기술문서	10. 7. 19. 작성

smpCTF 2010 Challenge 5 Writeup

by 6l4ck3y3
6l4ck3y3@gmail.com

Abstract

smpCTF 2010 예선에서 출제된 Challenge 5번의 풀이입니다. 다양한 압축 알고리즘, 파티션 이미지, 파일 헤더 분석, pcap 파일 분석, 게다가 스테가노그라피까지 포렌식의 다양한 형태를 하나로 뭉쳐놓은 문제입니다. 난이도 자체는 쉽지만 좋은 문제입니다.

* 본 문서는 대학정보보호동아리연합회(http://www.kucis.org)의 지원을 얻어 작성된 문서입니다.

Table of Content

Problem

We are sure we left, a flag in here somewhere... Right redsand?

Can you help find it? The file: download

이 문제는 주어진 파일 내에서 flag 값을 찾는 포렌식 유형입니다. 이런 경우, 보통 파티션 이미지나 그림 파일, 문서 파일, pcap 파일 등이 주어집니다.

파티션 이미지라면 flag 가 숨겨진 파일을 찾아야 하고, 파일이 주어졌다면 해당 포맷에 대한 이해를 바탕으로 flag 를 찾아야 합니다. 문제가 쉽게 나온다면 간단히 툴을 써서 쉽게 풀 수 있지만, 어렵게 꼬아서 낸다면 정말 어려워질 수 있는 유형이 포렌식이죠.

Analysis & Solution

Decopress with Various Libraries

```
$ file forensic1-image
forensic1-image: rzip compressed data - version 2.1 (15185973 bytes)
$ sudo apt-get install rzip
$ mv forensic1-image forensic1-image.rz
$ rzip -d forensic1-image.rz -o forensic1-image
```

우선, 다운로드 받은 파일을 file 명령어로 어떤 파일인지 확인합니다. forensic1-image 파일은 rzip 으로 압축된 파일입니다. rzip 은 큰 파일을 처리하기 위한 압축 알고리즘인데 rzip 을 설치해서 압축을 풀 수 있습니다.

```
$ file forensic1-image
```

```
forensic1-image: LHarc 1.x/ARX archive data [lh0]
$ sudo apt-get install lha
$ mv forensic1-image forensic1-image.lha
$ lha -x forensic1-image.lha
FS.tar    - Melted   :
ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
```

rzip 으로 압축을 풀면 또 압축 파일이 나옵니다. 이번에는 LHarc 파일인데 일본에서 만들어진 포맷이라고 합니다. lha 를 설치해서 압축을 풉니다.

```
$ file FS.tar
FS.tar: POSIX tar archive (GNU)
$ tar -xvf FS.tar
FS
```

이번에는 tar 파일이 나왔습니다. tar 는 Unix 시절부터 쓰인 워낙 유명한 압축 파일 포맷이죠. 역시 tar 로 압축을 풉니다.

```
$ file FS
FS: bzip2 compressed data, block size = 900k
$ mv FS FS.bz2
$ bzip2 -d FS.bz2
```

압축을 풀면 FS 라는 이름의 파일이 나옵니다. 이 파일은 bzip2 로 압축된 파일이고 bzip2 로 압축을 풀 수 있습니다.

```
$ file FS
FS: gzip compressed data, was "FS", from Unix, last modified: Wed Jun 30 10:42:18
2010, max compression
$ mv FS FS.gz
$ gzip -d FS.gz
```

또 압축 파일이 나왔습니다. 이번엔 GNU zip 으로 압축된 파일입니다. gzip 으로 압축을 풀어줍니다.

Analysis a Partition Image

```
$ file FS
FS: Linux rev 1.0 ext2 filesystem data,
UUID=c8a4643d-d89b-43db-bae8-6192db41dcc1 (large files)
```

Linux 에서 쓰이는 EXT2 파티션 이미지가 나왔습니다. 파티션(볼륨) 이미지를 분석해주는 툴은 많습니다. Encase 가 가장 유명하고 리눅스의 Autopsy 역시 많이 이용됩니다. 그리고 Windows 에서 쓸 수 있는 FTK Imager 역시 훌륭한 툴입니다.

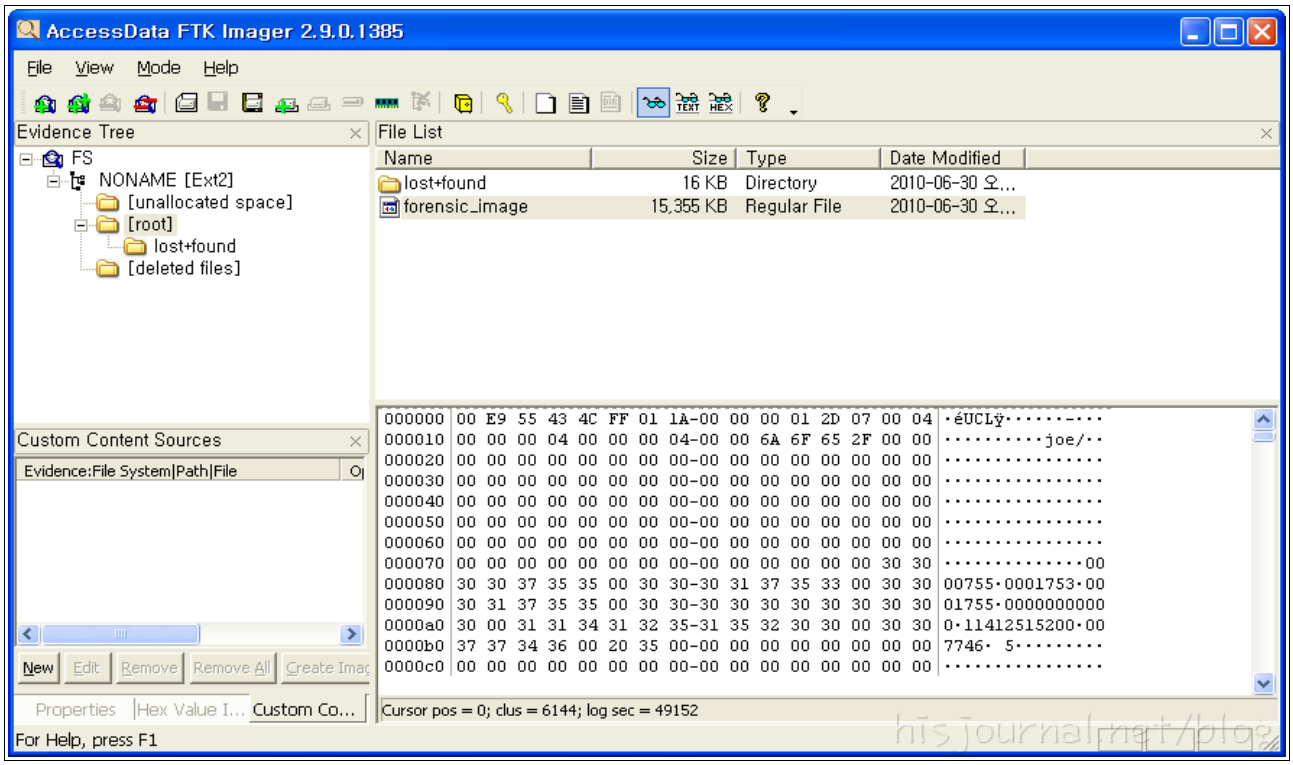

그림 1: FTK Imager 로 열어본 FS

이 파티션 이미지를 FTK Imager 로 열면 [그림 1] 과 같이 forensic_image 라는 파일 하나만 딸랑 존재합니다. 그 외엔 삭제되어서 그런지 할당되지 않은 영역이 존재하긴 한데, 거기서는 특별한 점을 찾을 수 없습니다. 대신 forensic_image 파일의 헤더를 보면 UCL 이라고 쓰여 있습니다. 검색해보면 압축

파일임을 알 수 있습니다.

FTK Imager 의 파일 추출 기능을 이용해서 forensic_image 파일을 뽑아냅니다. 그리고 UCL 라이브러리 소스를 다운로드 받아서 컴파일 하고, 예제로 있는 uclpack.c 도 컴파일합니다.

```
$ ./uclpack -d forensic_image out

UCL data compression library (v1.03, Jul 20 2004).
Copyright (C) 1996-2004 Markus Franz Xaver Johannes Oberhumer
http://www.oberhumer.com/opensource/ucl/

uclpack: block-size is 262144 bytes
uclpack: decompressed 15723366 into 31989760 bytes
$ file out
out: POSIX tar archive (GNU)
$ mv out out.tar
$ tar -xvf out.tar
```

uclpack 으로 압축을 풀면 또 tar 압축 파일이 나옵니다. tar 로 압축을 풉니다.

```
$ ls joe
Desktop          JoeHackerPrivate.gpg  Templates          network_sniff.pcap
Documents        Music                 Videos             scans
Downloads        Pictures              examples.desktop
Joe Hacker.asc   Public                gppg-stuff.txt
```

압축을 풀면 joe 라는 이름의 리눅스 홈폴더가 나타납니다. 이런 포렌식 유형은 flag 가 숨겨진 파일을 찾아야 합니다. 여기서 가장 의심되는 파일은 단연코 network_sniff.pcap 파일이겠네요. 파일 이름도 의심스럽고 pcap 파일이 많은 정보를 가지기 때문이기도 하구요.

Analysis a PCAP file

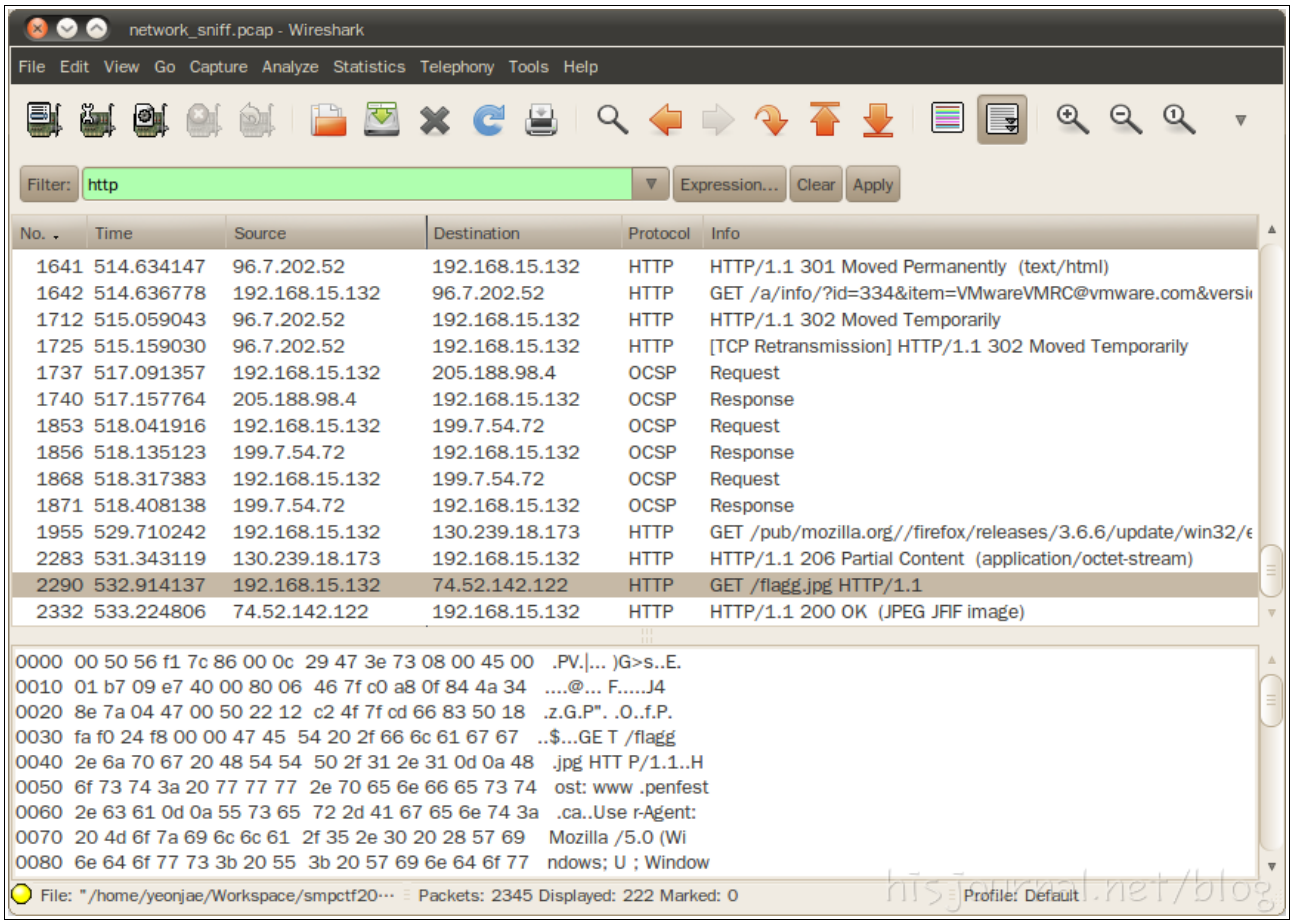

그림 2: Wireshark 로 열어본 network_sniff.pcap

Wireshark 로 network_sniff.pcap 파일을 열면 TCP 통신을 한 것을 알 수 있습니다. 그리고 조금 더 자세히 보면 HTTP 라는 사실도 알 수 있죠. 즉, 웹에서 뭔가를 하였고, 그것이 캡쳐되었다는 얘기지요. Filter 에 http 를 입력해서 웹에서 무엇을 했는지를 확인합니다. 그러면 [그림 2] 처럼 마지막에 flagg.jpg 파일을 요청하고 200 OK 메시지와 함께 수신이 완료되었다는 사실을 확인할 수 있습니다.

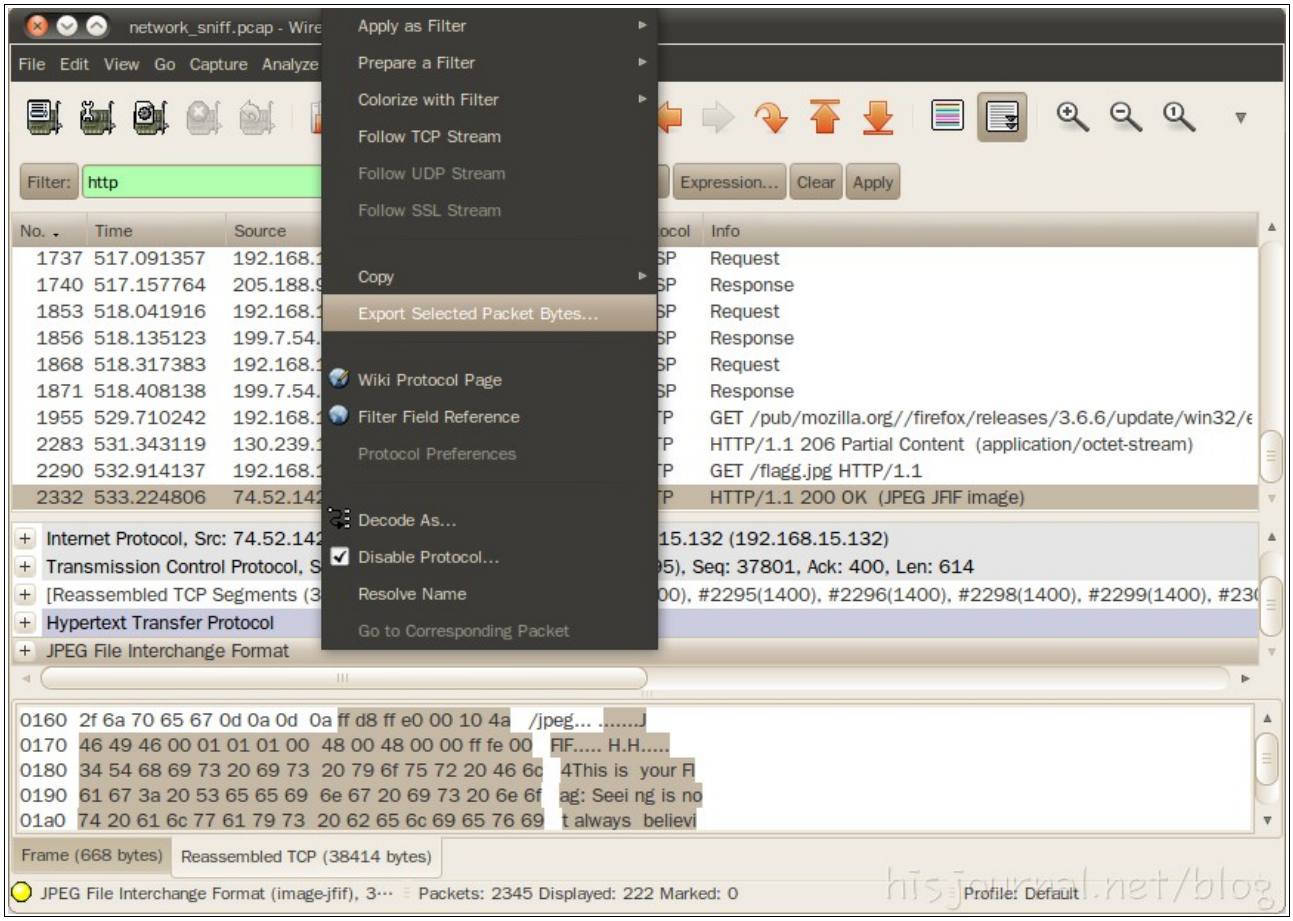

그림 3: Export "flagg.jpg"

[그림 3] 과 같이 flagg.jpg 의 요청에 대한 200 OK 메시지의 패킷을 선택하고, “JPEG FILE Interchange Format” 에서 오른쪽 마우스 버튼을 클릭합니다. 그리고 “Export Selected Packet Bytes...” 를 선택하여 파일을 추출합니다.

Steganography

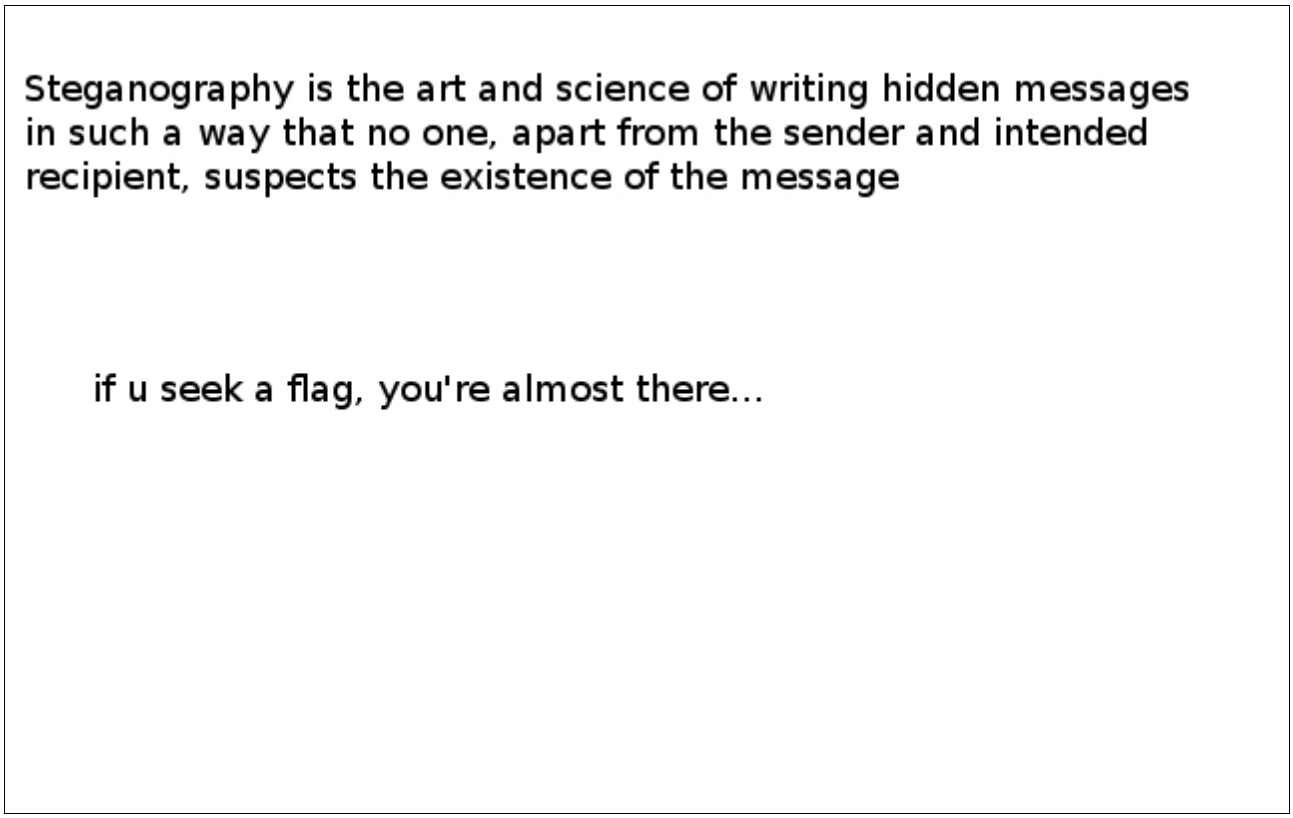

그림 4: flagg.jps

pcap 파일에서 추출한 flagg.jpg 은 [그림 4] 와 같습니다. flag 를 숨겨놓았고, 거의 다 찾아왔다는 메시지입니다. 결국은 그림 파일에 대한 스테가노그라피까지 왔네요.

```
$ strings flagg.jpg | head
JFIF
4This is your Flag: Seeing is not always believing!
!(7#159EWe
")AHh
" ""
" ""
" ""
" ""
" ""
```

```
" ""
```

스테가노그라피의 가장 단순한 형태는 그림 파일의 포맷에 메시지를 교묘하게 숨기는 것입니다.
strings 명령어로 flagg.jpg 파일에서 문자열들을 뽑습니다. 그러면 숨겨진 flag 가 드러납니다.

Reference

rzip :: http://en.wikipedia.org/wiki/Rzip

LHA (file format) :: http://en.wikipedia.org/wiki/LHA_(file_format)

tar (file format) :: http://en.wikipedia.org/wiki/Tar_(file_format)

bzip2 :: http://en.wikipedia.org/wiki/Bzip2

gzip :: http://en.wikipedia.org/wiki/Gzip

ext2 :: http://en.wikipedia.org/wiki/Ext2

UCL :: http://www.oberhumer.com/opensource/ucl/

Steganography :: http://en.wikipedia.org/wiki/Steganography