2009년 데프콘 CTF 예선 Crypto Badness
100, 200, 300 문제 풀이

2009. 06

eits1st@tistory.com

사실 2009년 데프콘 CTF 예선에 참가한 것은 아니다. 참가하기엔 미비한 실력인건 물론이며, 다른 일로 인해 시간적 여유가 되지 않았다. 하지만 해당 문제를 지인을 통해 넘겨 받았고 이번 데프콘에 관심분야(?)였던 암호학 쪽으로 새롭게 문제가 출제가 되었다는 것을 알게 되었다. 이 문서에는 풀이법과 동시에 풀이를 하면서 겪었던 시행착오 등을 나열식으로 적어보았다.

- 문제풀이 -

[Crypto 100번]

Question)

ASI JL DUJZTED SA J EJZD JVV NBTODI, VDD FOD AHB VBFD: OFFT://YYY.AHB.MSK/TJMD2/CDN08/NSCD_122908.OFZE, PSQD GSW MWDVV? LS, JNFWJE AHB RWBX.

Solve)

문제를 보자마자 대치암호 또는 치환암호 일거라고 확신이 들었다. 위에 줄 보다 밑에 줄에 있는 문장이 URL을 나타낸다는 사실은 명확했으며 그로 인해 쉽게(?) 풀 수가 있었다.

URL은 http://www. 이런 식으로 시작이 된다는 사실로 미루어 봤을 때 주어진 문장을 다음에 나타나는 표처럼 문장을 치환 시킬 수 있었다.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
					t									h					p					w	

ASI JL DUJZpED SA J EJZD JVV NBphDI, VDD thD AHB VBtD:
http://www.AHB.MSK/pJMD2/CDN08/NSCD_122908.htZE, PSQD GSW MWDVV? LS, JNtWJE AHB RWBX.

26개의 알파벳 중 4개를 치환했다. 알파벳 A 부터 Z 까지 1번부터 번호를 부여한 후 이 치환된 문자의 간격을 계산 해보면 일정 하지 안다는 것을 알 수 있다. 즉, **대치암호는 아니라는 것**이다. 결국 모든 문자에 대비되는 문자를 찾아야만 한다=_=;

치환암호는 주어진 알파벳 문자의 특성상 문자열의 빈도수를 이용하여 풀이를 할 수가 있다. 하지만 주어진 문제를 문장을 빈도수를 바탕으로 측정하여 풀이를 하게 되면 꽤 험난한(?) 풀이과정을 겪게 될 수도 있다. 직감을 이용하여, 그리고 약간의 노하우(?)를 이용하여 계속 풀이 해 보았다.

- 두 번째 문장인 URL 문장을 계속 보게 되면 URL이 끝나는 부분에 확장자를 확인 할 수 있는데 ht??로 끝나는 걸 봐서 최종적으로 html 이라는 것을 눈치 챌 수 있다.
- 첫 번째 문장에 thD에서 the가 아닐까 하는 예상을 해볼 수 있다.
- 첫 번째 문장에서 단 한 글자로 구성된 J 는 영어문장에서 가장 많이 나오는 한 글자 a 로 예상 해볼 수 있다.

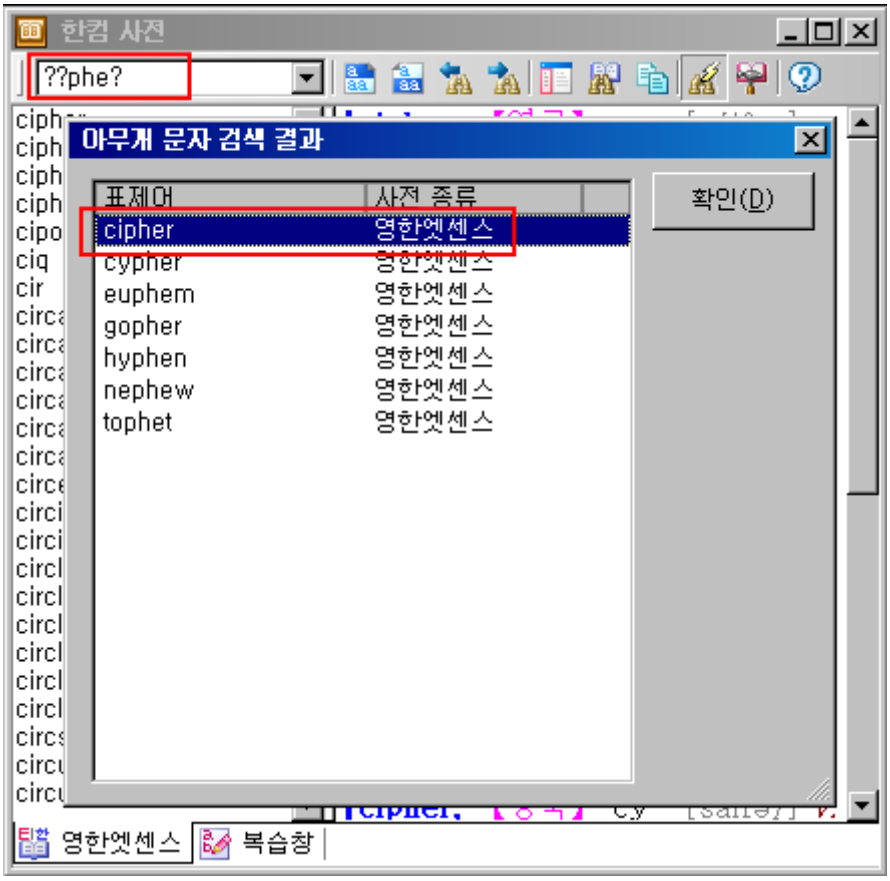
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
			e	l	t				a					h					p					w	m

ASI aL eUample SA a lame aVV NBpheI, Vee the AHB VBte:
http://www.AHB.MSK/paMe2/CeN08/NSCe_122908.**html**, PSQe GSW MWeVV? LS, aNtWal AHB RWBX.

다시 4개의 알파벳을 치환했다. 완성된 단어가 보이기도 하며, 특히 eUample 는 누가 봐도 example 이라는 것을 알 수 있다. 더불어 example 앞에 있는 aL 은 an 을 나타낸다는 사실도 알 수 있다.

이런 식으로 한 글자씩 유추하여 맞춰 갈 수 있는데 학창시절 암호학 수업을 들으며 치환암호 과제 할 때 써먹었던 방법을 이제 쓸려고 한다.(이것이 앞에서 말한 약간의 노하우(?)이다) 그건 바로 **한컴사전**을 이용하는 것이다. 한컴사전은 * 와 ? 문자를 와일드카드 값으로 인식하여 단어를 찾아주기 때문에 알파벳 수와 약간의 스펠링을 알고 있는 단어를 찾는데 도움이 된다.

문제로 주어진 문장에서 NBpheI 를 한컴사전으로 검색해보자. N, B, I는 우리가 알아야 할 알파벳이며 p ,h, e는 이미 알고 있는 알파벳이므로 ??phe?로 검색한다.



- **이런 식으로 한컴사전을 응용!!** -

한컴사전의 아무개 문자검색 결과에 나오는 ??phe? 에 해당하는 단어는 딱 7개 뿐이다. 그 중 cipher 이라는 단어가 왠지 눈에 들어오지 않는가? 암호 관련 문제이며, 컴퓨터와 밀접한 관련이 있는 cipher. 나는 여기서 과감히(?) N 을 c 로 B를 i 로 I 를 r 로 치환하였다.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	i		e	l	t			r	a		n		c	h					p	x				w	m

ASr an example SA a lame aVV cipher, Vee the AHi Vite:
http://www.AHi.MSK/**pa**Me2/C**e**c08/**c**SC**e**_122908.**html**, PSQ**e** GSW MW**e**VV? nS, **act**W**al** AHi RWiX.

actW**al** 은 actual 일 것이다. (한컴사전으로 act?al 을 확인하면 바로 나온다.) 이쯤에 다시 문장을 자세히 보면 첫 번째 문장의 V**ee the** AHi Vi**te**: 에서 AHi 부분이 URL 문장의 domain과 일치한다는 것을 알 수 있으며, 맨 마지막에도 한번 더 등장하는 단어라는 것을 알 수 있다. domain이라면 웹사이트의 이름일 것이다. 그리고 다시 V**ee the** AHi Vi**te** 를 보게 되면 See the ??i site 라는 문장을 유추해 낼 수 있다. 즉 V 는 s 가 되는 것이

다.

원래 점점 쉬워져야 되는데.. 어려워진다-_-; 계속 한컴사전으로 두들겨 보자. 두 번째 문장에서 **n**S 를 n?로 한컴에서 두들겨 보면 22개의 단어가 나온다. 하지만 정작 단어가 되는 것은 몇 개 없으며 주어진 문제의 앞뒤 정황을 봤을 때 **n**S는 no로 치환 할 수 있다. S를 o 로 치환하고 첫 번째 문장의 AS**r** 을 보면 for가 될 것이라는 것을 알 수 있다. 문장의 첫 단어이며 For example 이라는 숙어(?)도 있고 하니+_+;

다시 정리해보면 다음과 같다.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
f	i		e	l	t			r	a		n		c	h				o	p	x	s	u		w	m

fo**r an example o**f** a lame a**ss** cipher, **s**ee the **f**Hi **s**ite:
http://www.f Hi.M**o**K/**pa**M**e**2/C**ec**08/**co**C**e**_122908.**html**, P**o**Q**e** G**ou** M**ue**s**s**? **no**, **actua**l **f**Hi R**ui**X.

첫번째 문장은 알파벳 H 빼고 모두 치환이 되었다. 해석해보면 "**엉터리 바보 같은 암호문의 한가지 예로 fHi 사이트를 봐라**" 이렇게(?) 된다. 아직 URL 문장이 모두 치환이 되지 않아 웹사이트를 볼 수는 없으므로, 풀이를 계속 진행해 보자.

URL 문장의 중간에 M**ue**ss 는 추측이라는 뜻의 guess 를 나타내는 것이며 G**ou** 또한 you를 나타내는 것임을 알 수 있다. (이것 또한 한컴사전을 두들겨 보면 쉽게 알 수 있다.) M을 g로 치환 한 후 URL을 보면 **www.f**Hi.M**o**K 에서 Mok는 go?일 것이며 정부기관을 나타내는 도메인인 gov 임을 유추해 볼 수 있다. 계속 URL를 유심히 보자. 여기서 나는 파일명을 나타내는 마지막 부분인 **co**C**e**_122908.**html** 에서 **co**C**e** 부분에 C를 d로 생각 하여 후 전체 URL을 확인 했었고, URL의 느낌(?)상으로 C는 d로 치환 될 수 있다는 확신을 가지게 되었다. 정리하면 다음과 같다.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
f	i	d	e	l	t	y		r	a	v	n	g	c	h				o	p	x	s	u		w	m

for an example of a lame ass cipher, **see the fHi site**:
http://www.f Hi.**gov/pag**e2/**dec**08/**cod**e_122908.**html**, P**o**Q**e** **you guess**? **no**, **actual fHi** R**ui**X.

어느 정도 URL의 형태가 완성되었다. 아직 치환되지 않은 H는 남은 알파벳인 b, j, k, q, z 중에서 짝을 이루는 단어가 있을 것이다. 이를 조합하여 URL에 하나씩 접속해보면

http://www.fbi.gov/page2/dec08/code_122908.html 이라는 웹사이트에 접속 할 수 있다. FBI 라니.. 왜 알아채지 못했을까 하는 생각이 든다.
마지막으로 남은 알파벳은 j, k, q, z 이며 한컴 사전을 이용하여 P**o**Q**e** 를 ?o?e 로 검색, R**ui**X 를 ?ui? 로 검색하여 모든 알파벳을 다음과 같이 치환 할 수 있다.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
f	i	d	e	l	t	y	b	r	a	v	n	g	c	h	j	k	q	o	p	x	s	u	z	w	m

for an example of a lame ass cipher, **see the fbi site**:
http://www.fbi.gov/page2/**dec**08/**code**_122908.**html**, **joke you guess**? **no**, **actual fbi quiz**.

문장 완성 이후 나타나 있는 URL에 접속하니 비슷한 문제가 하나 더 있었으며, 치환되는 알파벳이 똑같으므로 쉽게 할 수 있다. 데프콘 CTF에 참여하지 않아 인증해야 하는 답이 어떤 것인지 알 수가 없지만 대충 정답(?), 혹은 정답에 근접한 문장은 나온 것 같다.

for an example of a lame ass cipher, see the fbi site:
http://www.fbi.gov/page2/dec08/code_122908.html, joke you guess? no, actual fbi quiz.

[Crypto 200번]

Question)

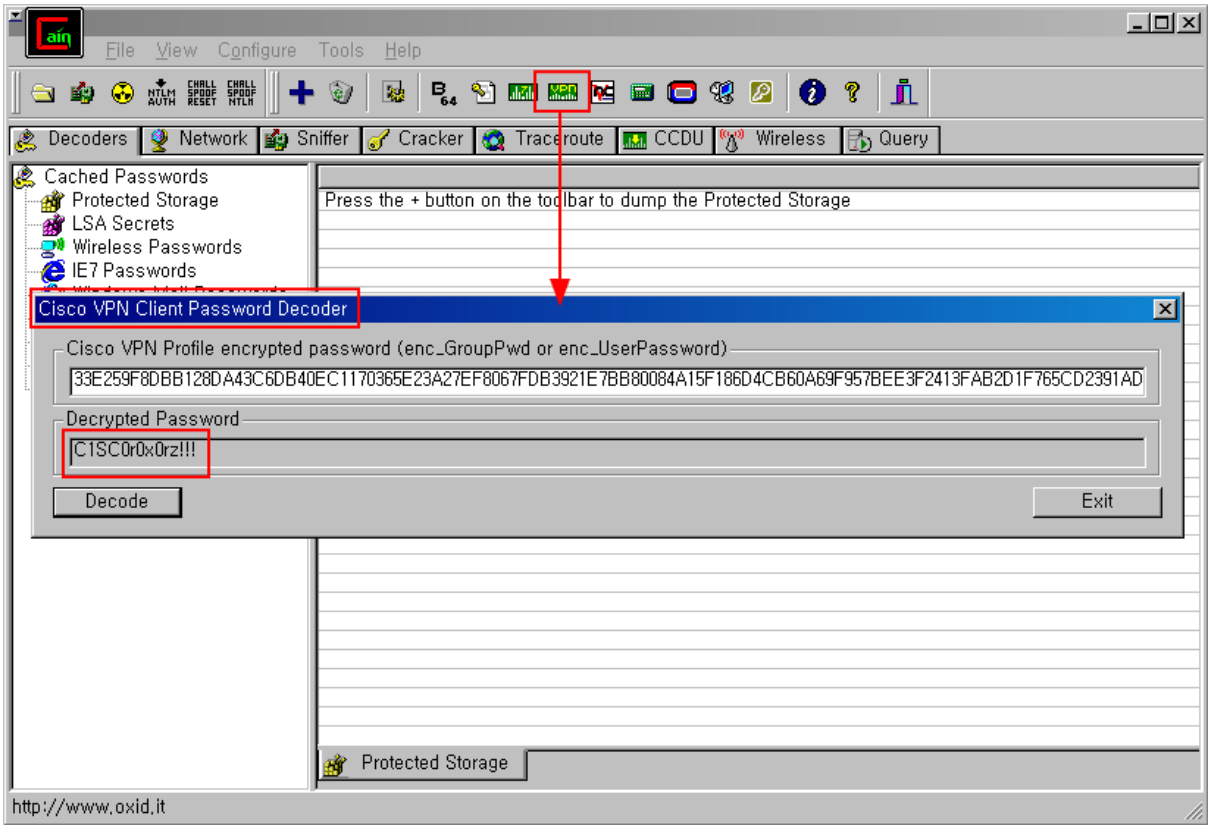
All your teal vpn are belong to us!

33E259F8DBB128DA43C6DB40EC1170365E23A27EF8067FDB3921E7BB80084A15F186D4CB60A69F957BEE3F2413FAB2D1F765CD2391AD065B

Solve)

처음 문제를 확인하고 어떠한 의미의 HEX 값이거나 혹은 어떠한 바이너리 HEX값의 일부분 일 것이라는 추측으로 처음 4byte를 google로 검색 해보는 등의 삽질을 많이 했다. 또한 teal VPN으로 검색을 해보기도 하고..

그러다가 툴로 크랙이 될려나 라는 생각에 Cain & Abel 프로그램으로 돌려본 결과, 패스워드를 알 수 있었다.



그 후 Cisco VPN Client Password 키워드로 검색으로 해보았고 몇몇 웹 사이트에서 크랙을 제공해주기도 하였다.

참고 사이트

- http://www.securiteam.com/securitynews/6X00T00EBM.html

Cisco VPN Client Password 크랙 사이트

- http://www.unix-ag.uni-kl.de/~massar/bin/cisco-decode
- http://coreygilmore.com/projects/decrypt-cisco-vpn-password/

의도하지 않았지만.. 간단히 풀었던 문제이다.

[Crypto 300번]

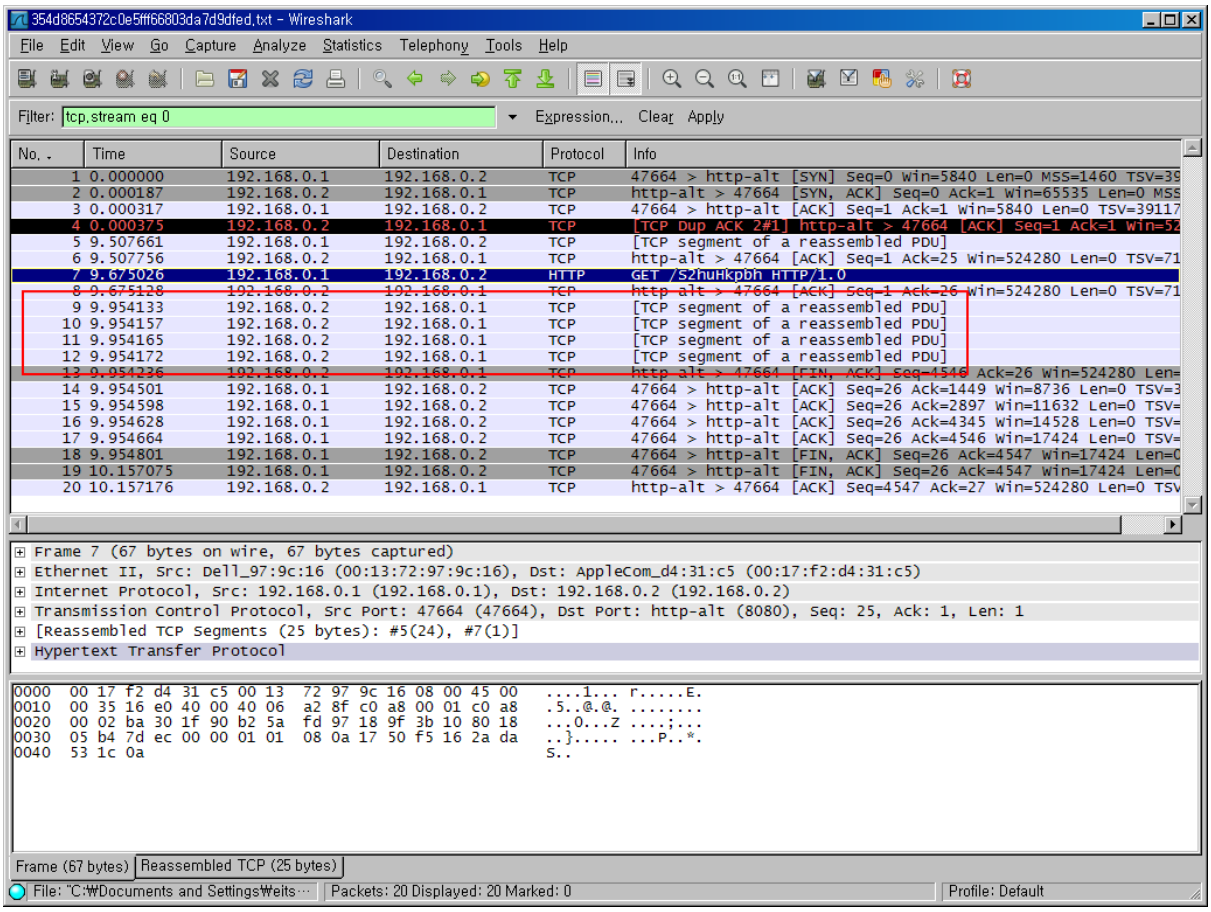
Question)

Work your way in.

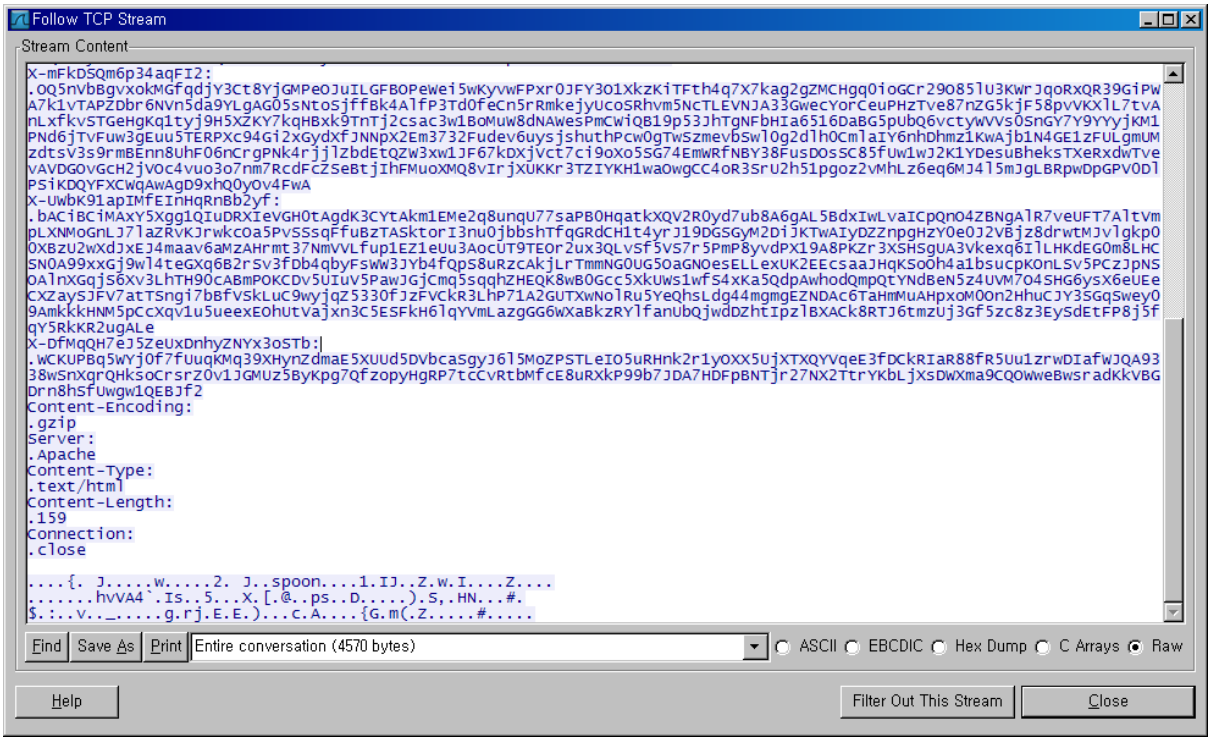
문제 파일 링크 - http://eits1st.tistory.com/attachment/cfile5.uf@143213164A5D7B4BCA90EE.txt

Solve)

문제 파일을 메모장으로 열어보면 "HTTP/1.1 200 OK", "Content-Encoding:", "SYN" 등의 문자열을볼 수 있으며, Packet Dump 파일이라는 것을 알 수 있다. 와이어샤크로 열어서 패킷을 분석 해보자.



Packet dump 내용은 그림에 보이는 것이 끝이며 GET /S2huHkpbh 에 대한 응답 값에 대한 패킷 외의 정보는 중요하지 않은 패킷이다. 패킷을 살펴보면 요청 값에 대한 응답 값으로 159byte의 어떤 파일을 받고 있다는 것을 알 수 있다.



개행문자(0d 0a)가 끝나는 지점, 즉 데이터가 시작되는 지점부터 159byte의 데이터를 Hex editor 프로그램으로 덤프 떠서 파일의 정보를 확인 해보자.

```
00001118  09 63 6c 6f 73 65 0d 0a  0d 0a 1f 8b 08 00 7b 07  .close.. ......{.
00001128  20 4a 02 03 01 88 00 77  ff 1f 8b 08 18 32 02 20   J.....w .....2.
00001138  4a 00 03 73 70 6f 6f 6e  00 d5 c4 84 31 1e 49 4a  J..spoon ....1.IJ
00001148  8f 94 5a 9b 77 c3 49 df  94 00 01 5a 00 a5 ff 8c  ..Z.w.I. ...Z....
00001158  0d 04 03 03 02 d1 da e6  68 76 56 41 34 60 c9 49  ........ hvVA4`.I
00001168  73 cb 86 35 89 d7 bf 58  ab 5b a2 40 fb b8 70 73  s..5...X .[.@..ps
00001178  91 82 44 bc f7 d1 9a a3  29 09 53 2c 0c 48 4e d6  ..D..... ).S,.HN.
00001188  da 04 23 d3 0a 24 b8 3a  eb b1 76 86 c1 5f b1 da  ..#..$.: ..v.._..
00001198  cf af a3 67 04 72 6a ea  45 10 45 f9 29 e3 a2 a7  ...g.rj. E.E.)...
000011A8  63 e9 41 b4 e2 b2 a7 7b  47 f5 6d 28 cf 5a 00 00  c.A....{ G.m(.Z..
000011B8  00 06 ed 23 a6 88 00 00  00                       ...#.... .
```

```
[root@localhost tmp]# file dumpfile
dumpfile: gzip compressed data, from Unix, last modified: Sat May 30 01:04:11
[root@localhost tmp]# xxd dumpfile
0000000: 1f8b 0800 7b07 204a 0203 0188 0077 ff1f  ....{. J.....w..
0000010: 8b08 1832 0220 4a00 0373 706f 6f6e 00d5  ...2. J..spoon..
0000020: c484 311e 494a 8f94 5a9b 77c3 49df 9400  ..1.IJ..Z.w.I...
0000030: 015a 00a5 ff8c 0d04 0303 02d1 dae6 6876  .Z............hv
0000040: 5641 3460 c949 73cb 8635 89d7 bf58 ab5b  VA4`.Is..5...X.[
0000050: a240 fbb8 7073 9182 44bc f7d1 9aa3 2909  .@..ps..D.....).
0000060: 532c 0c48 4ed6 da04 23d3 0a24 b83a ebb1  S,.HN...#..$.:..
0000070: 7686 c15f b1da cfaf a367 0472 6aea 4510  v.._.....g.rj.E.
0000080: 45f9 29e3 a2a7 63e9 41b4 e2b2 a77b 47f5  E.)...c.A....{G.
0000090: 6d28 cf5a 0000 0006 ed23 a688 0000 00    m(.Z.....#.....
[root@localhost tmp]#
```

gzip으로 압축된 파일 임을 알 수 있다. 압축을 풀어보자.

```
[root@localhost tmp]# mv dumpfile dumpfile.gz
[root@localhost tmp]# gzip -d dumpfile.gz
[root@localhost tmp]# file dumpfile
dumpfile: gzip compressed data, was "spoon", has comment, from Unix, comment,
[root@localhost tmp]# xxd dumpfile
0000000: 1f8b 0818 3202 204a 0003 7370 6f6f 6e00  ....2. J..spoon.
0000010: d5c4 8431 1e49 4a8f 945a 9b77 c349 df94  ...1.IJ..Z.w.I..
0000020: 0001 5a00 a5ff 8c0d 0403 0302 d1da e668  ..Z............h
0000030: 7656 4134 60c9 4973 cb86 3589 d7bf 58ab  vVA4`.Is..5...X.
0000040: 5ba2 40fb b870 7391 8244 bcf7 d19a a329  [.@..ps..D.....)
0000050: 0953 2c0c 484e d6da 0423 d30a 24b8 3aeb  .S,.HN...#..$.:.
0000060: b176 86c1 5fb1 dacf afa3 6704 726a ea45  .v.._.....g.rj.E
0000070: 1045 f929 e3a2 a763 e941 b4e2 b2a7 7b47  .E.)...c.A....{G
0000080: f56d 28cf 5a00 0000                      .m(.Z...
[root@localhost tmp]# mv dumpfile dumpfile.gz
[root@localhost tmp]# gzip -d dumpfile.gz
[root@localhost tmp]# file dumpfile
dumpfile: data
[root@localhost tmp]# xxd dumpfile
0000000: 8c0d 0403 0302 d1da e668 7656 4134 60c9  .........hvVA4`.
0000010: 4973 cb86 3589 d7bf 58ab 5ba2 40fb b870  Is..5...X.[.@..p
0000020: 7391 8244 bcf7 d19a a329 0953 2c0c 484e  s..D.....).S,.HN
0000030: d6da 0423 d30a 24b8 3aeb b176 86c1 5fb1  ...#..$.:..v.._.
0000040: dacf afa3 6704 726a ea45 1045 f929 e3a2  ....g.rj.E.E.)..
0000050: a763 e941 b4e2 b2a7 7b47                 .c.A....{G
[root@localhost tmp]#
```

한번 압축을 풀었으나 또 압축된 파일임으로 다시 풀었다. 마침내 data 형식의 file을 구할 수 있었다.

파일 첫 부분의 Hex값은 대부분 어떤 파일의 형식인지 나타내므로 8c 0d 04 03 으로 검색을 하였고, 그 결과 CAST5 방식으로 암호화된 data 파일이란 것을 알 수 있었다. 암호화된 data를 복호화 한다면 원하는 값을 얻을 수 있을 것이다. 하지만 복호화를 시도하게 되면 passphrase 값, 즉 암호화 시 사용되었던 키 값을 알아야 한다.

관련 정보 URL : http://lists.gnupg.org/pipermail/gnupg-users/2006-November/029658.html

```
[root@localhost tmp]# gpg -d dumpfile
gpg: CAST5 encrypted data
Enter passphrase:
```

```
[root@localhost tmp]# gpg -d dumpfile
gpg: CAST5 encrypted data
gpg: encrypted with 1 passphrase
gpg: decryption failed: bad key
[root@localhost tmp]#
```

사실 이 부분에서 passphrase 값을 알아내기 위해 수많은 삽질;;을 하였으나, 결국 알아내지 못하였다. 좌절 후 지인에게 **gzip file format을 잘 살펴보라는 힌트**를 들은 후 문제를 풀 수 있었다.

문제를 통해 gzip 파일을 2개 얻을 수가 있으며, 둘 다 구조를 분석을 했었다. 여기서는 그 중 정답이 있는 두 번째 gzip 파일 구조에 대해서 확인해 보자.

```
0000000: 1f8b 0818 3202 204a 0003 7370 6f6f 6e00  ....2. J..spoon.
0000010: d5c4 8431 1e49 4a8f 945a 9b77 c349 df94  ...1.IJ..Z.w.I..
0000020: 0001 5a00 a5ff 8c0d 0403 0302 d1da e668  ..Z............h
0000030: 7656 4134 60c9 4973 cb86 3589 d7bf 58ab  vVA4`.Is..5...X.
0000040: 5ba2 40fb b870 7391 8244 bcf7 d19a a329  [.@..ps..D.....)
0000050: 0953 2c0c 484e d6da 0423 d30a 24b8 3aeb  .S,.HN...#..$.:.
0000060: b176 86c1 5fb1 dacf afa3 6704 726a ea45  .v.._.....g.rj.E
0000070: 1045 f929 e3a2 a763 e941 b4e2 b2a7 7b47  .E.)...c.A....{G
0000080: f56d 28cf 5a00 0000                      .m(.Z...
```

처음 10byte는 gzip 파일의 member Header 이다. 우선 맨 앞의 2byte는 파일의 포맷을 알려주는 부분이며 나와있는 0x1f 0x8b 이 이를 나타낸다. 그 다음 1byte는 Compression Method 로 gzip 파일의 경우 8로 셋팅이 된다고 한다. 역시나 0x08로 되어있다. 그 다음 1byte는 FLAG 값을 나타내며 해당 bit의 값에 따라 기본헤더 이후에 오는 Hex 값의 내용이 달라진다. 해당 1byte를 아래 표를 보며 확인을 해보자. 현재 0x18 로 되어있으니 BIT 3과 BIT 4가 Hex 내용에 포함이 될 것이다. 앞으로는 이를 고려하여 파일 형식의 내용을 확인해야 한다.

Bit 0	FTEXT
Bit 1	FHCRC
Bit 2	FEXTRA
Bit 3	FNAME
Bit 4	FCOMMENT
Bit 5	Reserved
Bit 6	Reserved
Bit 7	Reserved

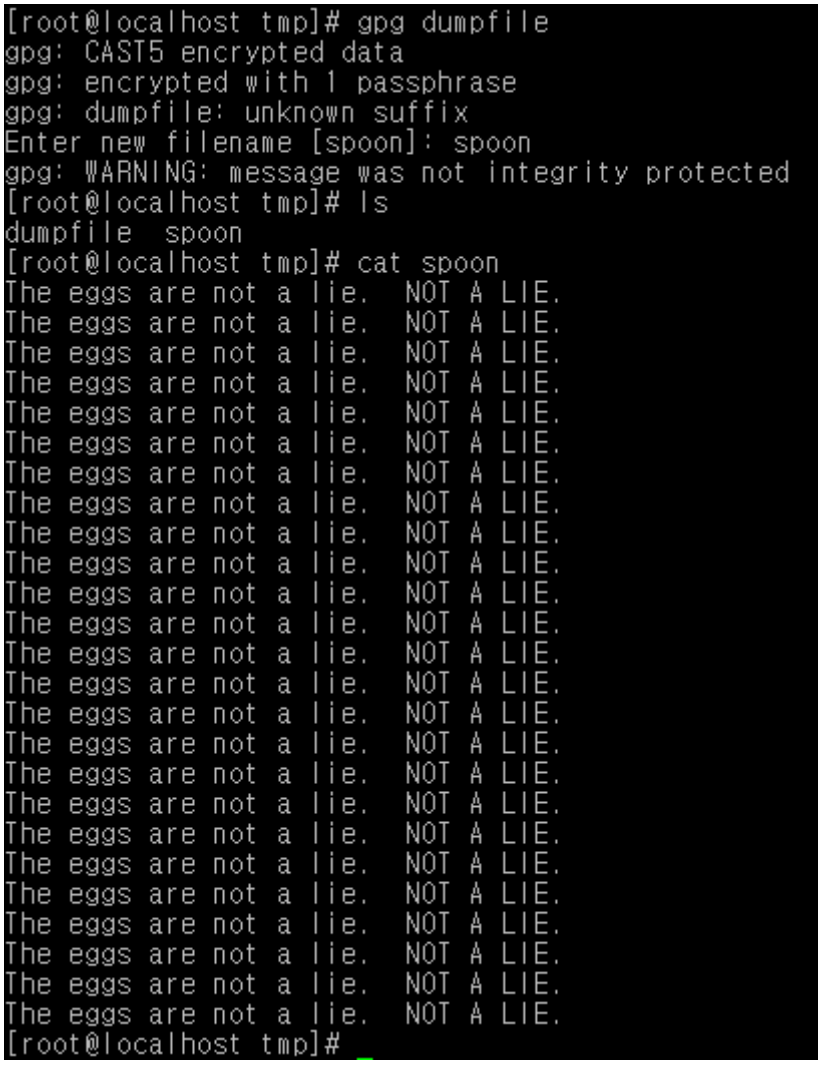
헤더 내용 중 4byte를 확인 하였다. 그 다음 4byte의 값인 0x32 0x02 0x20 0x4a 는 압축파일이 생성된 시간을 나타내며, UNIXSTAMP 값으로 되어있다. 그 뒤에 오는 각 1byte씩은 XFL 와 OS에 관련된 값이며 압축률와 압축이 일어난 파일 시스템을 나타낸다. Gzip

파일의 헤더 부분을 모두 보았으며 그 다음은 내용부터 헤더에 셋팅 된 FLAG값을 토대로 확인 하면 될 것이다.

FLAG 값에 의해 기본헤더 다음에 오는 내용은 FNAME과 FCOMMENT 이다. FNAME이 셋팅 되면 NULL로 끝나는 원본 파일의 이름이 추가되며, FCOMMEMT는 NULL로 끝나는 간단한 설명이 들어간다고 한다. 현재 HEX 코드에 는 0x00으로 끝나는 값이 두 개 있다. 헤더 값 바로 뒤의 6f 6f 6e값과 d5c4 8431 1e49 4a8f 945a 9b77 c349 df94 값이다. 6f 6f 6e 는 원본 파일이름이 "spoon" 이라는 것을 나타내는 것이며 d5c4 8431 1e49 4a8f 945a 9b77 c349 df94 는 코멘트를 의미하는 것일 것이다.

그 뒤에 오는 부분은 압축된 본문내용이 올 것이며 마지막 8byte인 f56d 28cf 5a00 0000 는 4byte 씩 각각 CRC32와 압축되기 전의 파일 사이즈를 의미한다. 최종적으로 얻은 dumpfile의 사이즈가 90byte이므로 5a의 값과 일치한다. Gzip 파일 형식에 대한 HEX값의 분석이 끝났다. 의심되는 부분은 원본 파일이름을 나타내는 6f 6f 6e 부분과 코멘트 부분인 d5c4 8431 1e49 4a8f 945a 9b77 c349 df94 이다.

사실 이렇게 하고도 복호화 키 값을 알아차리지 못하였다. 키 값을 먼저 말하자면 코멘트 부분의 hex 값 그대로 "d5c484311e494a8f945a9b77c349df94" 이다. 이것이 passphrase 의 값이 되며, 생성되는 파일의 이름으로 "spoon"을 넣어주면 복호화 된 spoon 파일을 얻을 수 있다.



```
[root@localhost tmp]# gpg dumpfile
gpg: CAST5 encrypted data
gpg: encrypted with 1 passphrase
gpg: dumpfile: unknown suffix
Enter new filename [spoon]: spoon
gpg: WARNING: message was not integrity protected
[root@localhost tmp]# ls
dumpfile  spoon
[root@localhost tmp]# cat spoon
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
The eggs are not a lie.  NOT A LIE.
[root@localhost tmp]# _
```

복호화 "The eggs are not a lie. NOT A LIE." 문자열이 최종 정답이다.

반복되는 삽질 속에 정답을 나중에야 알 수 있었다. ^^; 300번 문제를 풀며 대충 알고 있었던 PGP/GPG에 대한 부분을 조금 더 자세히 알 수 있었던 것 같다.

Gzip file format 구조 참고 URL

- http://kaistizen.net/project/Zip/GzipFileFormat.html