ISSN 2287-3252

ICCT 2015

"The 5th International Conference on Convergence Technology 2015"

Vol.5 No.1

- **Date : June 29 – July 2, 2015**
- **Place : Chateraise Gateaux Kingdom Sapporo Hotel, Hokkaido, Japan**
- **Co-organized by :**
  - Korea Convergence Society
  - Korea Institute of Science and Technology Information
  - The Korean Association for Comparative Government
  - The Society of Digital Policy & Management
  - Convergence Society for SMB
  - Konyang Univ. Well-Dying LAB
  - Korea Mobile Enterprise Promotion Association
  - DAEHAN Society of Industrial Management
- **Sponsored by :**

KISTI NTIS Division & GSDC Center, LG Hitachi Co., Ltd, ALLforLAND Co., Ltd, KYUNGBONG Co., Ltd, SOFTITECH Co., Ltd, TAEJIN Infortech Co., Ltd, Geo Matics Co., Ltd, MIJU C&D Co., Ltd, R2soft Co., Ltd, Hanbit Academy, Inc., Korea IT Consulting Co., Ltd, Open Link System Co., Ltd, SungWon-IT Co., Ltd, SJ info&communications Co., Ltd, Neighbor system Co., Ltd, Able IT Co., Ltd, INFORMADE Co., Ltd, SelimTSG Co. Ltd, KORNEC Co., Ltd, MTData Co., Ltd, Mobile Law Co., Ltd, DELTASYSTEM Co., Ltd, Daewoo Information Systems Co., Ltd, LG CNS Co., Ltd, ICTWAY Co., Ltd, GFT Co., Ltd, QbizOn Co., Ltd, NANUS Information Co., Ltd, Hankyung I-NET Co., Ltd, VITZROSYS Co., Ltd, Duplex Co., Ltd, Maverick Systems Co., Ltd, Bizmerce Co., Ltd, DAWON ICT Co., Ltd, INNOZIUM Co., Ltd, MetaBiz Co., Ltd, Human Information Co., Ltd, AtechIns Co., Ltd, Comtec System Co., Ltd

KCS KOREA CONVERGENCE SOCIETY
http://www.kcons.or.kr

Session 5–C (SW Visualization (SE Center)) Chair R. Young Chul Kim(Hongik Univ.)

● 13:00~14:20 Tuesday June 30, 2015

01. W-13-01_Hybrid Visual Security Analysis to Prove Vulnerability for Safety / 262
Seok Mo Kim(Dankook Univ., Korea), Sang Eun Lee(Software Engineering Center, Korea), Su Nam Jeon(Software Engineering Center, Korea), Young B. Park(Dankook Univ., Korea)

02. W-13-02_Compatibility Enhancing Agent for Managing Software Toolchain / 264
Eun Seung Lee(Dankook Univ., Korea), Young Soo Kim(Software Engineering Center, Korea), Byungho Park(Hongik Univ., Korea), Young B. Park(Dankook Univ., Korea)

03. W-13-03_A Visualized Blocking Methtod against a Hidden Malware in the Image / 266
Byungho Park(Ministry of National Defence, Korea), R.Young Chul Kim(Hongik Univ., Korea), Young B. Park(Dankook Univ., Korea), Daecheol Shin(Korea e-Government Exp. Associations, Korea), Young Soo Kim(NIPA, Korea), SangEun Lee(NIPA, Korea)

04. W-13-04_A Guideline for Realization on extracting automatic size maturity level of diverse component via Source Codes / 268
JunSun Hwang(Hongik Univ., Korea), R. Youngchul Kim(Hongik Univ., Korea), SangEun Lee(NIPA, Korea)

05. W-13-05_Design of a Flamework of 3D Geofence and Geocode / 270
Jun Cho(Gangneung–Wonju National Univ., Korea), Kihyun Kim(Gangneung–Wonju National Univ., Korea), Jinhyung Park(Gangneung–Wonju National Univ., Korea), Sungjin Cho(Gangneung–Wonju National Univ., Korea), Byungkook Jeon(Gangneung–Wonju National Univ., Korea), Sungkuk Cho(Gangneung–Wonju National Univ., Korea)

06. W-13-06_Design of a Temporal Geofence System / 272
Kihyun Kim(Gangneung–Wonju National Univ., Korea), Jun Cho(Gangneung–Wonju National Univ., Korea), Jinhyung Park(Gangneung–Wonju National Univ., Korea), Sungjin Cho(Gangneung–Wonju National Univ., Korea), Byungkook Jeon(Gangneung–Wonju National Univ., Korea), Sungkuk Cho(Gangneung–Wonju National Univ., Korea)

07. W-13-07_Extracting Designs via Code on Reverse Engineering / 274
Haeun Kwon(Hongik Univ., Korea), Bokyung Park(Hongik Univ., Korea), R. Youngchul Kim(Hongik Univ., Korea), SangEun Lee(NIPA, Korea)

08. W-13-08_Extracting performance factors against performance degradation through Code Visualization / 276
Geon–Hee Kang(Hongik Univ., Korea), R.Young Chul Kim(Hongik Univ., Korea), SangEun Lee(NIPA, Korea), Su Nam Jeon(NIPA, Korea)

A Visualized Blocking Methtod againist a Hidden Malware in the Image

[1] Byungho Park, [2] R.Young Chul Kim, [3] Young B. Park [4] Daecheol Shin,
[5] Young Soo Kim, [6] SangEun Lee
[1, First Author,] *Ministry of National Defence, Seoul, Korea, sunsonbob@naver.com*
[2,] *Dept. of CIC, Hongik University, Sejong, Korea, bob@selab.hongik.ac.kr*
[3,] *Dept. of Computer Science, Dankook University, Korea, ybpark@dankook.ac.kr*
[4,] *Korea e-Government Exp. Associations, Seoul, Korea, dcshin04@nate.com*
[5, *6 Corresponding Author] *NIPA, Seoul, Korea, {ysgold,selee}@nipa.kr*

Abstract In the Internet, we conveniently download the image file which doesn't know whether it contains a malious code or not. Moreover, the Internet Image files are passages which are easy enough to flow in a closed network, and a malicious code, which is secretly inserted into a picture file, can be performed as a malicious offensive code, that is, a hacking code with triggers such as HTML files or JS files.

So, we suggest an idea for safe use with a visualized blocking method of the image file which dosen't know whether to contain a malicious performance code in a closed network or not.

Keywords: *Code Visualization, Malicious Code, Image hacking*

1. Introduction

Various information can be taken advantage of on the Internet which is conveniently used, and particularly, pictures are reused by downloading picture files on the Internet easily. They are known to have Stegano graphy to conceal data in a picture, but generally, it does not have an offensive function[3].

This paper identifies how the images, which are created by putting a malicious performance code in the image file, are serious, and for its preparation, it examines image management schemes as far as a general management one which does not need a special control and the inside of a file in accordance with each closed network. Then it investigates the existence of a malicious hacking code, security management which can be used if it is clear, and finally, the possibility of hacking the inside of a picture. It suggests a high security management method to create and store a separate image capture file in order to fundamentally block internal hacking data which may exist by using an image capture tool if clear.

2. Main Subject

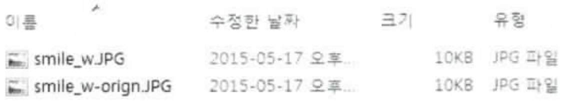
Generally, a malware refers to a software or a malicious code which is created with an incorrect purpose or a wrong intention to conduct harmful behaviors, and it also includes a script virus[2].

The following pictures show two images, a clean one and the other including a malicious code respectively.

Fig1. Original picture (smile_w-orign.jpg)

Fig2. The picture containing a hacking code (smile_w.jpg)

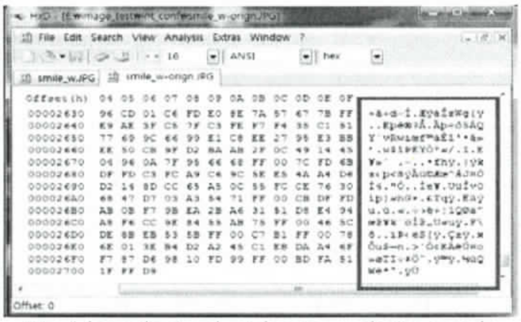
As seen, those two images are the same in size and quality of the picture.



이름	수정한 날짜	크기	유형
smile_w.JPG	2015-05-17 오후...	10KB	JPG 파일
smile_w-orign.JPG	2015-05-17 오후...	10KB	JPG 파일

Fig3. The images are the same between the original and a hacking code

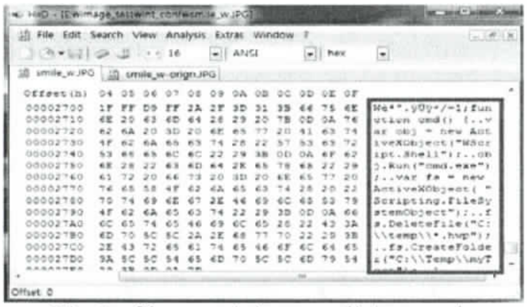
However, they are different in their internal image structures. We show the hexadata code using a hexa editor tool.



```
Offset(h)  04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00002630   96 CD 01 C4 FD E0 8E 7A 57 47 7B FF
00002640   E9 AE 3F C5 7F C3 FE F7 F4 35 C1 51
00002650   77 69 9C 66 99 E1 C8 EE 27 95 E3 BB
00002660   EE 50 CB 9F D2 8A AB 2F 0C 49 14 45
00002670   04 96 0A 7F 95 66 68 FF 00 7C FD 6B
00002680   DF FD C3 FC A9 C6 9C 5E E5 4A A4 D6
00002690   D2 14 8D CC 65 A5 0C 55 FC CE 76 30
000026A0   68 47 D7 03 A3 54 71 FF 00 CB DF FD
000026B0   AB 0B F7 9B EA 2B A6 31 51 D8 E4 94
000026C0   A8 F6 CC 9E 84 55 AB 75 FF 00 46 5C
000026D0   DE 8B EB 53 5B FF 00 C7 B1 FF 00 78
000026E0   6E 01 3E B4 D2 A2 45 C1 E8 DA A4 6F
000026F0   F7 87 D6 98 10 FD 99 FF 00 BD FA 51
00002700   1F FF D9
```

HxD - [E:\image_test\int_conf\smile_w-orign.JPG]
File Edit Search View Analysis Extras Window ?
16
ANSI
hex
smile_w.JPG
smile_w-orign.JPG
Offset: 0

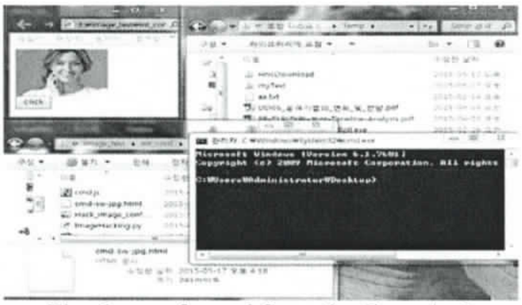
Fig.4 hexadecimal without a malicious code (smile_w-orign.jpg)



```
Offset(h)  04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00002700   1F FF D9 FF 2A 2F 3D 31 3B 66 75 6E   He*".ÿÙÿ*/=1;fun
00002710   6E 20 63 6D 64 28 29 20 7B 0D 0A 76   ction cmd() {..v
00002720   62 6A 20 3D 20 6E 65 77 20 41 63 74   ar obj = new Act
00002730   4F 62 6A 65 63 74 28 22 57 53 63 72   iveXObject("WScr
00002740   53 68 65 6C 6C 22 29 3B 0D 0A 6F 62   ipt.Shell");..ob
00002750   6E 28 22 63 6D 64 2E 65 78 65 22 29   j.Run("cmd.exe")
00002760   61 72 20 66 73 20 3D 20 6E 65 77 20   ;..var fs = new 
00002770   76 65 58 4F 62 6A 65 63 74 28 20 22   ActiveXObject( "
00002780   70 74 69 6E 67 2E 46 69 6C 65 53 79   Scripting.FileSy
00002790   4F 62 6A 65 63 74 22 29 3B 0D 0A 66   stemObject");..f
000027A0   6C 65 74 65 46 69 6C 65 28 22 43 3A   s.DeleteFile("C:
000027B0   6D 70 5C 5C 2A 2E 68 77 70 22 29 3B   \\temp\\*.hwp");
000027C0   2E 43 72 65 61 74 65 46 6F 6C 64 65   ..fs.CreateFolde
000027D0   3A 5C 5C 54 65 6D 70 5C 5C 6D 79 54   r("C:\\Temp\\myT
```

HxD - [E:\image_test\int_conf\smile_w.JPG]
File Edit Search View Analysis Extras Window ?
16
ANSI
hex
smile_w.JPG
smile_w-orign.JPG
Offset: 0

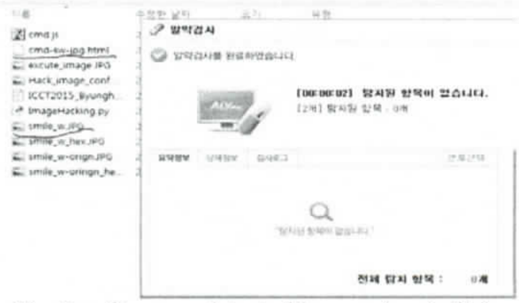
Fig.5 a file containing a malicious code (smile_w.jpg)



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights

C:₩Users₩Administrator₩Desktop>
```

Fig. 6. a performed figure(smile_w.jpg)



정밀검사

정밀검사를 완료하였습니다

[00:00:02] 탐지된 항목이 없습니다.
[2개] 탐지된 항목 : 0개

탐지된 항목이 없습니다

전체 탐지 항목 : 0개

Fig. 7. a figure undetected by vaccines(pills)

So, we suggest an idea for safe use with a visualized blocking method of the image file which dosen't know whether to contain a malicious performance code in a closed network or not.

On an important network, an image management plan is performed in the following procedure:

1) A general internet network does not have a special measure.

2) An important security network visualizes a file internally and then uses an image.

3) In a security network of high level, the image file only uses the captured image using an image capture tool.

3. Conclusion

Internet Image files are passages which are easy enough to flow in a closed network, and a malicious code, which is secretly inserted into a picture file, can be performed as a malicious offensive code, that is, a hacking code with triggers such as HTML files or JS files [5].

This paper presents three methods for managing an image file as follows.

A typical internet network can be used without special measures, and an important security network can visualize a file internally to use an image. Finally, in a security network of high level, the image file only uses the captured image using an image capture tool in order to fundamentally block the flow of a malicious code in a closed network.

References

[1] http://ja.wikipedia.org/wiki

[2] Sungmun Cho, Yonghun Jeong, Introduction to Python hacking, Freelex, 2014.12

[3] http://www.openstego.info/

[4] http://www.sw-eng.kr/member/index.do

[5] http://www.sw-eng.kr/