암호화폐 채굴 악성코드 공격 시나리오와 대응 방안

박현성*

*순천향대학교 정보보호학과

Cryptocurrency Mining Malware Attack Scenarios and Countermeasures

Hyeon-Seong Park*

*Division of information security, Soonchunhyang University.

요 약

법정화폐와 같은 금전적 가치를 지닌 블록체인을 활용한 암호화폐의 등장으로 온라인상에서 거래되고 생성되는 가치에 대한 관심이 뜨겁다. 암호화폐 채굴 프로그램이나 거래 프로그램, 트레이딩 서비스 페이지, 거래소 등과 같은 암호화폐를 다루는 서비스에 대한 직접적인 공격과 일반적인 컴퓨터에 침입하여 CPU성능을 이용한 암호화폐 채굴도 등장하였다. 추가적인 2차 악성코드의 감염과 피해로 이어질 수 있다는 분석 결과가 공개되었으며 인터넷과 연결된 모든 기기에서 불법 채굴이 가능하다. 본 논문에서는 암호화폐 채굴 악성코드를 분석하고 공격 시나리오를 도출한다. 그리고 그에 따른 대응 방안을 제시한다.

I. 서론

악성코드의 진화와 더불어 악성코드의 주요 대상과 목적이 금전을 겨냥한 공격과 피해가 증가하고 있다. 과거 은행사이트나 트레이딩 프로그램을 대상으로 한 악성코드의 피해로 인한 파급력은 매우 컸다. 이와 비슷하게 최근 암호화폐 시장의 급격한 성장에 따라 암호화폐 거래 프로그램이나 거래홈페이지, 채굴 프로그램, 거래소를 대상으로 한 악성코드가 급증하였다. 쉽게 접하거나 다운받아서 이용할 수 있는 기술이기 때문에 각별한 주의가 필요하다. 하지만 많은 사용자들이 사용하고 있는 암호화폐 관련 프로그램들이나 거래소의 취약점이 드러나고 있는 상황이 발생하고 있으며, 대부분 오픈소스로 관리되고 있기 때문에 금전적 가치를 다루는 프로그램의 해킹으로 인한 피해 규모가 심각할 것으로 예상된다. 암호화폐가 더욱 대중화되고 많은 거래량과 채굴이 이루어지고 있는 시점에서, 이와 비례하여 발생하는 보안문제에 대한 이슈가 증가하였다. 암호화폐를 거래하거나 채굴하는 프로그램은 각 사용자의 컴퓨터를 사용하기 때문에 자체적인 높은 보안 수준이 요구된다.

본 논문의 목표를 위해 살펴본 공격으로는 다양한 암호화폐 프로그램으로의 접근뿐만 아니라 무작위의 사용자의 컴퓨터에 진입하여 악성코드를 사용하여 채굴을 하는 공격도 가능하다. 공격자는 금전적인 수익을 얻을 수 있으며, 피해자는 자신의 컴퓨터 자산이 도난당할 수 있다. 이에 따라 앞서 언급한 악성코드 프로그램이 개인 컴퓨터에 설치되었을 때 발생하는 보안위협과 공격 시나리오를 도출하고 실습을 통해 분석해본다.

본 논문의 구성은 다음과 같이 구성되어 있다. 2장에서는 관련 연구로 암호화폐 채굴 악성코드를 분석하고, 3장에서는 공격 시나리오를 통한 공격 가능성을 진단한다. 4장에서는 실습을 통해 공격 시나리오가 가능한지 살펴보고 5

장에서는 대응 방안을 도출한다. 마지막으로 6장에서 결론으로 마무리한다.

1.1 소문단

II. 관련 연구

2.1 암호화폐 채굴 악성코드 분석

국가 화폐는 중앙은행에서 발행된다면, 암호화폐는 채굴로 발행된다. 채굴은 블록체인의 문제를 맞힌 사용자에게 대가로 가상화폐를 지급하는 과정을 의미한다. 대부분의 암호화폐의 경우, 암호화된 문자열을 제시한다. 그리고 이를 가장 먼저 해독한 사용자에게 암호화폐를 제공한다. 예를 들어 비트코인의 채굴 과정은 다음과 같다.

1. 무작위 값을 가진 해쉬 값이 주어짐

2. 채굴자는 '무작위 넌스 대입법'으로 암호화 값을 해독함

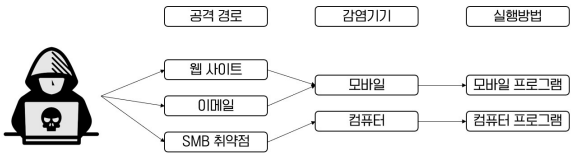
3. 채굴과 검증의 대가로 채굴자는 비트코인을 얻음

내용의 암호화 값을 해독하기 위해서는 수많은 연산을 통해 도출하는 방법밖에 존재하지 않는다. 결국 내용을 빠르게 입력하고 연산하는 사용자가 보상을 받게 된다. 즉, 암호화폐 채굴의 주요 핵심은 컴퓨터 성능에 달려있다. 이러한 이유로, 무작위 사용자의 컴퓨터를 이용하여 채굴을 하는 경우가 늘어나고 컴퓨터가 많을수록 성능이 좋아지기 때문에 채굴자는 점점 증가하고 있다.

가상화폐의 시세하락이 발생하였지만 여전히 높은 상황이기 때문에 '크립토재킹'이라는 악성코드가 등장하게 된 계기가 되었다. 해커는 크립토재킹으로 사용자의 개인기기를 채굴 장비로 사용할 수 있으며, 해커의 비용을 들이지 않고 남의 자산으로 채굴을 할 수 있는 것이다.

2.2 크립토재킹 원리

크립토재킹 실행 원리는 [그림 1]과 같다.



해커는 3가지 방법으로 크립토재킹을 전파한다. 해커는 '워터링 홀'을 이용하여 크립토재킹을 전파한다. 사이트에 악성코드를 몰래 심어서 방문하는 사용자를 감염시키는 방법으로 크립토재킹의 성공률을 높이기 위해 감염 사용자 수를 늘리기 위해 사용된다. 두 번째 방법으로 악성 이메일을 이용하는 것이다. 워드파일이나 한글 파일의 형태로 악성 매크로나 악성코드를 감추어 크립토재킹 설치파일을 유포하는 것이다. 피해기기에서 다운 후 실행하게 되면, 스토리지에 저장된다. 마지막으로 SMB(Server Messaging Block)를 이용하여 유포할 수 있다. SMB는 파일의 공유를 목적으로 만들어진 통신 프로토콜로 제로데이와 같은 형태로 공격이 이루어진다고 할 수 있다. 취약 기기가 발견되면 더블펄사를 이용하여 크립토재킹을 은닉시키게 된다.

감염되는 기기는 주로 컴퓨터이다. 그러나 최근 모바일의 성능이 향상됨에 따라 대상에 포함되어지고 있다. 아직 모바일 대상의 이례적 사례가 발견되지는 않았지만, 암호화폐 시장이 더욱 커진다면 충분히 일어날 수 있다. 실행방법은 두 가지로, 우선 감염기기에 보이지 않는 폴더를 생성하여 채굴 악성코드를 심은 후 채굴에 이용하는 방법이 있다. 또 자바스크립트 기반 사이트에서 채굴 악성코드를 다운시켜 채굴에 이용하게 할 수 있다. 애드웨어 형태로 보이는 경우가 많다.

크립토재킹은 실제로 사용자에게 위협적인 공격이나 직접적인 금전 피해를 가져다주지는 않는다. 사용자 기기 성능에 영향을 주기 때문이다. 하지만 크립토재킹을 통하여 추가적인 악성코드가 감염될 수 있다는 점이 더 중요할 것이다ㅣ.

III. 공격 시나리오

악성코드를 이용한 불법 암호화폐 채굴시나리오는 다음과 같다.

Step 1-1. 공격자는 암호화폐 불법 채굴 악성코드를 제작하여, 다양한 파일과 확장자에 은닉하여 배포를 준비한다.

Step 1-2. 적당한 타겟의 스피어피싱 또는 무작위 대상의 피싱 공격으로 많은 수의 기기들을 감염시킨다.

Step 1-3. 악성 홈페이지나 불법 다운로드로 위장한 홀을 통하여 악성코드를 전파한다.

Step 2. 감염된 기기에 실행 가능한 프로그램을 설치하거나 스크립트를 서버로부터 전송받고 파일을 은닉한다. 파일의 은닉 기법으로 스테가노그래피가 대표적인 예이다. 사용자가 주로 사용하는 확장자나 속임을 꾀하기 쉬운 파일로 위장하여 파일을 숨기는 것으로, EOL(End of Line)의 비트 부분을 변조하여 파일을 바이너리로 삽입할 수 있다. 또 가장 기본적인 방법으로 시스템폴더 또는 숨김폴더 및 파일을 이용하여 사용자가 쉽게 발견할 수 없도록한다.

Step 3. 유휴상태의 CPU나 GPU를 사용하여 암호화폐를 채굴하게 된다. 채굴을 통해 얻은 보상은 공격자의 지갑으로 전송된다.

Step 4. 표시되는 CPU 작업량을 조작하거나 프로세스를 은닉하여 감염된 기기의 이용자가 쉽게 알 수 없도록 조작한다. 악성코드는 해당 기기에서 작업관리자를 실행하거나 프로세스 검색을 수행하면 기능을 종료하거나 정보를 삭제한다.

Step 5. 네트워크 패턴 인식을 피하기 위해 비주기적으로 동작하여 악성코드를 은닉한다.

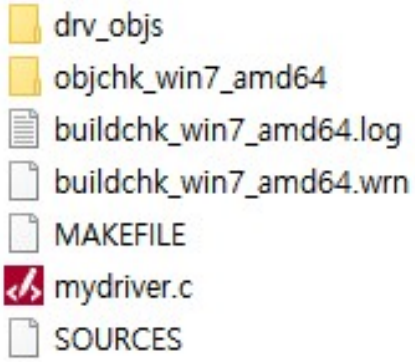
Step 6. 감염된 기기의 인터넷이나 이동식 저장매체를 통해 다른 기기로 악성코드를 전파한다.

IV. 공격 결과

공격의 기초를 먼저 다지기 위해 [그림 2]와 같이 코드를 작성하였다. WDK를 사용하여 컴파일 하였으며 컴파일 후 모습은 [그림 3]과 같다.

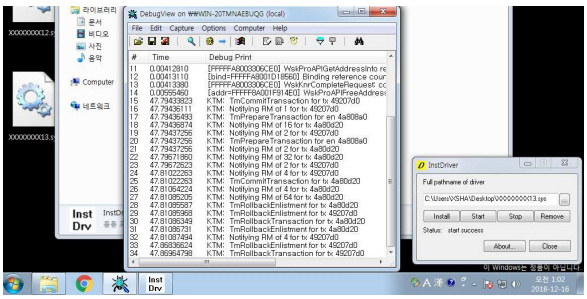
```
#include <ntddk.h>

DRIVER_INITIALIZE DriverEntry;
DRIVER_DISPATCH OnStubDispatch;
DRIVER_UNLOAD OnUnload;

NTSTATUS
OnStubDispatch(IN PDEVICE_OBJECT DeviceObject, IN PIRP Irp)
{
   Irp->IoStatus.Status = STATUS_SUCCESS;
   IoCompleteRequest(Irp, IO_NO_INCREMENT);
   return STATUS_SUCCESS;
}

// unload 함수
void OnUnload(IN PDRIVER_OBJECT DriverObject)
{
   DbgPrint("OnUnload called\n");
}

NTSTATUS DriverEntry(IN PDRIVER_OBJECT theDriverObject, IN PUNICODE_STRING theRegistryPath)
{
   DbgPrint("Start Load Driver\n");
   // DriverObject의 언로드 포인터를 세팅한다.
   theDriverObject->DriverUnload = OnUnload;
   return STATUS_SUCCESS;
}
```

[그림 2] 기본 드라이버 코드 작성 화면



[그림 3] 컴파일 후 파일 생성 모습

DebugView 프로그램을 통해 디버그 프린트를 확인하였으나 오류가 발생하였다. 그러나 다른 VM환경을 설정했을 때는 정상적인 “Start Load Driver”메시지가 출력되었다.

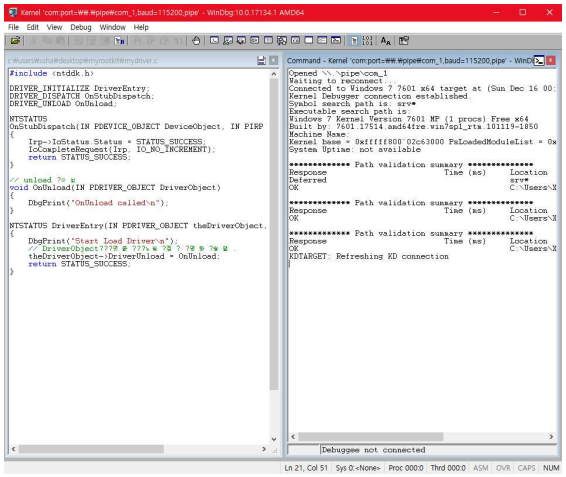


[그림 4] DebugView

WinDbg를 통해 드라이버 로드를 진행할 VMware와 연결한다. 초반 연결은 잘 되었으나 후에 VM과 WinDbg가 연결되지않아 더 이상의 진행이 불가능하였다(Fail to read system\currentcontrolset\services\Lmhosts\Par

ameters\EnableUserMode, error=2).

[그림 5] WinDbg 대기 화면



[그림 6] 우 - 기본 드라이버 코드, 좌 - Vmware와 연결 중 화면

V. 대응 방안

5.1 네트워크 트래픽과 패턴분석을 이용한 대응 방안

크립토재킹의 증가에 따라 다양한 네트워크 보안 관련 솔루션 공급업체에서 네트워크 수준에서의 암호화폐 채굴 동작을 탐지하는 기술을 개발하고 있다. 이러한 기법을 실제로 적용하여 상용화가 진행된 상태이다. 모바일, 데스크톱, 임베디드 장치, IoT(Internet of Thing), PLC, 노트북 컴퓨터, 데이터 서버, 클라우드 서버 등 종류가 다양하기 때문이다. 또한 의도적인 행위와 의도적이지 않은 행위가 존재하기 때문에 실행 범위가 광범위하다.

크립토재킹 악성코드의 한 가지 공통점은 암호화폐를 채굴하기 위해서는 서버와 통신을 하고, 새로운 해시를 받거나, 계산된 해시를 다시 서버로 보내고 지갑으로 보상을 받아야한다. 결론적으로, 악의적인 암호화폐 채굴과 관련된 활동이나 악성코드를 탐지할 수 있는 효율적인 방법은 네트워크를 직접적으로 검사하여 패턴화하는 방법이다. 현재의 네트워크 트래픽에 의심스러운 동작이 있는지 모니터링을 통해 검사한다. 불행히도, 암호화폐 채굴 트래픽을 다른 프로토콜과 구별하기 어렵다는 문제가 존재한다. 메시지를 함축하거나 암호화하여 통신하는 등 다양한 기법으로 이를 감춘다. 이를 해결하기 위해서 머신러닝으로 네트워크를 통과하는 수많은 데이터에서 악의적인 패턴이나 의도적인 행위를 파악하는 방법으로 해결가능하다. 악성코드 개발자가 간격을 랜덤화하는 등 통신의 주기를 감추거나 변경하려는 시도를 할 수 있지만, 암호화폐 채굴은 주기적인 트래픽이 특징이다. 또 메시지 길이가 특이하다. 전송을 받는 트래픽의 데이터인 해시값은 길이가 다소 짧고, 검증과 마이닝을 완료한 후 전송하는 아웃바운드 트래픽은 인바운드 트래픽보다 더 길다. 최초 요청은 응답보다 짧고, 응답은 긴 것인데 일반적인 상황과 트래픽과는 다르게 보인다. 이러한 패턴과 기타 신호들을 결합하여 크립토재킹을 효율적으로 분류하는 소프트웨어가 등장하기도 했다.

5.2 엔드포인트 보안을 통한 대응 방안

크립토재킹을 탐지하는 방법 중 가장 효과적인 방법은 엔드포인트 보호다. 해커는 데이터 암호화와 일반 통신 채널이 아닌 숨겨진 통신 채널을 사용해 5.1에서 언급한 네트워크 보안체계를 뚫고 시스템에 접근할 수 있다. 때문에 가장 효과적인 탐지 방법은 엔드포인트를 직접 보호하는 방법이다. 엔드포인트를 직접 보호하면서 시스템의 변경 및 변동 사항을 모니터링할 수 있어야한다. 효율적인 수행으로 이어지지 않는다면 오히려 취약점이 발생될 수 있다. 또한 변경된 시스템과 사항에 대해 올바르게 승인된 변경이나 명령인지에 대한 여부를 확인할 수 있어야한다.

5.3 운영체제 업데이트

SMB 제로데이나 다양한 운영체제의 취약점을 이용한 악성코드이기 때문에, 항시 업데이트를 통해서 감염으로부터 예방해야한다.

5.4 메일 검토

의심되는 메일을 열지 않는 습관과 수칙이 중요한 악성코드 예방법이다. 출처를 알 수 없는 메일로 위장할 수도 있으며, 정확한 타겟을 가진 악성메일일 가능성을 배제할 수 없다. 항상 메일의 목적과 분류의 명확함을 통해 악성코드의 감염으로부터 벗어나야한다.

VI. 결론

본 논문에서는 암호화폐 채굴 악성코드를 이용한 공격 시나리오와 대응 방안을 살펴보았다.

개인 사용자에 설치된 프로그램이나 악의적인 사이트에서 발생할 수 있는 취약점이나 사이버 공격들을 완전히 없애는 것은 불가능할 것이다. 그렇기 때문에 주기적인 보안패치와 안전한 인터넷 접근을 통해 악성코드의 감염으로부터 안전하도록 조치해야할 것이며, 악성코드에 대응하기 위해 네트워크와 엔드포인트의 머신러닝 기반 패턴탐지나 안티바이러스 개발 등의 많은 연구가 필요하고 요구될 것으로 사료된다.

[참고문헌]

[1] 암호 화폐 채굴 악성코드를 탐지하고 방지하는 방법, http://www.itworld.co.kr/news/108845

[2] 루트킷 : 윈도우 커널 조작의 미학

[3] 루트킷을 이용하는 악성코드, NCSC-TR050024

[4] Rootkit and Kernel Intergrity Protection, KAIST CySec Lab

[5] OS커널의 태스크스위칭 설명, https://kldp.org/node/83749