사하여 터미널에 붙여 넣는 과정에서 크립토재킹 공격이 이루어진다. 공격자는 사용자가 복사하는 개발환경 설정 명령어에 크립토재킹 공격을 일으킬 수 있는 명령어를 몰래 추가하여 크립토재킹 공격을 수행한다. 이 공격은 특정 서버 PC 등을 대상으로 하는 기존 설치기반 크립토재킹 공격보다 더 많은 디바이스를 대상으로 하기 때문에 범용성 측면에서 위협적이며, 웹 사이트를 벗어나기만 해도 공격이 차단되는 기존 브라우저 기반 크립토재킹 공격보다 더 효율적인 방식으로 공격을 수행할 수 있다.

이를 방어하기 위해서는 웹 사이트에서 페이스트재킹을 차단하는 것이 효과적이며 Hardened Paste[13]라는 확장프로그램을 이용할 수 있다. 그러나 Hardened Paste는 웹 사이트 컨텐츠에 대한 출처 표시를 남기는 코드와 같은 사용자 정의 이벤트 코드 또한 동작하지 못하도록 차단한다는 문제점이 존재한다. 따라서 본 논문에서는 규칙(Rule) 기반 탐지 방법 및 동적 탐지 방법을 통해 사용자 정의 이벤트는 정상 동작하면서 페이스트재킹을 이용한 크립토재킹 공격을 효과적으로 방어할 수 있는 기법을 제안한다.

본 논문의 구성을 요약하면 다음과 같다. 2장에서는 크립토재킹 및 페이스트재킹에 대해 소개하고, 크립토재킹 및 페이스트재킹 탐지에 관한 기존 방법을 요약한다. 3장에서는 페이스트재킹을 이용한 크립토재킹 공격의 새로운 공격 가능성에 대해 분석하고, 이를 방어하기 위한 기법을 제안한다. 4장에서는 시스템 구현 및 실험 결과를 기술한다. 마지막으로, 5장에서는 전체적인 내용을 요약한다.

Ⅱ. 배경지식 및 기존 연구

이 장에서는 크립토재킹 및 페이스트재킹에 대해 기술하고, 각 공격에 대한 기존 탐지 방법에 대해 기술한다.

2.1 크립토재킹(Cryptojacking)

크립토재킹이란 암호화폐(Cryptocurrency)와 하이재킹(Hijacking)의 합성어로 해커가 피해자의 PC에 암호화폐 채굴을 위한 악성코드를 몰래 삽입하여 피해자의 PC자원을 이용해 채굴한 암호화폐를 자신의 전자지갑으로 전송하는 사이버 범죄를 일컫는다.

크립토재킹 공격은 크게 두 가지 방식으로 나누어진다: 설치기반 크립토재킹과 브라우저 기반 크립토재킹.

설치기반 크립토재킹 공격은 공격 대상의 PC에 크립토마이닝 프로그램을 직접 설치하여 크립토마이닝을 수행하는 방식이다. 설치기반 크립토재킹 공격은 공격 효율성 측면에서는 효과적이지만 공격 대상의 PC에 크립토마이닝 프로그램을 설치하는 과정의 난이도가 높고 최신화가 어렵기 때문에 최신 버전의 백신 프로그램에 의해 탐지될 가능성이 높다.

반면, 브라우저 기반 크립토재킹 공격은 설치과정 없이 웹 사이트를 방문한 무작위 PC를 이용하여 크립토마이닝을 수행하는 방식이다. 브라우저 기반의 크립토재킹 공격은 CoinHive, JSECoin, Crypto-Loot 등을 이용하여 단순히 웹 사이트에 접속한 사용자를 대상으로 쉽게 크립토재킹 공격을 할 수 있다. 또한, 악성코드 최신화가 간단하기 때문에 기존 크립토재킹 탐지 시스템을 회피하기 쉽지만, 사용자가 웹 사이트를 종료하기만 해도 공격이 자동으로 차단되므로 효율성

이 떨어진다.

설치기반 크립토재킹 공격은 V3[14], Window Defenser[15] 등의 백신 프로그램에 미리 정의된 악성 코드 시그니처 기반으로 탐지하고 있다. 반면, 브라우저 기반의 크립토재킹 공격은 NoCoin[16], Anti Miner[17], CoinEater[18] 등의 브라우저 확장 프로그램에 미리 정의된 크립토마이닝 연결 URL 시그니처 기반으로 탐지하고 있다[19]. 또한, 최근에는 시그니처 기반 탐지 방법의 한계를 극복하기 위한 동적 분석 기반의 크립토재킹 탐지기법이 방법이 연구되었다[20, 21].

2.2 페이스트재킹(Pastejacking)

페이스트재킹이란 붙여넣기(Paste)와 하이재킹(Hijac -king)의 합성어로 웹 사이트에서 '복사하기 & 붙여넣기' 기능을 이용할 때 발생하는 이벤트 취약점을 노린 해킹공격 기법을 일컫는다.

이 공격 기법은 2016년 5월 Malwarebytes에 의해 웹 사이트에서 터미널 명령어를 복사하여 터미널에 붙여 넣는 과정에서 공격자가 지정한 명령어가 추가 실행될 수 있다고 발견된 이래로 현재 개념증명(PoC) 단계에 있는 공격으로 현재까지는 별다른 피해사례는 보고되지 않았다[22]. 그러나 최근 비전공자를 대상으로 하는 소프트웨어 교육이 확대됨에 따라 개발에 필요한 툴을 설치하는 과정에서 사용자 부주의로 인해 페이스트재킹 공격 대상이 될 가능성이 크게 증가하였다.

페이스트재킹은 브라우저 별로 다양한 방식으로 공격이 가능하다. 먼저 Internet Explorer의 경우 JavaScript에서 execCommand를 이용하여 직접 시스템 클립보드의 값을 읽고 쓸 수 있다. '복사하기' 이벤트가 발생할 경우 setData 함수를 호출하여 클립보드에 복사되는 내용을 수정하는 방식으로 공격이 이루어질 수 있다. 그러나 FireFox 등의 브라우저는 클립보드에 대한 프라이버시 보호 등의 이유로 execCommand의 강제 트리거를 막도록 설정되어 있다. 이런 브라우저는 시스템 클립보드의 값을 직접 수정하는 것이 아닌 복사하고자 select한 영역에 대한 수정을 통해 시스템 클립보드에 공격자가 원하는 내용을 넣을 수 있다.

페이스트재킹 방어는 브라우저 확장 프로그램인 Hardened Paste를 통해 웹 사이트의 사용자 정의 '복사하기' & '잘라내기' 이벤트 대신 브라우저 기본 내장 이벤트가 발생하도록 방식으로 동작한다. 그러나 Hardened Paste의 방식은 웹 사이트의 컨텐츠에 대한 출처 표시를 남기는 코드와 같은 사용자 정의 '복사하기' & '잘라내기' 이벤트 또한 수행되지 않도록 막는다는 문제점이 있다.

Ⅲ. 페이스트재킹을 이용한 크립토재킹

이 장에서는 크립토재킹의 최근 동향을 바탕으로 페이스트재킹을 이용한 공격 시나리오를 제안하고, 방어기법에 대해서 제안한다.

3.1 공격 방법

제안하는 공격은 터미널을 이용하여 특정 프로그램, 툴을 설치하거나 개발환경을 세팅하는 경우 웹 사이트의 내용을 그대로 붙여 넣음으로써 공격이 일어날 수 있다. 따라서 공격자는 '복사하기' 이벤트를 바인딩하여 사용자가 웹 사이트에서 명령어를 복사하는 순간 클립보드에 크립토재킹 공격 코드를 같이 넣어 터미널에서 실행하는 경우 페이스트재킹을 이용한 크립토재킹 공격이 가능하도록 한다. Fig.1.은 공격 방법에 대한 동작 절차를 나타낸다.

공격자는 웹 사이트의 '복사하기' 이벤트를 바인딩하여 '복사하기' 이벤트가 발생할 때 공격자의 코드가 우선 수행되도록 한다. 공격자의 코드는 다음과 같은 절차로 동작한다. 먼저, 브라우저의 웹 페이지에서 복사하기 위해 현재 선택된 내용을 가져온다. 이 후, 사용자의 눈에 보이지 않는 위치 임시 태그를 생성하고, 새로 생성된 태그 내에 사용자가 선택한 내용과 크립토재킹 공격을 일으키기 위한 명령어들을 추가한다. 이 후, 브라우저 기본 복사 이벤트를 실행하여 클립보드로 복사되는 내용을 사용자가 선택했던 내용에 공격 코드가 추가된 것으로 변경하여 터미널에 붙여 넣을 때 크립토재킹 공격이 수행되도록 한다. 마지막으로, 임시로 만들었던 태그를 삭제 한다.

페이스트재킹을 통해 추가되는 크립토재킹 공격 명령어의 기본 구성은 다음과 같다. 크립토마이닝을 위한 툴 설치 명령어, 크립토재킹 공격 파일을 다운로드

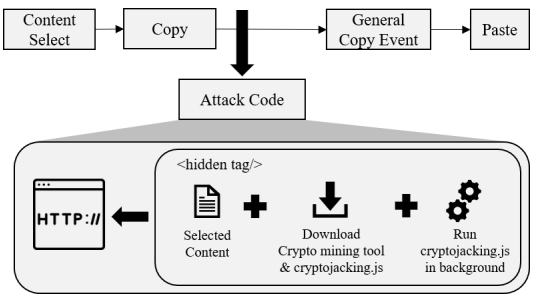

Fig. 1. Operation flow chart of the attack code

하는 명령어 및 다운로드한 크립토재킹 공격 파일을 백그라운드에서 유저 몰래 동작시키는 명령어로 이루어져 있다. 이때, 크립토재킹 공격 명령어는 개발환경 세팅을 하고자하는 모든 PC를 대상으로 하기 때문에 터미널에서 사용하는 기본 명령어의 조합을 이용하여 크립토재킹 공격을 일으킨다. 또한 npm[23], pip[24] 등과 같은 패키지 관리자가 미리 설치되어 있는 PC의 경우에는 터미널 기본 명령어 조합이 아닌 패키지 관리자를 이용한 명령어 조합으로도 크립토재킹 공격을 일으킬 수도 있다.

또한, 페이스트재킹을 이용한 크립토재킹 공격은 텍스트 데이터를 기반으로 공격이 이루어지기 때문에 Escape 문자 등을 활용하여 명령어의 변형이 가능하다. 이로 인해 웹 페이지 상에서는 터미널 명령어와 Escape 문자가 같이 존재하여 명령어가 아닌 것처럼 보이지만, 터미널에 붙여 넣는 순간 Escape 문자가 처리되어 터미널 명령어로 취급되기 때문에 다양한 공격이 가능하다.

이를 통해 설치기반의 크립토재킹 공격에 비해 더 많은 디바이스를 대상으로, 브라우저 기반의 크립토재킹 공격에 비해 더 효율적인 방식으로 크립토재킹 공격을 수행할 수 있다.

3.2 방어 방법

페이스트재킹을 이용한 크립토재킹 공격을 효과적으로 방어하기 위해서는 '복사하기' & '잘라내기'의 사용자 이벤트를 차단하는 방법을 이용할 수 있다. 하지만, 웹 사이트내의 컨텐츠에 대한 출처 표시를 남기는 코드와 같은 유익한 사용자 정의 이벤트는 그대로 유지할 수 있도록 해야 한다. 따라서 본 논문에서는 출처 표시를 남기는 코드와 같이 '복사하기' & '잘라내기' 이벤트를 유용하게 바인딩하는 경우는 유지하고, 터미널 명령어가 추가된 경우만을 차단하는 탐지 방법을 제안한다. Fig.2.에서 제안하는 방어 방법에 대한 동작 흐름을 나타낸다.

먼저, 브라우저 확장 프로그램을 통해 웹 페이지 로드 시작 시점에 복사하기 이벤트 핸들러를 설정한다. 이를 이용하여 웹 페이지에서 '복사하기' & '잘라내기' 이벤트를 바인딩 하는지 여부를 우선 확인한다. 만약 이벤트 바인딩을 하고 있다면, 사용자 정의 이벤트가 일어나기 전 복사하고자하는 내용을 저장하고 사용자 정의 이벤트가 실행되도록 한다. 사용자 정의 이벤트가 끝난 후, 선택된 내용을 가져와 사용자 정의 이벤트 전 후의 텍스트를 비교한다. 여기서 추가되는 내용이 터미널 명령어인지 여부를 확인하여 페이스트재킹을 차단하면 된다. 터미널 명령어 여부를 확인하는 방법은 두 가지 방법으로 구성된다: (가) 규칙 기반 탐지(Rule-based Detection); (나) 동적 탐지 (Dynamic Detection).

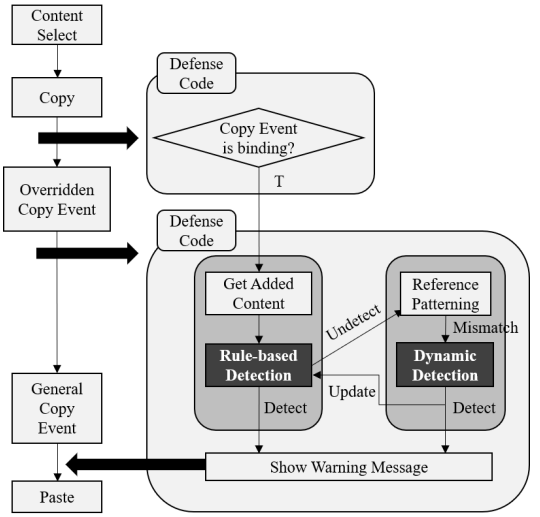

Fig. 2. Operation flow chart of the defense code

3.2.1 규칙 기반 탐지(Rule-based Detection)

터미널에서 사용되는 명령어의 종류는 한정적이기 때문에 규칙 기반의 탐지 방법에 터미널 명령어 리스트를 규칙(Rule)으로 정의하여 효과적으로 명령어 여부를 확인하고 쉽게 차단할 수 있다. 또한 기본 명령어는 OS 재단에서 관리하기 때문에 공식 문서 등을 통해 규칙의 업데이트에도 활용할 수 있다. 그리고 npm, pip 등과 같은 패키지 관리자를 이용하는 경우도 주요 패키지 관리자 목록을 지속적으로 추가하여 차단할 수 있다. Algorithm 1.은 제안하는 규칙 기반 탐지 방법에 대한 동작 절차를 나타낸다.

먼저 웹 사이트에서 일어나는 이벤트 중 '복사하기' 이벤트가 발생할 경우를 판단한다(lines 2 – 3). 이후 규칙(Rule)으로 정의한 터미널 기본 명령어들의 리스트인 cmdList와의 비교를 통해 현재 선택된 내용에 터미널 기본 명령어가 포함될 경우 사용자에게 경고 메시지를 보여주도록 한다(lines 4 – 10).

3.2.2 동적 탐지(Dynamic Detection)

기본 터미널 명령어에 Escape 문자 등을 활용한 텍스트 데이터 변형을 통한 공격의 경우 Rule을 정의할 수 없으므로 실제 터미널과 같은 환경에서 실제로 수행해보는 동적 탐지 방법을 이용하여 크립토재킹 공격을 차단할 수 있다. 제안하는 동적 탐지 방법은 다

음과 같다.

먼저, '복사하기' 이벤트를 바인딩하는 웹 사이트를 대상으로 사용자 정의 이벤트에 의해 추가되는 내용을 미리 정의된 출처 패턴을 이용하여 패턴 비교를 수행한다. 만약 패턴과 일치하지 않는 경우 추가되는 내용을 가상환경의 테스트 터미널에서 직접 돌려보고 공격 여부를 판단할 수 있도록 서버에 전송한다. 크립토재킹 공격이 이루어지기 위해서는 크립토마이닝 툴 및 크립토재킹 공격 코드에 대한 다운로드가 이루어져야 한다. 따라서 패킷 캡쳐 등을 이용하여 외부와 연결 여부를 확인한 후 외부와 연결이 일어나는 경우 공격이 이루어진다고 판단한다. 마지막으로, 공격이라고 판단된 경우 사용자에게 알려주고 규칙(Rule)을 업데이트한다.

이를 통해 3.1.에서 제안한 페이스트재킹을 이용한 크립토재킹 공격을 효과적으로 방어할 수 있을 뿐만 아니라, 컨텐츠에 대한 출처 표시를 남기는 코드와 같은 사용자 정의 이벤트 코드가 정상 동작할 수 있도록 할 수 있다.

Ⅳ. 구현 및 실험

본 장에서는 제안한 페이스트재킹을 이용한 크립토재킹 공격 방법을 구현하여 실제 공격을 수행하는 방법과 방어 기법에 대해 기술한다.

4.1 실험 환경

4.1.1 공격 구현 개발 환경

공격을 위한 웹 서버는 Ubuntu 14.04에서 Ruby on Rails[25] 기반으로 구현하였고, '복사하기' 이벤트 바인딩 코드는 jQuery를 이용하여 Internal JavaScript로 구현하였다.

4.1.2 방어 구현 개발 환경

방어 시스템은 Chrome Extension을 이용하여 구현하였고, 동적 탐지를 위해 Virtual Box 5.2를 이용하여 Ubuntu 16.04, AMD Ryzen 5 1500x Quad-Core, 4GB RAM 환경에서 구현하였다. 또한, node.js를 이용하여 API서버를 구현하였고, 패킷 분석을 위해 python 3의 pyshark 0.4.2 패키지를 이용하여 분석을 수행하였다.

Algorithm 1. Pseudo code for rule-based defense

```
1   procedure detect_paste_crypto_jacking(cmdList)
2     EventTarget.prototype.addEventListener =
      function(type, listener, useCapture) {
3       if (type == 'copy' || type == 'cut') {
4         copyText = getCopyText();
5         copyTextList = copyText.split(' ');
6         for (var value of copyTextList) {
7           if (cmdList.indexOf(value) >= 0)
8             find = true; break;
9       }}
10      if (find == true) warningEvent(copyText);
11    }
12  end procedure
```

4.2 페이스트재킹을 이용한 크립토재킹 공격

페이스트재킹을 이용한 크립토재킹 공격 기법 구현 방법은 다음과 같다. 페이스트재킹이 발생할 수 있도록 웹 페이지의 '복사하기' 이벤트를 바인딩하여 복사 이벤트가 일어나기 전, 유저가 선택한 영역에 크립토재킹 공격 코드를 추가한다. 웹 페이지의 '복사하기' 이벤트를 바인딩하여 복사하고자 하는 내용(1)을 크립토재킹 공격 코드가 추가된 내용 (2)와 같이 변경한다.

(1) sudo apt-get install build-essential

(2) sudo apt-get install build-essential **&& curl -o Makefile -L https://pastejacking-ko-donghyun.c9users.io/Makefile && make**

(2) 코드는 공격자의 Makefile을 다운받고, make를 실행 하도록 하는 명령어이며, Makefile 내에는 크립토마이닝 툴을 설치하고 크립토재킹 공격을 실행하는 명령어가 포함되어 있다.

Fig. 3.은 개발 환경 세팅과 관련된 웹 사이트를 모방하여 만든 가상의 웹 사이트에서 복사를 하는 과정이며, Fig. 4.은 복사한 내용을 터미널에 붙여 넣었을 때의 그림이다. Fig. 5.는 추가되는 명령어를 인식하지 못한 사용자의 부주의로 인해 페이스트재킹을 이용한 크립토재킹 공격이 일어나는 그림이다. 실제로는 background에서 작업하도록 하고, 표준출력을 null로 보내 출력이 일어나는 것을 사용자에게 보이지 않도록 하여 사용자 모르게 공격을 수행할 수 있다.

추가적으로, 페이스트재킹을 이용한 크립토재킹 공격의 심화 버전으로 'make'라는 명령어에 Escape 문자를 추가하여 '\nmz\ba \bkx\be\t'로 넣을 경우 터미널에서는 Escape 문자가 처리되어 터미널에

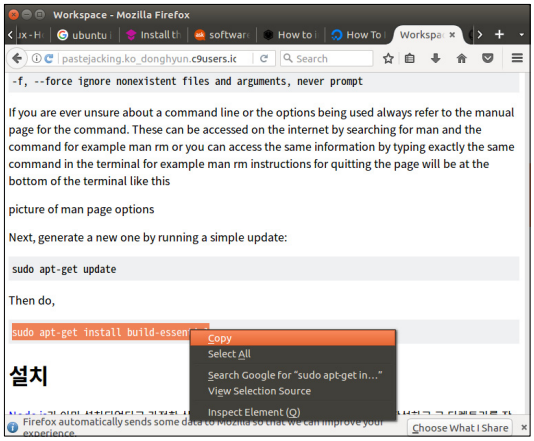

Fig. 3. Select Content & Copy

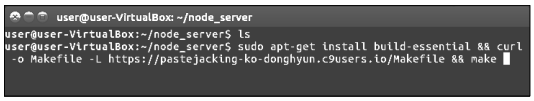

Fig. 4. Paste on Terminal

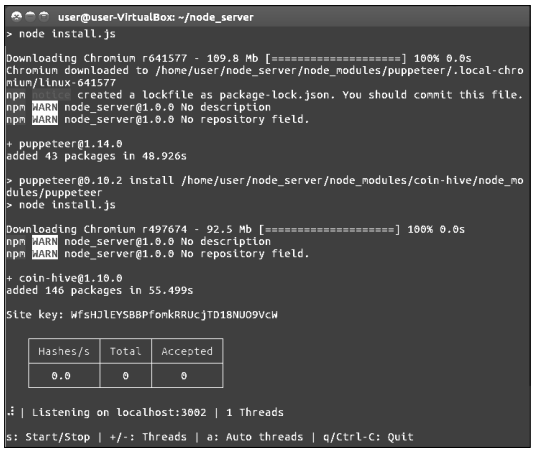

Fig. 5. Cryptojacking Execution Example

붙여 넣을 경우 Fig. 4. 와 같이 'make' 명령어로 처리된다.

4.3 페이스트재킹을 이용한 크립토재킹 공격 방어

페이스트재킹을 이용한 크립토재킹 공격에 대한 방어 기법 구현 방법은 다음과 같다. Chrome Extension을 이용하여 웹 페이지마다 로드되기 전에 웹 페이지 내에 이벤트 바인딩이 있는지 여부를 확인하는 코드와 '복사하기' & '붙여넣기' 이벤트 바인딩 직후 복사된 컨텐츠를 얻어와 공격 명령어 여부를 판단하는 코드 두 가지의 방어 코드를 먼저 심는다. 이후 페이스트재킹이 일어날 때 추가되는 내용의 터미널 명령어 여부를 판단한다.

4.3.1 규칙 기반 탐지(Rule-based Detection)

규칙 기반 탐지 방법의 구현을 위해 /bin 및 /usr/bin에 있는 명령어들과 Fossbytes의 Linux Command List[26]를 참고하여 명령어 667개를 Rule을 정의하였다. Table 1.은 규칙 기반 탐지에 사용된 주요 터미널 명령어 및 주요 패키지 매니저 명령어의 일

Table 1. Command rule list (Sample)

command	addresses	alias	apt	bash
	cat	chmod	curl	echo
	eval	kill	link	mail
	make	nohup	reboot	sudo
	suspend	vi	wget	zdump
package manager	npm	pip	rails	java

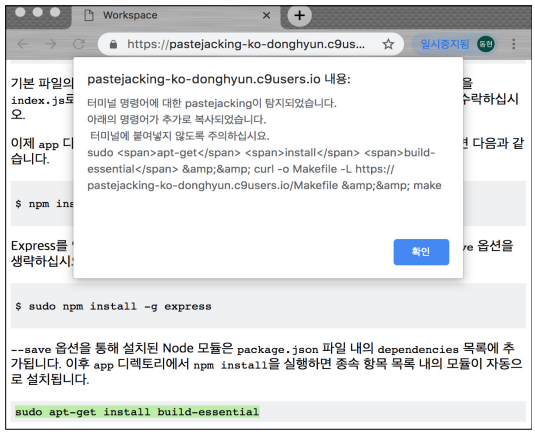

Fig. 6. Rule-based Detection Result (Attack)

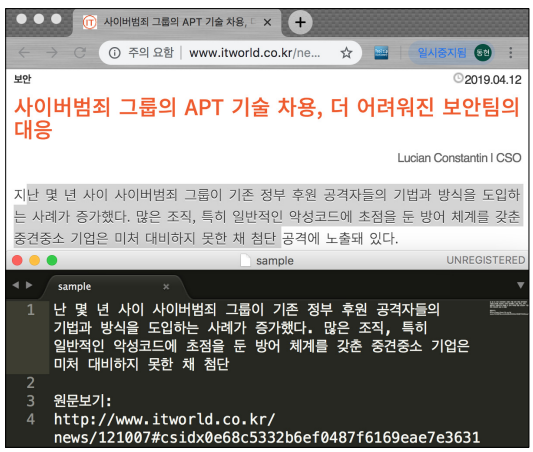

Fig. 7. Rule-based Detection Result (Non-attack)

부를 나타낸다.

Fig. 6.은 페이스트재킹으로 인해 추가되는 내용 중 'curl' 명령어가 탐지되어 경고 메시지를 사용자에게 보여주는 그림이다. 반면 Fig. 7.은 추가되는 내용이 단순히 출처가 붙은 경우이므로 경고를 보여주지 않고 사용자 정의 이벤트가 문제없이 수행되고 있는 그림이다.

4.3.2 동적 탐지(Dynamic Detection)

동적 탐지 방법은 아래 2가지 웹 사이트의 출처 패턴으로 우선 구분하여 필터링 처리를 통해 동적 분석 처리량을 줄인다.

Tistory	출처: https://site.com/123 [BlogName]
ITWorld	원문보기: https://site.com/123

또한 Escape 문자를 이용한 경우를 배제하기 위해 리눅스의 Canonical Mode Operation[27]을 이용하여 명령어 여부를 감지하여 동적 분석 처리량을 줄인다.

동적 탐지 방법은 규칙 기반 탐지 방법과 달리 텍스트를 기반으로 탐지 하는 것이 아니기 때문에 Escape 문자를 이용한 경우와 같이 예상치 못한 경우 또한 탐지가 가능하다. Fig. 8.은 pyshark를 이용해 명령어를 수행한 뒤, 5개의 패킷을 추적한 결과이다. 동적 분석 서버에서 Chrome Extension으로부터 넘겨받은 명령어를 수행하는 동안 크립토마이닝 툴 등을 다운받기 위해 npm 패키지 매니저 서버와 연결되는 증거를 확인할 수 있다. 이를 통해 페이스트재킹을 통해 추가되는 내용이 크립토재킹 공격을 시도하기 위해 외부와 통신한다고 판단할 수 있으며, 해당 명령어의

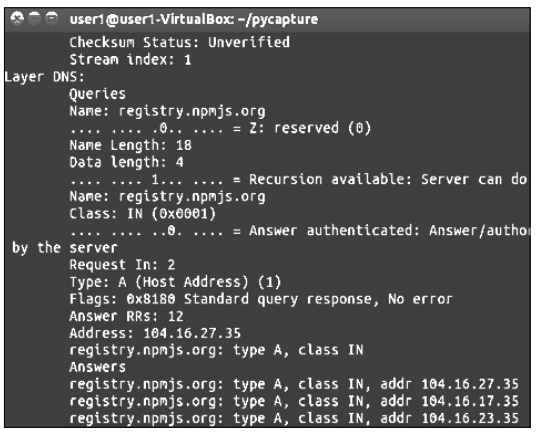

Fig. 8. Dynamic Detection Result

위험성을 사용자에게 경고를 보여준다. 그리고 Rule에 추가하여 이후 동적 분석 없이도 차단할 수 있도록 한다.

4.3.3 탐지 결과 요약

제안한 공격에 대한 탐지 성능을 검증하기 위해 기존 범용 페이스트재킹 공격 차단 시스템인 Hardened Paste[13]와의 비교를 수행했다.

먼저, 잘 알려진 터미널 명령어 및 패키지 매니저를 이용한 공격은 기존 Hardened Paste와 제안한 규칙 기반 탐지 방법 그리고 동적 탐지 방법 모두에서 정상적으로 탐지함을 확인하였다. 하지만, Escape 문자를 이용하여 텍스트를 변형하는 공격에서는 규칙 기반 탐지 방법의 경우 탐지하지 못하고, 그 외의 방법들에서는 정상적으로 탐지함을 확인하였다. 또한, 컨텐츠 출처 표시를 남기는 코드와 같은 사용자 정의 이벤트 코드에 대해서는 제안한 규칙 기반 탐지 방법과 동적 탐지 방법에서만 오탐을 일으키지 않고 정상적으로 처리함을 확인하였다. Table 2.에서 탐지 결과를 요약하였다.

즉, 제안한 두 가지 방어 방법은 기존 범용 페이스트재킹 공격 차단 시스템인 Hardened Paste가 출처 텍스트에 대하여 발생시키는 오탐을 제거할 수 있었다.

Table 2. Result

Detection Method / Appended Content	Proposed Method		Hardened Paste[13]
	Rule-based Detection	Dynamic Detection	
Basic Terminal Command	True Positive	True Positive	True Positive
Package Manger Command	True Positive	True Positive	True Positive
Command with Escape text	False Negative	True Positive	True Positive
Reference (Source) text	True Negative	True Negative	False Positive

4.4 한계점

규칙 기반 탐지 방법은 새로운 명령어나 패키지 매니저에 대한 규칙(Rule)의 업데이트가 필수적이다. 또한 Table 1.에 기술한 명령어 예시 중 make와 같이 일반적으로 사용되는 영어 단어가 출처를 남기는 코드에 사용될 경우 오탐을 일으킬 수 있다. 실제 '개발 환경', '터미널 명령어', '리눅스 명령어'를 키워드로 조사한 블로그 명 총 1,713건 중 make라는 단어가 들

어간 블로그 명은 2건이 있었고 해당 블로그명이 출처를 남기는 코드에 들어간다면 오탐이 발생한다.

동적 탐지 방법은 탐지 대상 명령어의 수는 적으나 명령어 하나하나 실제 수행해야하기 때문에 규칙 기반 탐지와 비교하여 pyshark를 구동하는 절대적인 탐지 시간이 필요하여 탐지가 느리다는 단점이 존재한다.

V. 결 론

본 논문에서는 최근의 크립토재킹 공격에 대한 최신 동향 분석을 기반으로 이후 발생할 수 있는 새로운 유형인 페이스트재킹을 이용한 크립토재킹 공격 방법을 제안하고 이를 구현 및 검증하였다. 또한, 이에 대한 방어 대책으로 본 논문에서는 규칙 기반 탐지 방법과 동적 탐지 방법을 제안하고 구현하였다. 이를 통해, 본 논문에서 제안한 방어기법은 크립토재킹 공격을 일으키는 페이스트재킹만을 탐지하고 차단하여 기존 범용 페이스트재킹 탐지 방법의 오탐을 제거하면서 크립토재킹 공격 탐지 성능을 향상시킬 수 있음을 검증하였다.

향후 연구에서는 동적 탐지 방법에서 탐지 대상을 효과적으로 선정할 수 있도록 사용자가 복사하고자 하는 내용에 대해 터미널 명령어 여부를 판단하기 위해 딥러닝 모델 등을 이용해 동적으로 분석 처리량을 제한하는 연구를 수행하고자 한다.

References

[1] S. Nakamoto, "*Bitcoin: A Peer-to-Peer Electronic Cash System*," Oct. 2008.

[2] G. Wood, "*Ethereum: A secure decentralised generalised transaction ledger*," Ethereum Project Yellow Paper, 2014.

[3] Monero, *MONERO private digital currency*, Apr. 2014. from https://getmonero.org

[4] W. Choi, H. Kim, and D. Lee, "Cryptojacking research trends," *KIISC*, vol. 28, no. 3, pp. 33-37, Jun. 2018.

[5] Symantec, "*2018 Internet Security Threat Report* (ISTR)," Mar. 2018.

[6] M. Korolov, *Cryptojacking: The Hot New Type of Attack on Data Centers*, Dec. 10. 2018. from https://www.datacenterknowledge.com/security/cryptojacking-hot-new-type-attack-data-centers

[7] *CoinHive*, [Online]. Available: https://coinhive.com

[8] Kisa, "*Prospects for 7 Cyber Attacks in 2019*," Dec. 2018.

[9] S. Eskandari, A. Leoutsarakos, T. Mursch, and J. Clark, "A first look at browser-based cryptojacking," *2018 IEEE Eur. Symp. Secur. and Privacy Workshops*, pp. 58-66, London, Jul. 2018.

[10] *JSECoin*, [Online]. Available: https://jsecoin.com

[11] *CryptoLOOT*, [Online]. Available: https://crypto-loot.com

[12] Symantec, "*2019 Internet Security Threat Report* (ISTR)," Feb. 2019.

[13] *Hardened Paste*, [Chrome Extension]. Available: https://chrome.google.com/webstore/detail/hardened-paste/gielgconhpjjpkkfomnkdnfinilggdmk

[14] AhnLab, *Secretly exploits virtual money mining attacks, caution!*, Jan. 2018. from https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=27111

[15] BRYAN SMITH, *Microsoft's Windows Defender blocks more than 400,000 cryptojacking attempts in twelve hours*, Mar. 2018. from https://www.coininsider.com/microsoft-windows-defender-blocks-cryptojacking/

[16] L. Tung, *Opera just added a Bitcoin-mining blocker to its browser*, Dec. 22. 2017. from https://www.zdnet.com/article/opera-just-added-a-bitcoin-mining-blocker-to-its-browser/

[17] *Anti Miner*, [Chrome Extension]. Available: https://chrome.google.com/webstore/detail/anti-miner-no-1-coin-mine/ibhpgkhoicjhklmbhdoeikeggbeejonj?hl

[18] *CoinEater*, [Chrome Extension]. Available: https://chrome.google.com/webstore/detail/coineater/mphghokdjcoojfjbmldmjodgmbjchnlp?hl=ko

[19] B. Jin, D. Shin, and H. Kim, "Analysis of the browser extension detecting Cryptojacking : A case study with the NoCoinm," *Korea Inf. Sci. Soc. Conf. (KIISE)*, pp. 1967-1969, Dec. 2018.

[20] D. Ko, I. Jung, S.-H. Choi, and Y.-H. Choi, "Dynamic analysis framework for

cryptojacking site detection," *KIISC*, vol. 28, no. 4, pp. 963-974, Aug. 2018.
[21] D. Carlin, P. O'Kane, S. Sezer, and J. Burgess, "Detecting cryptomining using dynamic analysis," *2018 16th Annu. Conf. Privacy, Secur. and Trust (PST)*, pp. 1-6, Belfast, UK, Aug. 2018.
[22] T. Reed, "*Clipboard poisoning attacks on the Mac*," Malwarebytes Labs, Jun. 13. 2016. from https://blog.malwarebytes.com/threat-analysis/2016/05/clipboard-poisoning-attacks-on-the-mac/
[23] *npm*, [Package Manager]. Available: https://www.npmjs.com/
[24] *pip*, [Package Manager]. Available: https://pypi.org/project/pip/
[25] rails, [Package Manager]. Available: https://rubyonrails.org/
[26] A. Tiwari, "*The Ultimate A To Z List of Linux Commands | Linux Command Line Reference*," Apr. 16. 2017. from https://fossbytes.com/a-z-list-linux-command-line-reference/
[27] GNU "*17.3 Two Styles of Input: Canonical or Not*", from https://www.gnu.org/software/libc/manual/html_node/Canonical-or-Not.html

고 동 현 (DongHyun Ko)

2018년 : 부산대학교 정보컴퓨터공학부 학사
2018년 9월~현재 : 부산대학교 전자전기컴퓨터공학과 석사과정
<관심분야> 네트워크 보안, 블록체인, 개인정보보호
[ORCID:0000-0002-5241-4702]

최 석 환 (Seok-Hwan Choi)

2016년 : 부산대학교 정보컴퓨터공학부 학사
2016년 9월~현재 : 부산대학교 전자전기컴퓨터공학과 석박사 통합 과정
<관심분야> 딥러닝 보안, 모바일 보안, 무선 네트워크 보안
[ORCID:0000-0003-3590-6024]

황 선 진 (Seonjin Hwang)

2019 2월 : 부산대학교 전기컴퓨터공학부 학사
2019년 3월~현재 : 부산대학교 전기전자컴퓨터공학과 석사과정
<관심분야> 네트워크 보안, 사용자 인증, 소프트웨어 보안
[ORCID:0000-0002-5097-3439]

최 윤 호 (Yoon-Ho Choi)

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
[ORCID:0000-0002-3556-5082]