MS Windows에서 JPEG 관련 버퍼 오버런의 취약성 분석

오현수°, 장혜영*, 조성제*, 김홍근**
*단국대 정보컴퓨터학부 컴퓨터과학및통계학과
**한국정보보호진흥원
pyxis3@hanafos.com°, {chenil, sjcho}@dankook.ac.kr, hgkim@kisa.or.kr

Analysis of a Buffer Overrun Vulnerability of JPEG on MS Windows

Hyunsoo Oh° Hye-Young Chang*, Seongje Cho*, Hong-Guen Kim**
*Division of Information and Computer Science, Dankook University
**Korea Information Security Agency

요 약

본 논문에서는 JPEG 파일의 구조를 먼저 살펴하고, MS Windows 운영체제 상에서 비정상적인 JPEG 파일을 접근(open)할 때 발생할 수 있는 버퍼 오버런 취약성(MS04-028)을 재연하여 분석한다. JPEG 파일의 헤더에 코멘트(comment) 부분이 있을 경우 길이 필드가 잘못되어 있고 JPEG 파일의 몸체에 쉘코드(cmd.exe) 생성부분을 가지고 있을 경우, 버퍼(heap) 오버런 공격이 발생되어 예기치 못한 결과들이 발생할 수 있다. 본 논문에서는 디버거(WinDBG) 및 역공학 도구(IDAPro)를 이용하여, 이러한 JPEG 파일 관련 취약성을 분석하면서 바이너리 코드만 주어진 경우의 취약성 분석 절차를 이해하고 보안 결함 부분을 추적하는 연구를 수행한다.

1. 서 론

일반적으로 공격자는 목표 시스템을 공격하기 위한 사전 단계로 해당 시스템의 취약성 정보를 수집하고 분석한다. 거의 모든 시스템이 여러 보안 취약성을 가지고 있으며, 이들 취약성으로 인해 다양한 공격들이 이루어지고 있다. 2004년 9월 마이크로소프트사 제품에서 비정상 JPEG 파일 처리 시의 취약점을 이용하여 임의권한을 획득하는 문제가 발생하였다. 또한 이 취약점을 이용하여 바이러스 공격이 발생되어 큰 문제가 되고 있다.

F-시큐어 안티바이러스 연구소장 미코 히포넌에 따르면 "일반 바이러스 백신 소프트웨어는 기본 값으로 JPEG 파일을 검사하지 않는다. 바이러스 백신 검사에서 JPEG를 찾는 설정으로 바꿀 수는 있지만 이것도 JPEG 파일의 확장자를 바꾸는 것만으로 간단하게 우회할 수 있게 된다"고 말했다. JPEG와 같은 형식이면서 확장자 이름이 다른 것은 '.icon', 'jpg2'등 약 11종류나 된다. 따라서 악성 JPEG 파일 검사가 한층 더 어려워질 수밖에 없다고 히포넌은 지적한다[1].

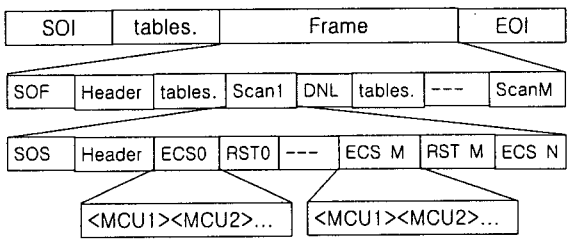
본 논문은, MS Windows XP 운영체제 상에서 콘솔 창을 새로 띄우는 보안에 취약한 JPEG 파일을 생성하고 이를 공격에 이용하여, 버퍼 오버런이 발생하는 과정을 역공학 기법을 통해 추적하고 체계적으로 분석한 내용을 기술한다. 분석된 결과를 이용하여 안전한 프로그램 및 데이터 파일 개발에 기여하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 JPEG 파일 구성 및 MS04-028 취약점에 대해 기술한다. 3장에서는 그 취약성을 재연시켜보고 분석한다. 4장에서는 대응 방법을 간단하게 기술하고, 5장에서 결론을 내린다.

2. JPEG의 구조 및 취약점 개요

2.1 JPEG의 구조

JPEG은 정지영상의 표준화를 위하여 국제전문가회의에서 정한 표준이며 정지영상이나 연속적인 톤의 이미지를 표현하기 위한 압축표준이다. 기본적으로 인간의 시각 시스템의 한계를 이용한 것으로 휘도 신호에는 민감하지만 색차 신호에는 둔감함을 이용한 압축 방법이다.



(그림 2-1) JPEG 파일 구조

JPEG는 Entropy-coded Segments와 Marker Segment로 구성되고 위의 (그림 2-1)과 같이 4 계층으로 나눌 수 있다. 각 층은 시작 마커 코드에 의해 분리된다. 모든 마커 코드는 2바이트로 표시되고 첫 번째 바이트는 0xFF, 두 번째 바이트에 코드를 삽입하게 된다. 그 구조는 아래의 (그림 2-2)와 같다[2]. 그림에서 정상적인 길이(length) 필드는 'Length + Data'의 바이트 길이를 나타낸다. 마커코드 중에서 문제될 소지가 있는 부분은 첫 계층의 tables 필드 내에 있는 코멘트(comment)의 길이 부분이다.

Marker code(2)	Length (2)	Data (max 65536)

(그림 2-2) JPEG 마커 코드 구조

2.2 MS04-028 취약점 개요

분석 대상 취약점 정보가 아래 [표 2-1]에 요약되어 있다 [3][4]. 취약한 GDI+ 라이브러리(gdiplus.dll)를 사용하는 응용 및 운영체제에서 이 취약점이 발생된다. MS Windows 소프트웨어 상에서 비정상 JPEG 파일을 악용하여 공격자가 관리자 권한을 획득할 수 있다. 이 경우 공격자는 프로그램 설치, 임의 데이터 보기, 변경 또는 삭제, 모든 권한을 가진 새 계정 만들기 등의 작업을 포함하여 해당 시스템을 완전히 제어할 수 있다.

[표 2-1] 취약점 개요

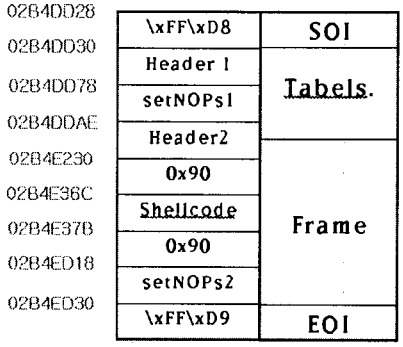
취약점 이름	취약점으로 인한 영향	공격 위험도
CAN-2004-0200	원격코드실행	긴급

JPEG 파일의 헤더에 ff fe 00 01 혹은 ff fe 00 00을 불임으로써 보안상 취약한 JPEG 파일을 만들게 된다. 왜냐하면 (그림 2-2)에서, 마커코드 ff fe 뒤에 길이를 나타내는 2 바이트가 정상 길이 값(최소 0x0002 이상)이 아닌 0x0001 혹은 0x0000값으로 설정될 경우 오버런이 발생하게 된다.[5]

3. 취약성 재연 및 취약점 분석

3.1 취약점 공격

실제 취약점을 재연하기 위해, WindowsXP 운영체제와 WinXP SP1(영문버전)을 설치하였다. 그 다음 (그림 3-1)과 같은 구조의 비정상 JPEG 파일을 만들었다. (그림 3-1)의 첫 세 부분(Header1, Set NOPs 1, Header2의 일부)은 (그림 2-1)의 tables 필드에 해당되며, 나머지 Header2의 일부 및 EOI(0xFFD9)를 제외한 부분은 (그림 2-1)의 Frame에 해당된다. 그림 왼쪽의 숫자는 메모리에 할당되었을 때의 주소이다. (그림 3-1)처럼 생성된 JPEG 파일을 접근하거나 클릭하면 cmd.exe(콘솔 창)이 실행된다.

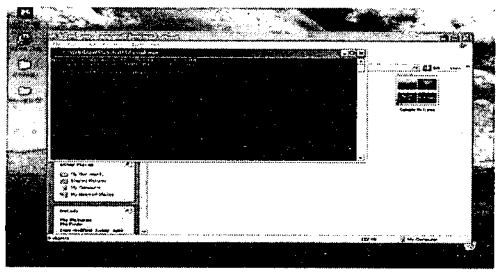


(그림 3-1) 공격 JPEG 파일의 구조

3.2 취약점 재연 및 분석

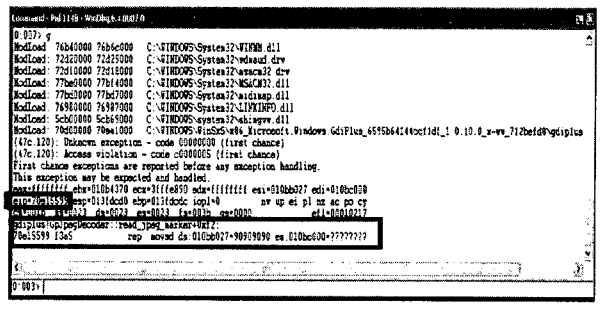
비정상 JPEG 관련 취약점을 분석하기 위해서는 SoftICE와 같은 동적분석 도구, IDA Pro와 같은 정적분석 도구, 윈도우 디버거(WinDBG), 비주얼 스튜디오 디버거 등을 사용하게 된다. 이들 도구 중 본 논문에서는 IDA Pro와 WinDBG를 사용하여 비정상 JPEG 파일 접근 시, 유발되는 예외상황을 통해 분석하였다.

우선 비정상 JPEG 파일을 실행하면 (그림 3-2)와 같이 cmd.exe가 실행되어 사용자(이 JPEG 파일의 접근자)의 권한으로 여러 가지 작업을 할 수 있게 된다. 이러한 이상 현상은 취약한 GDI+를 사용하는 모든 MS계열 소프트웨어들(Explorer, Office, SQL 등) 상에서 발생될 수 있다.



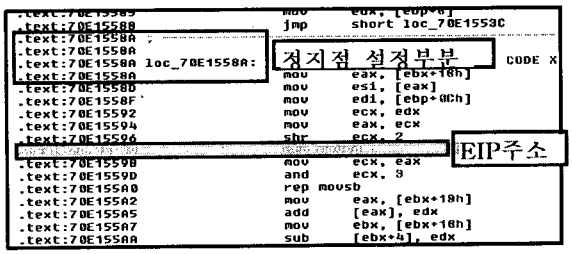
(그림 3-2) 취약한 JPEG 파일 공격 결과

본 논문에서는, 취약점을 분석하기 위해 비정상 JPEG 파일을 열어볼 Explore를 띄운 후 WinDBG을 attach시킨다. 그리고 Explore로 JPEG 파일을 열면 (그림 3-3)과 같이 WinDBG가 동작하게 된다. 이때 예외상황이 발생하게 되고 WinDBG를 통해 해당 EIP 주소가 70E15599이고, 이 주소는 Gdiplus.dll 라이브러리 내의 GpJpegDecoder 중 read_jpeg_marker 함수에 포함됨을 알 수 있다.



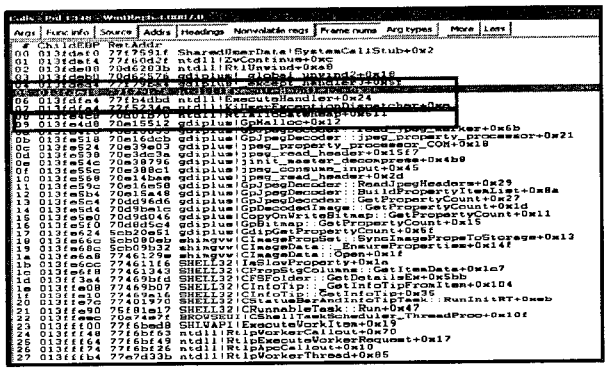
(그림 3-3) WinDBG를 이용한 EIP주소

다음으로, IDA-Pro를 사용하여 디버거에서 사용할 정지점 위치를 확인한다. 정지점은 예외상황이 발생한 EIP보다 앞쪽에 설정되어야 하기 때문에 (그림 3-4)와 같이 정지점을 77E1558A에 설정하여 다시 실험하였다.



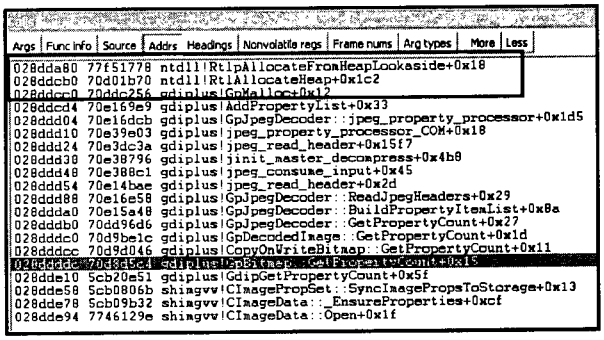
(그림 3-4) IDA-Pro를 이용한 정지점 추측

그리고 WinDBG에서 Trace하여 콘솔 창이 열리기 전까지의 콜 스택을 살펴보면 (그림 3-5)와 같다.



(그림 3-5) 비정상 JPEG 파일의 콜 스택

(그림 3-5)가 정상적인 JPEG 파일이 접근될 때의 콜 스택과 어떻게 다른지를 확인하기 위해서 (그림 3-6)과 같이 정상적인 JPEG 파일을 만들어 콜 스택을 확인해 보았다. 정상적인 JPEG 파일이란, 헤더의 코멘트 부분에 있는 길이 값을 제대로 설정해 놓은 파일을 의미한다.



(그림 3-6) 정상적인 JPEG파일의 스택 콜

(그림 3-5)와 (그림 3-6)을 비교해 보면 둘 다 JPEG파일의 헤더를 읽은 후 Malloc한다. 이때, 비정상 JPEG 파일의 콜 스택은 정상적인 JPEG 파일의 콜 스택과 달리 Malloc한 후 예외상황(Exception)을 발생시킴을 확인할 수 있다. 즉, 예외상황 처리 중에 실행(제어) 주소가 JPEG 파일 내에 있는 cmd.exe 실행 부로 변경되어 콘솔 창이 열림을 확인할 수 있다.

4. 대응 방법

이 문제는 2가지로 해결될 수 있다. 첫째는 JPEG 파일의 코멘트 부분에 있는 길이 값을 조사하여 정상이 아닐 경우 해당 프로그램을 즉시 종료하는 방안이고, 둘째는 쉘 코드에서 system() 함수 주소를 이용하므로 운영체제에서 system()의 주소를 숨기는 방안이다. 마이크로소프트사에서는 이 문제를 서비스팩 2를 출시함으로써 해결하였다. XP 서비스팩 2에서는 PEB Randomization, Heap Header cookie, Safe unlinking을 통해 힙 영역의 취약점을 보완하였다.

5. 결론 및 향후 연구과제

비정상 JPEG 관련 버퍼 오버런으로 인한 쉘코드 실행 문제는 최대 위험등급이므로 그 원인과 대책을 체계적으로 분석할 필요가 있다. 본 논문에서는 문서 MS04-028을 기반으로 운영체제 WindowsXP에서 비정상적 JPEG 파일 처리 시 발생하는 오버런 취약점을 역공학을 적용하여 재연하고 분석하였다. 본 연구의 결과는 바이너리 코드만 주어진 상황에서 소프트웨어 및 데이터 취약점을 추적하고 그 원인을 분석하는데 유용하게 사용될 수 있다.

6. 참고문헌

[1] http://www.zdnet.co.kr/news/internet/0,3902414,39130648,00.htm

[2] 허영민, "JPEG 압축원리 이해와 파일분석 및 문제해결", 한빛미디어

[3] http://www.microsoft.com/technet/security/bulletin/MS04-028.mspx

[4] http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0200

[5] http://www.securityfocus.com/bid/11173/exploit