Analyzing a doc file

1. No. of objects in the file.

```
remnux@remnux: ~/Downloads/pdf-doc
File  Edit  Tabs  Help
remnux@remnux:~$ cd Downloads/pdf-doc
remnux@remnux:~/Downloads/pdf-doc$ pdfid doc.pdf
PDFiD 0.2.1 doc.pdf
 PDF Header: %PDF-1.1
 obj                    9
 endobj                 9
 stream                 2
 endstream              2
 xref                   1
 trailer                1
 startxref              1
 /Page                  1
 /Encrypt               0
 /ObjStm                0
 /JS                    1
 /JavaScript            1
 /AA                    0
 /OpenAction            1
 /AcroForm              0
 /JBIG2Decode           0
 /RichMedia             0
```

Pdfid shows that there are 9 objects and java script code in the file.

2. Determine whether the file is compressed or not.

remnux@remnux: ~/Downloads/pdf-doc

File Edit Tabs Help

```
remnux@remnux:~/Downloads/pdf-doc$ pdf-parser.py --content doc
.pdf
PDF Comment '%PDF-1.1\r\n'

PDF Comment '%\xd0\xd0\xd0\xd0\r\n'

obj 1 0
 Type: /Catalog
 Referencing: 2 0 R, 3 0 R, 7 0 R, 9 0 R

  <<
    /Type /Catalog
    /Outlines 2 0 R
    /Pages 3 0 R
    /Names
      <<
        /EmbeddedFiles
          <<
            /Names [(eicar-dropper.doc) 7 0 R]
          >>
      >>
```

remnux@remnux: ~/Downloads/pdf-doc

File Edit Tabs Help

```
<<
 /Type /Catalog
 /Outlines 2 0 R
 /Pages 3 0 R
 /Names << /EmbeddedFiles << /Names [(eicar-dropper.doc) 7 0 R
] >> >>
 /OpenAction 9 0 R
>>


obj 2 0
 Type: /Outlines
 Referencing:

  <<
    /Type /Outlines
    /Count 0
  >>


<<
```

```
obj 3 0
 Type: /Pages
 Referencing: 4 0 R

  <<
    /Type /Pages
    /Kids [4 0 R]
    /Count 1
  >>


<<
 /Type /Pages
 /Kids [4 0 R]
 /Count 1
>>


obj 4 0
 Type: /Page
 Referencing: 3 0 R, 5 0 R, 6 0 R
```

```
obj 8 0
 Type: /EmbeddedFile
 Referencing:
 Contains stream

  <<
    /Length 8952
    /Filter /FlateDecode
    /Type /EmbeddedFile
  >>
```

Above picture shows that file is compressed.

```
obj 9 0
 Type: /Action
 Referencing:

  <<
    /Type /Action
    /S /JavaScript
    /JS (this.exportDataObject({ cName: "eicar-dropper.doc", n
Launch: 2 });)
  >>


<<
 /Type /Action
 /S /JavaScript
 /JS (this.exportDataObject({ cName: "eicar-dropper.doc", nLau
nch: 2 });)
>>
```

/Filter represents that the file is encoded with Flat Encode Compression algorithm. We can see that there is java script code embedded in it.

3. Determine whether the file is obfuscated or not.

```
remnux@remnux:~/Downloads/pdf-doc$ pdfinfo doc.pdf
Tagged:         no
Form:           none
Pages:          1
Encrypted:      no
Page size:      612 x 792 pts (letter)
Page rot:       0
File size:      10381 bytes
Optimized:      no
PDF version:    1.1
remnux@remnux:~/Downloads/pdf-doc$
```

4. The contents in the pdf file are extracted using pdfextract.

```
File  Edit  Tabs  Help
remnux@remnux:~/Downloads/pdf-doc$ pdfextract doc.pdf
Extracted 2 PDF streams to 'doc.dump/streams'.
Extracted 1 scripts to 'doc.dump/scripts'.
Extracted 1 attachments to 'doc.dump/attachments'.
Extracted 0 fonts to 'doc.dump/fonts'.
Extracted 0 images to 'doc.dump/images'.
remnux@remnux:~/Downloads/pdf-doc$ cd doc.dump/attachments
remnux@remnux:~/Downloads/pdf-doc/doc.dump/attachments$ ls -al
total 44
drwxrwxr-x 2 remnux remnux  4096 Feb 17 16:57 .
drwxrwxr-x 7 remnux remnux  4096 Feb 15 13:00 ..
-rw-rw-r-- 1 remnux remnux 31744 Feb 17 20:35 attached_eicar-d
-rw-rw-r-- 1 remnux remnux   924 Feb 17 16:57 m1
remnux@remnux:~/Downloads/pdf-doc/doc.dump/attachments$ oledum
p.py attached_eicar-dropper.doc
  1:       114 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:      6509 '1Table'
  5:       409 'Macros/PROJECT'
  6:        65 'Macros/PROJECTwm'
  7: M    3716 'Macros/VBA/Module1'
  8: m     924 'Macros/VBA/ThisDocument'
  9:      2601 'Macros/VBA/_VBA_PROJECT'
 10:       563 'Macros/VBA/dir'
 11:      4096 'WordDocument'
remnux@remnux:~/Downloads/pdf-doc/doc.dump/attachments$
```

doc.dump folder is created with streams, attachments, scripts, fonts and images. Now, open the attachments folder, we can see that there is one doc file inside it which is named as attached_eicar_dropper.doc

oledump.py is a program to analyze OLE files (Compound File Binary Format). These files contain streams of data. oledump allows you to analyze these streams. Using this oledump.py to see what is in the attached doc file, we can see that there are macro files in it.

Opening these macro files, the output looks in the following way,

For macro 8

```
remnux@remnux:~/Downloads/pdf-doc/doc.dump/attachments$ oledum
p.py attached_eicar-dropper.doc -s 8 -d
ME
1Normal.Th
isDocument
Attribute VB_Name = "Thi
sDocumen"

Bas 1Normal Global! Spac Flse
Crea abl Pre decla d
BExose TemplateDeri $Custom
iz C 1remnux@remnux:~/Downloads/
pdf-doc/doc.dump/attachments$
```

For macro 7

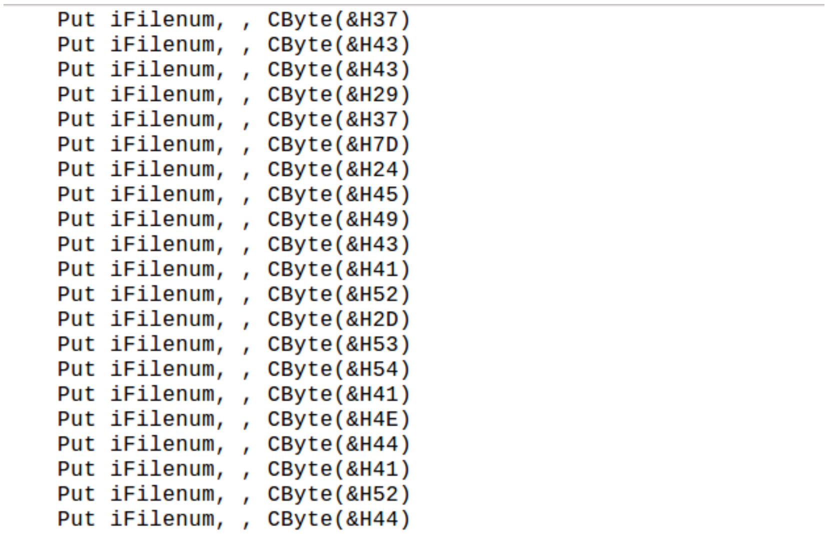
```
iz C 1remnux@remnux:~/Downloads/
p.py attached_eicar-dropper.doc -s 7 -d
ME
S
p
S H
$" (  $X
Scripting.Fi
leSystemObject$0 .  Se emp$2 4 * Sys *
s XXD ir 5XD \" OXD pN !XD n PXD Bi
%XD it XD AXD PXD [XD 4XD
P \XD C PXD Pu ZXD CB XXD ut 5XD By
4XD t (XD yt PXD i ^XD te )XD iF 7XD e(
CXD Fi CXD (& )XD il 7XD &H }XD le $XD
H3 EXD en IXD 5C CXD nu AXD 0) RXD um
-XD) SXD m, TXD AXD, NXD DXD
AXD RXD, DXD P -XD C AXD Pu NX
D CB TXD ut IXD By VXD t IXD yt RXD i
UXD te SXD iF -XD TXD EXD, SXD
TXD -XD FXD, IXD P LXD C EXD
Pu !XD CB $XD ut HXD By +XD t HXD yt
*XD i #ICAR test file written: A@6 52) o
Attribute VB_Name = "Module1"
Sub AutoOpen()
Dim sFlen As S)ng8i um 6IntegeBr8oFSO O bject = Fr
ee Set = C reate ("Scriptin . Syste m = Environ(
"%p") & "
\" .Get Temp {vFor Binary Acce ss Wri tC ^
```

If observed, the macro file contains XD repeatedly (that looks something like XOR obfuscation).

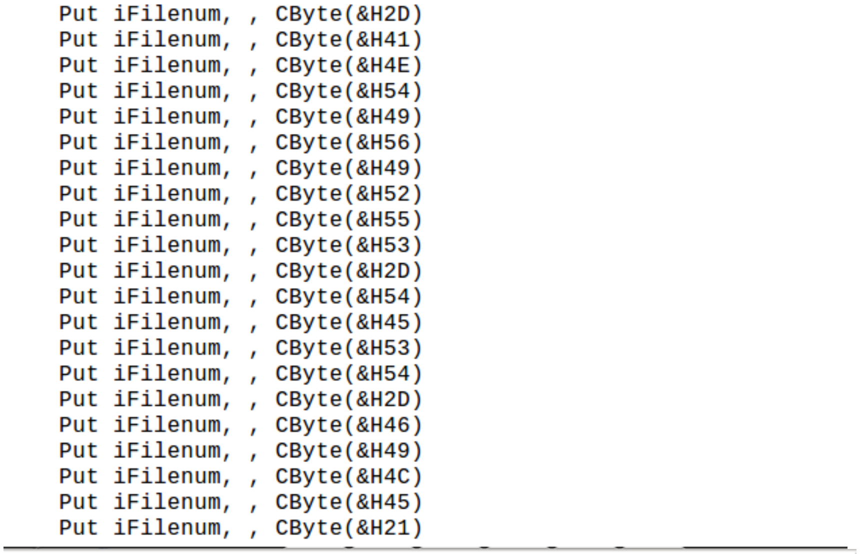
Olevba is used to extract and analyze VBA Macro source code from MS Office documents.

```
remnux@remnux:~/Downloads/pdf-doc/doc.dump/attachments$ olevba
.py attached_eicar-dropper.doc
olevba 0.27 - http://decalage.info/python/oletools
Flags       Filename

----------- -----------------------------------------------
---------------
OLE:MAS---- attached_eicar-dropper.doc

(Flags: OpX=OpenXML, XML=Word2003XML, MHT=MHTML, M=Macros, A=A
uto-executable, S=Suspicious keywords, I=IOCs, H=Hex strings,
B=Base64 strings, D=Dridex strings, ?=Unknown)

===============================================================
=================
FILE: attached_eicar-dropper.doc
Type: OLE
---------------------------------------------------------------
-----------------
VBA MACRO ThisDocument.cls
in file: attached_eicar-dropper.doc - OLE stream: u'Macros/VBA
```

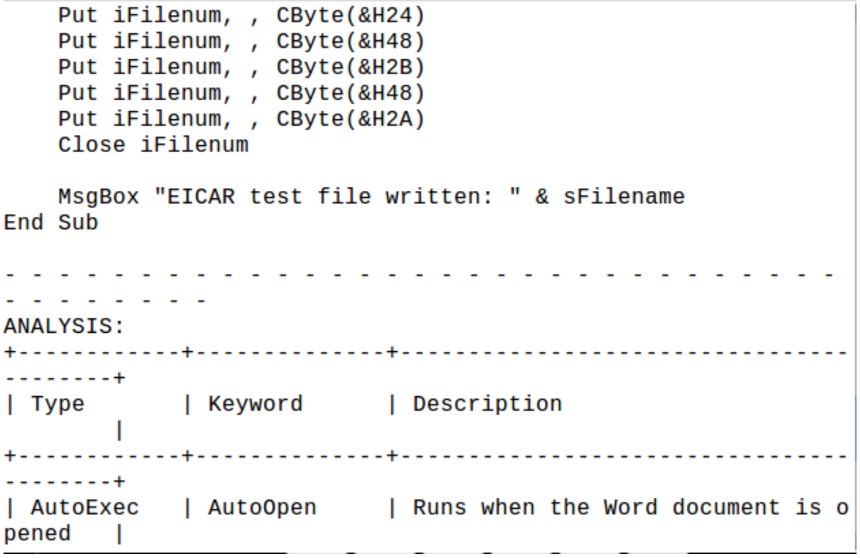
```
/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
(empty macro)
---------------------------------------------------------------
-----------------
VBA MACRO Module1.bas
in file: attached_eicar-dropper.doc - OLE stream: u'Macros/VBA
/Module1'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - -
Sub AutoOpen()
    Dim sFilename As String
    Dim iFilenum As Integer
    Dim oFSO As Object

    iFilenum = FreeFile
    Set oFSO = CreateObject("Scripting.FileSystemObject")
    sFilename = Environ("temp") & "\" & oFSO.GetTempName

    Open sFilename For Binary Access Write As iFilenum
    Put iFilenum, , CByte(&H58)
    Put iFilenum, , CByte(&H35)
    Put iFilenum, , CByte(&H4F)
    Put iFilenum, , CByte(&H21)
    Put iFilenum, , CByte(&H50)
    Put iFilenum, , CByte(&H25)
    Put iFilenum, , CByte(&H40)
    Put iFilenum, , CByte(&H41)
    Put iFilenum, , CByte(&H50)
    Put iFilenum, , CByte(&H5B)
    Put iFilenum, , CByte(&H34)
    Put iFilenum, , CByte(&H5C)
    Put iFilenum, , CByte(&H50)
    Put iFilenum, , CByte(&H5A)
    Put iFilenum, , CByte(&H58)
    Put iFilenum, , CByte(&H35)
    Put iFilenum, , CByte(&H34)
    Put iFilenum, , CByte(&H28)
    Put iFilenum, , CByte(&H50)
    Put iFilenum, , CByte(&H5E)
    Put iFilenum, , CByte(&H29)
```



```
    Put iFilenum, , CByte(&H37)
    Put iFilenum, , CByte(&H43)
    Put iFilenum, , CByte(&H43)
    Put iFilenum, , CByte(&H29)
    Put iFilenum, , CByte(&H37)
    Put iFilenum, , CByte(&H7D)
    Put iFilenum, , CByte(&H24)
    Put iFilenum, , CByte(&H45)
    Put iFilenum, , CByte(&H49)
    Put iFilenum, , CByte(&H43)
    Put iFilenum, , CByte(&H41)
    Put iFilenum, , CByte(&H52)
    Put iFilenum, , CByte(&H2D)
    Put iFilenum, , CByte(&H53)
    Put iFilenum, , CByte(&H54)
    Put iFilenum, , CByte(&H41)
    Put iFilenum, , CByte(&H4E)
    Put iFilenum, , CByte(&H44)
    Put iFilenum, , CByte(&H41)
    Put iFilenum, , CByte(&H52)
    Put iFilenum, , CByte(&H44)
```



```
    Put iFilenum, , CByte(&H2D)
    Put iFilenum, , CByte(&H41)
    Put iFilenum, , CByte(&H4E)
    Put iFilenum, , CByte(&H54)
    Put iFilenum, , CByte(&H49)
    Put iFilenum, , CByte(&H56)
    Put iFilenum, , CByte(&H49)
    Put iFilenum, , CByte(&H52)
    Put iFilenum, , CByte(&H55)
    Put iFilenum, , CByte(&H53)
    Put iFilenum, , CByte(&H2D)
    Put iFilenum, , CByte(&H54)
    Put iFilenum, , CByte(&H45)
    Put iFilenum, , CByte(&H53)
    Put iFilenum, , CByte(&H54)
    Put iFilenum, , CByte(&H2D)
    Put iFilenum, , CByte(&H46)
    Put iFilenum, , CByte(&H49)
    Put iFilenum, , CByte(&H4C)
    Put iFilenum, , CByte(&H45)
    Put iFilenum, , CByte(&H21)
```



```
    Put iFilenum, , CByte(&H24)
    Put iFilenum, , CByte(&H48)
    Put iFilenum, , CByte(&H2B)
    Put iFilenum, , CByte(&H48)
    Put iFilenum, , CByte(&H2A)
    Close iFilenum

    MsgBox "EICAR test file written: " & sFilename
End Sub

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - -
ANALYSIS:
+------------+-------------+-------------------------------
--------+
| Type       | Keyword     | Description
        |
+------------+-------------+-------------------------------
--------+
| AutoExec   | AutoOpen    | Runs when the Word document is o
pened    |
```

```
| Suspicious | Open         | May open a file
        |
| Suspicious | CreateObject | May create an OLE object
        |
| Suspicious | Environ      | May read system environment vari
ables   |
| Suspicious | Binary       | May read or write a binary file
(if     |
|            |              | combined with Open)
        |
| Suspicious | Write        | May write to a file (if combined
 with   |
|            |              | Open)
        |
| Suspicious | Put          | May write to a file (if combined
 with   |
|            |              | Open)
        |
+------------+--------------+--------------------------------
--------+
```

Extracting all the hex key from above (58354F2150254041505B345C505A58353428505E29374329377D2425494341522D5354424E444152442D414E544956495255532D544553542D46494C452124482B482A) and converting into ascii text, the result obtained is

X5O!P%@AP[4\PZX54(P^)7C)7}$%ICAR-STBNDARD-ANTIVIRUS-TEST-FILE!$H+H*

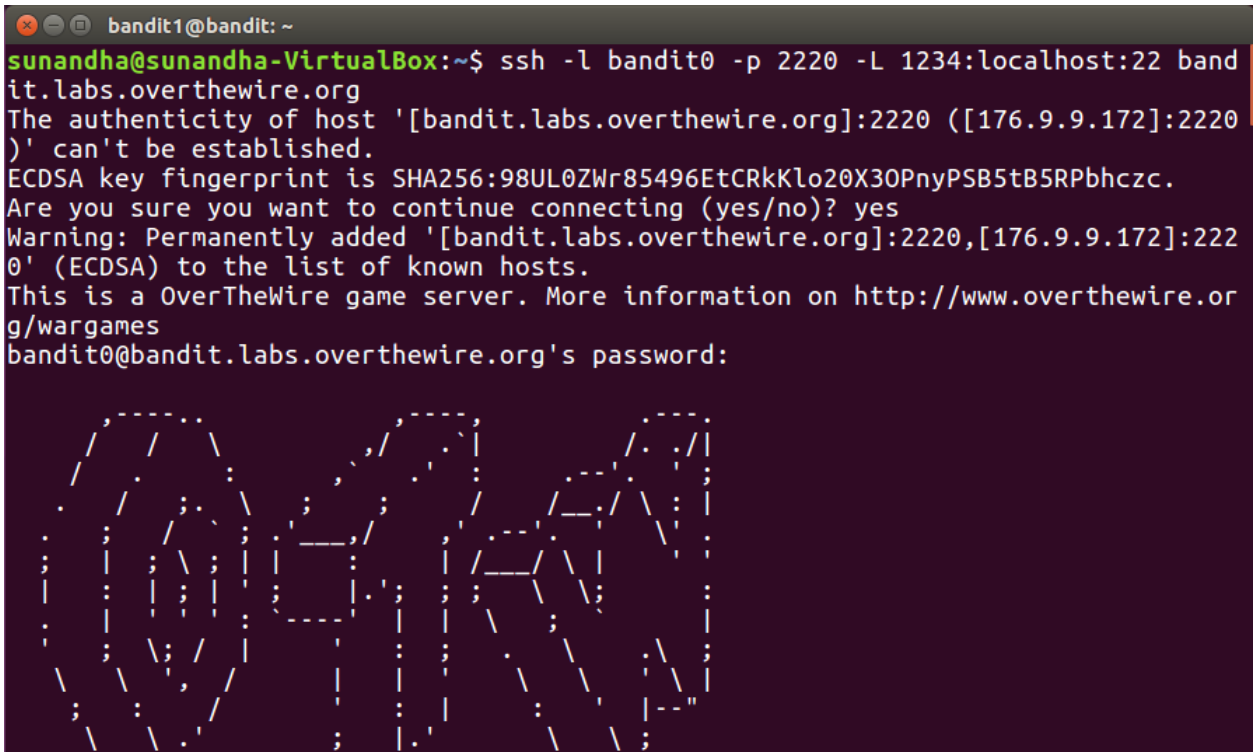
The EICAR Anti-Virus Test File is computer file to test the response of computer antivirus programs. This test file allows people to test anti-virus software without having to use a real computer virus. The file is a text file of between 68 and 128 bytes. This is a legitimate executable file called COM file that can be run by MS-DOS. The EICAR test string reads:

X5O!P%@AP[4\PZX54(P^)7C)7}$%ICAR-STBNDARD-ANTIVIRUS-TEST-FILE!$H+H*

the result that we obtained above after converting the hex key.

2. Bandit levels

Level 0



Level 0-1

The password for the next level is stored in a file called **readme** located in the home directory.

```
bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ cat readme
boJ9jbbUNNfktd78OOpsqOltutMc3MY1
bandit0@bandit:~$ ^C
bandit0@bandit:~$ ssh bandit1@bandit.labs.overthewire.org
ssh: Could not resolve hostname bandit.labs.overthewire.org: Temporary failure in name resolution
bandit0@bandit:~$ ssh -l bandit0 -p 2220 -L 1234:localhost:22 bandit.labs.overthewire.org
ssh: Could not resolve hostname bandit.labs.overthewire.org: Temporary failure in name resolution
bandit0@bandit:~$ ssh -l bandit1 -p 2220 -L 1234:localhost:22 bandit.labs.overthewire.org
ssh: Could not resolve hostname bandit.labs.overthewire.org: Temporary failure in name resolution
bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
sunandha@sunandha-VirtualBox:~$ ssh bandit1@bandit.labs.overthewire.org
^[[A
^C
sunandha@sunandha-VirtualBox:~$ ssh -l bandit1 -p 2220 -L 1234:localhost:22 bandit.labs.overthewire.org
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames
bandit1@bandit.labs.overthewire.org's password:

      ,------.            ,----,          .---.
     /   /   \         ,/   .`|         /. ./|
    /   .     :      ,`    .'  :     .--'.  ' ;
   .   /   ;.  \   ;    ;     /    /__./ \ : |
  .   ;   /  ` ; .'___,/    ,' .--'.     '   \' .
  ;   |  ; \ ; | |    :    | /___/ \ |    ' '
  |   :  | ; | ' ;    |.';  ; ;   \  \;      :
  .   |  ' ' ' : `----'  |  |  \   ;  `      |
  '   ;  \; /  |      '   :  ;   .   \    .\  ;
   \   \  ',  /       |   |  '    \   \   ' \ |
    ;   :    /        '   :  |     :   '  |--"
     \   \ .'         ;   |.'       \   \ ;
  www. `---` ver      '---' he       '---" ire.org
```

Level 1-2

The password for the next level is stored in a file called - located in the home directory.

```
bandit1@bandit:~$ ls -a
-  .  ..  .bash_logout  .bashrc  .profile
bandit1@bandit:~$ cat ./-
CV1DtqXWVFXTvM2F0k09SHz0YwRINYA9
bandit1@bandit:~$
```

Level 2-3

The password for the next level is stored in a file called **spaces** in this filename located in the home directory.

```
bandit3@bandit: ~

bandit2@bandit:~$ dir
spaces\ in\ this\ filename
bandit2@bandit:~$ cat spaces\in\this\filename
cat: spacesinthisfilename: No such file or directory
bandit2@bandit:~$ cat spaces\ in\ this\ filename
UmHadQclWmgdLOKQ3YNgjWxGoRMb5luK
bandit2@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
sunandha@sunandha-VirtualBox:~$ ssh -l bandit3 -p 2220 -L 1234:localhost:22 band
it.labs.overthewire.org
This is a OverTheWire game server. More information on http://www.overthewire.or
g/wargames
bandit3@bandit.labs.overthewire.org's password:
```

Level 3-4

The password for the next level is stored in a hidden file in the **inhere** directory.

```
bandit4@bandit: ~

bandit3@bandit:~$ ls
inhere
bandit3@bandit:~$ cd inhere
bandit3@bandit:~/inhere$ ls -a
.  ..  .hidden
bandit3@bandit:~/inhere$ cat .hidden
pIwrPrtPN36QITSp3EQaw936yaFoFgAB
bandit3@bandit:~/inhere$ exit
logout
Connection to bandit.labs.overthewire.org closed.
sunandha@sunandha-VirtualBox:~$ ssh -l bandit4 -p 2220 -L 1234:localhost:22 band
it.labs.overthewire.org
This is a OverTheWire game server. More information on http://www.overthewire.or
g/wargames
bandit4@bandit.labs.overthewire.org's password:
```

Level 4-5

The password for the next level is stored in the only human-readable file in the **inhere** directory.

```
bandit5@bandit: ~
bandit4@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  .profile  inhere
bandit4@bandit:~$ cd inhere
bandit4@bandit:~/inhere$ ls -a
-file00  -file02  -file04  -file06  -file08  .
-file01  -file03  -file05  -file07  -file09  ..
bandit4@bandit:~/inhere$ file ./-*
./-file00: data
./-file01: data
./-file02: data
./-file03: data
./-file04: data
./-file05: data
./-file06: data
./-file07: ASCII text
./-file08: data
./-file09: data
bandit4@bandit:~/inhere$ cat ./-file07
koReBOKuIDDepwhWk7jZC0RTdopnAYKh
bandit4@bandit:~/inhere$ ^C
bandit4@bandit:~/inhere$ exit
logout
Connection to bandit.labs.overthewire.org closed.
sunandha@sunandha-VirtualBox:~$ ssh -l bandit5 -p 2220 -L 1234:localhost:22 band
it.labs.overthewire.org
This is a OverTheWire game server. More information on http://www.overthewire.or
g/wargames
bandit5@bandit.labs.overthewire.org's password:
```

Level 5-6

The password for the next level is stored in a file somewhere under the **inhere** directory and is human-readable - 1033 bytes in size - not executable.

```
sunandha@sunandha-VirtualBox: ~
bandit5@bandit:~$ ls
inhere
bandit5@bandit:~$ cd inhere
bandit5@bandit:~/inhere$ ls -a
.            maybehere02  maybehere06  maybehere10  maybehere14  maybehere18
..           maybehere03  maybehere07  maybehere11  maybehere15  maybehere19
maybehere00  maybehere04  maybehere08  maybehere12  maybehere16
maybehere01  maybehere05  maybehere09  maybehere13  maybehere17
bandit5@bandit:~/inhere$ find -typef -size 1033c
find: unknown predicate `-typef'
bandit5@bandit:~/inhere$ find -type f -size 1033c
./maybehere07/.file2
bandit5@bandit:~/inhere$ cat ./maybehere07/.file2
DXjZPULLxYr17uwoI01bNLQbtFemEgo7
```

Level 6-7

The password for the next level is stored **somewhere on the server** and is owned by user bandit7 - owned by group bandit6 - 33 bytes in size.

```
bandit6@bandit:~$ find / -user bandit7 -group bandit6 -size 33c -type f 2>/dev/n
ull
/var/lib/dpkg/info/bandit7.password
bandit6@bandit:~$ cat /var/lib/dpkg/info/bandit7.password
HKBPTKQnIay4Fw76bEy8PVxKEDQRKTzs
bandit6@bandit:~$
```

Level 7-8

The password for the next level is stored in the file **data.txt** next to the word **millionth.**

```
bandit7@bandit: ~
bandit7@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  .profile  data.txt
bandit7@bandit:~$ awk '/^millionth/ {print $2;}' data.txt
cvX2JJa4CFALtqS87jk27qwqGhBM9plV
bandit7@bandit:~$
```

Level 8-9

The password for the next level is stored in the file **data.txt** and is the only line of text that occurs only once.

```
bandit8@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  .profile  data.txt
bandit8@bandit:~$ cat data.txt | sort | uniq -u
UsvVyFSfZZWbi6wgC7dAFyFuR6jQQUhR
bandit8@bandit:~$
```

Level 9-10

The password for the next level is stored in the file **data.txt** in one of the few human-readable strings, beginning with several '=' characters.

```
bandit9@bandit: ~
bandit9@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  .profile  data.txt
bandit9@bandit:~$ strings data.txt | grep "="
nfZ=
U=R*q
=-VW+
========== theP`
        =uN
\<P5J7=^
========== password
L='.
L========== isA
G&eB_=
9T=8?
9=!/"
========== truKLdjsbJ5g7yyJ2X2R0o3a5HQJFuLk
bandit9@bandit:~$
```

Level 10-11

The password for the next level is stored in the file **data.txt** which contains base64 encoded data.

```
bandit10@bandit: ~
bandit10@bandit:~$ ls -a
.  ..  .bash_logout  .bashrc  .profile  data.txt
bandit10@bandit:~$ cat data.txt
VGhlIHBhc3N3b3JkIGlzIElGdWt3S0dzRlc4TU9xM0lSRnFyeEUxaHhUTkViVVBSCg==
bandit10@bandit:~$ echo VGhlIHBhc3N3b3JkIGlzIElGdWt3S0dzRlc4TU9xM0lSRnFyeEUxaHhU
TkViVVBSCg== | base64 --dec
The password is IFukwKGsFW8MOq3IRFqrxE1hxTNEbUPR
bandit10@bandit:~$
```

Level 11-12

tr is use to translate or delete characters. Usage tr [OPTION] … SET1 [SET2]. pipe the output of data.txt into tr.

```
bandit11@bandit: ~
bandit11@bandit:~$ ls
data.txt
bandit11@bandit:~$ cat data.txt | tr a-zA-Z n-za-mN-ZA-M
The password is 5Te8Y4drgCRfCx8ugdwuEX8KFC6k2EUu
bandit11@bandit:~$
```

Level 12-13

Move the file to a new directory in the **/tmp** folder under your name. Perform a reverse hashdump using xxd command. Write the output to a file. Identify what type of file is this, using the **file** command. Rename it to that particular file format, using the **mv** command to change its file type. Decompress/unzip the files using the correct type of tool. Repeat the above process until you have the file.

```
bandit12@bandit:~$ ls
data.txt
bandit12@bandit:~$ mkdir /tmp/sunandha
bandit12@bandit:~$ cp data.txt /tmp/sunandha
bandit12@bandit:~$ cd /tmp/sunandha
bandit12@bandit:/tmp/sunandha$ xxd -r data.txt > sunfile
bandit12@bandit:/tmp/sunandha$ file sunfile
sunfile: gzip compressed data, was "data2.bin", last modified: Thu Dec 28 13:34:36 2017, max compression, from Unix
bandit12@bandit:/tmp/sunandha$ mv sunfile sunfile.gz
bandit12@bandit:/tmp/sunandha$ gzip -d sunfile.gz
bandit12@bandit:/tmp/sunandha$ file sunfile
sunfile: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/sunandha$ mv sunfile sunfile.bz2
bandit12@bandit:/tmp/sunandha$ bzip2 -d sunfile.bz2
bandit12@bandit:/tmp/sunandha$ file sunfile
sunfile: gzip compressed data, was "data4.bin", last modified: Thu Dec 28 13:34:36 2017, max compression, from Unix
bandit12@bandit:/tmp/sunandha$ mv sunfile sunfile.gz
bandit12@bandit:/tmp/sunandha$ gzip -d sunfile.gz
bandit12@bandit:/tmp/sunandha$ file sunfile
sunfile: POSIX tar archive (GNU)
bandit12@bandit:/tmp/sunandha$ mv sunfile sunfile.tar
bandit12@bandit:/tmp/sunandha$ tar xvf sunfile.tar
data5.bin
bandit12@bandit:/tmp/sunandha$ file data5.bin
data5.bin: POSIX tar archive (GNU)
bandit12@bandit:/tmp/sunandha$ mv data5.bin data5.tar
bandit12@bandit:/tmp/sunandha$ tar xvf data5.tar
data6.bin
bandit12@bandit:/tmp/sunandha$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/sunandha$ mv data6 data6.tar
mv: cannot stat 'data6': No such file or directory
bandit12@bandit:/tmp/sunandha$ mv data6.bin data6.tar
bandit12@bandit:/tmp/sunandha$ tar xvf data6.tar
data8.bin
bandit12@bandit:/tmp/sunandha$ file data8.bin
data8.bin: gzip compressed data, was "data9.bin", last modified: Thu Dec 28 13:34:36 2017, max compression, from Unix
bandit12@bandit:/tmp/sunandha$ mv data8.bin data8.gz
bandit12@bandit:/tmp/sunandha$ gzip -d data8.gz
bandit12@bandit:/tmp/sunandha$ file data8
data8: ASCII text
bandit12@bandit:/tmp/sunandha$ cat data8
The password is 8ZjyCRiBWFYkneahHwxCv3wb2a1ORpYL
bandit12@bandit:/tmp/sunandha$
```

Level 13-14

localhost is a hostname that refer to the machine that we are working on. So we must login as bandit14 using this private key. After log in as bandit14, all need to do is look at the content of the file it mentioned to find the password for the next level.

```
bandit13@bandit:~$ ls
sshkey.private
bandit13@bandit:~$ cat sshkey.private
-----BEGIN RSA PRIVATE KEY-----
```

```
bandit13@bandit:~$ ssh -i ./sshkey.private bandit14@localhost
```

```
bandit14@bandit:~$ cat /etc/bandit_pass/bandit14
4wcYUJFw0k0XLShlDzztnTBHiqxU3b3e
```

Level 14-15

nc - netcat utility is used for just about anything under the sun involving TCP, UDP. We send our password by piping it to a command that will allow us to connect to a local port. This generates the password for next level.

```
bandit14@bandit:~$ nc localhost 30000 < /etc/bandit_pass/bandit14
Correct!
BfMYroe26WYalil77FoDi9qh59eK5xNr

bandit14@bandit:~$
```

Level 15-16

the s_client command implement a generic SSL/TLS client. -quiet option is for no s_client output. we pass the password file to the correct host and port and return the next password.

```
bandit15@bandit:~$ openssl s_client -connect localhost:30001 -quiet < /etc/bandi
t_pass/bandit15
depth=0 CN = bandit
verify error:num=18:self signed certificate
verify return:1
depth=0 CN = bandit
verify return:1
Correct!
cluFn7wTiGryunymYOu4RcffSxQluehd
```

Level 16-17

The credentials for the next level can be retrieved by submitting the password of the current level to a port on localhost in the range 31000 to 32000. Do a port scan to identify the open ports between the range of 31000 to 32000. 2 ports produced error output because they are configured to restrict connectivity to SSL only. Then connect using openssl with s_client option, to check if there is any correct output. The private key for accessing the next level is stored in

port 31790.Copy this key to sshkey file inside /tmp/key_floder. modify the file permissions before using it.

```
bandit17@bandit: ~
31790/tcp open  unknown
31960/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
bandit16@bandit:~$ cat /etc/bandit_pass/bandit16 | openssl s_client -connect loc
alhost:31790 -quiet > /tmp/key/b16pkey
-bash: /tmp/key/b16pkey: No such file or directory
bandit16@bandit:~$ clear

bandit16@bandit:~$ nmap localhost -p31000-32000

Starting Nmap 7.01 ( https://nmap.org ) at 2018-02-19 20:19 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00019s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 996 closed ports
PORT      STATE SERVICE
31046/tcp open  unknown
31518/tcp open  unknown
31691/tcp open  unknown
31790/tcp open  unknown
31960/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds
bandit16@bandit:~$ echo test | nc -v localhost 31046
nc: connect to localhost port 31046 (tcp) failed: Connection refused
Connection to localhost 31046 port [tcp/*] succeeded!
test
bandit16@bandit:~$ echo test | nc -v localhost 31518
nc: connect to localhost port 31518 (tcp) failed: Connection refused
Connection to localhost 31518 port [tcp/*] succeeded!
bandit16@bandit:~$ echo test | nc -v localhost 31691
nc: connect to localhost port 31691 (tcp) failed: Connection refused
Connection to localhost 31691 port [tcp/*] succeeded!
test
bandit16@bandit:~$ echo test | nc -v localhost 31790
nc: connect to localhost port 31790 (tcp) failed: Connection refused
Connection to localhost 31790 port [tcp/*] succeeded!
bandit16@bandit:~$ echo test | nc -v localhost 31960
nc: connect to localhost port 31960 (tcp) failed: Connection refused
Connection to localhost 31960 port [tcp/*] succeeded!
test
bandit16@bandit:~$ echo cluFn7wTiGryunymYOu4RcffSxQluehd | openssl s_client -qui
et -connect localhost:31790
depth=0 CN = bandit
verify error:num=18:self signed certificate
verify return:1
depth=0 CN = bandit
verify return:1
Correct!
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ
```

```
bandit16@bandit:/$ mkdir -p /tmp/key/
bandit16@bandit:/$ cd /tmp/key/
bandit16@bandit:/tmp/key$ touch sshkey.private
bandit16@bandit:/tmp/key$ vim sshkey.private
bandit16@bandit:/tmp/key$ ssh -i ./sshkey.private bandit17@localhost
Could not create directory '/home/bandit16/.ssh'.
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/bandit16/.ssh/known_hos
ts).
This is a OverTheWire game server. More information on http://www.overthewire.or
g/wargames
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for './sshkey.private' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "./sshkey.private": bad permissions
bandit17@localhost's password:

[2]+  Stopped                 ssh -i ./sshkey.private bandit17@localhost
bandit16@bandit:/tmp/key$ chmod 400 sshkey.private
bandit16@bandit:/tmp/key$ ls -l
total 4
-r-------- 1 bandit16 bandit16 1675 Feb 19 20:29 sshkey.private
bandit16@bandit:/tmp/key$ ssh -i sshkey.private bandit17@localhost
Could not create directory '/home/bandit16/.ssh'.
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/bandit16/.ssh/known_hos
ts).
This is a OverTheWire game server. More information on http://www.overthewire.or
g/wargames

      ,-----.           ,----,          .----.
     /   /   \         ,/   .`|         /. ./|
    /   .     :      ,`    .'  :     .--'.  ' ;
  .   /   ;.  \   ;    ;     /    /__./ \ : |
  Enjoy your stay!

bandit17@bandit:~$ cat /etc/bandit_pass/bandit17
xLYVMN9WE5zQ5vHacb0sZEVqbrp7nBTn
bandit17@bandit:~$
```

Level 17-18

diff will output all the lines that are different between two files. The top line is from passwords.new and it is the next password. -f will read one or more newline separated pattern from file. -v will invert the match.

```
bandit17@bandit:~$ ls
passwords.new  passwords.old
bandit17@bandit:~$ diff passwords.new passwords.old
42c42
< kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd
---
> 6vcSC74ROI95NqkKaeEC2ABVMDX9TyUr
bandit17@bandit:~$ grep -vf passwords.old passwords.new
kfBf3eYk5BPBRzwjqutbbfE887SVc5Yd
bandit17@bandit:~$
```

Level 18-19

I tried to log in but I was logged out. This is because .bashrc is sourced as soon as the shell is opened which in this case has been configured to exit the connection. This actually implies the pseudo-terminal(pty) is closed. ssh uses a pseudo-terminal(pty) and not a text-terminal(tty). $cat readme gives us the password for the next level.

```
bandit17@bandit:~$ ssh -t bandit18@localhost /bin/sh
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/bandit17/.ssh/known_hosts).
This is a OverTheWire game server. More information on http://www.overthewire.org/wargames
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for '/home/bandit17/.ssh/id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/home/bandit17/.ssh/id_rsa": bad permissions
bandit18@localhost's password:
$ ls
readme
$ cat readme
IueksS7Ubh8G3DCwVzrTd8rAVOwq3M5x
$
```

Level 19-20

The red highlight signifies that the file has elevated permissions and any commands executed through the runtime of the file will be run as bandit20. the owner of the bandit20-do is bandit20.

```
bandit19@bandit:~$ ls
bandit20-do
bandit19@bandit:~$ ll
total 28
drwxr-xr-x  2 root     root     4096 Dec 28 14:34 ./
drwxr-xr-x 29 root     root     4096 Dec 28 14:34 ../
-rw-r--r--  1 root     root      220 Sep  1  2015 .bash_logout
-rw-r--r--  1 root     root     3771 Sep  1  2015 .bashrc
-rw-r--r--  1 root     root      655 Jun 24  2016 .profile
-rwsr-x---  1 bandit20 bandit19 7408 Dec 28 14:34 bandit20-do*
bandit19@bandit:~$ ./bandit20-do
Run a command as another user.
  Example: ./bandit20-do id
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit20
GbKksEFF4yrVs6il55v6gwY5aVje5f0j
bandit19@bandit:~$
bandit19@bandit:~$
```

Level 20-21

This one has a program, when ran it connects to the given port, read a line from that connection, check if it matches the current password and if so, give us the next password. We can actually accomplish this with nc. The -l flag will make nc listen on a port. We'll have nc send the current password when read and the program should give us the right password. We'll also run the server in the background so we can run the given program. That number is just the process id (PID) of the running command in case we need to get at it. Connect to the port I used (9876). So suconnect connects to our running server at 9876, reads the password that we read from the password file, and sent back the next password.

```
bandit20@bandit: ~
PORT      STATE SERVICE
22/tcp    open  ssh
113/tcp   open  ident
30000/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
bandit20@bandit:~$ echo "GbKksEFF4yrVs6il55v6gwY5aVje5f0j" | no -l -p 60000
no: command not found
bandit20@bandit:~$ echo "GbKksEFF4yrVs6il55v6gwY5aVje5f0j" | nc -l 1337 &
[3] 26233
bandit20@bandit:~$ echo "GbKksEFF4yrVs6il55v6gwY5aVje5f0j" | nc -l 1337 & 3729
[4] 26661
nc: Address already in use
3729: command not found
[4]   Exit 1                  echo "GbKksEFF4yrVs6il55v6gwY5aVje5f0j" | nc -l 13
37
bandit20@bandit:~$ cat /etc/bandit_pass/bandit20 | nc -l localhost 9876 &
[4] 27397
bandit20@bandit:~$ [1] 22433
[1]: command not found
bandit20@bandit:~$ cat /etc/bandit_pass/bandit20 | nc -l localhost 9876 &
[5] 28192
bandit20@bandit:~$ gE269g2h3mw3pwgrj0Ha9Uoqen1c9DGr
```

```
bandit20@bandit: ~
  http://www.overthewire.org/wargames/

  For support, questions or comments, contact us through IRC on
  irc.overthewire.org #wargames.

  Enjoy your stay!

bandit20@bandit:~$ ./suconnect 32123
^Z
[1]+  Stopped                 ./suconnect 32123
bandit20@bandit:~$ ./suconnect 60000
ERROR: Can't connect
bandit20@bandit:~$ ./suconnect 60000
ERROR: Can't connect
bandit20@bandit:~$ ./suconnect 60000
ERROR: Can't connect
bandit20@bandit:~$ ./sucoonect 1337
-bash: ./sucoonect: No such file or directory
bandit20@bandit:~$ ./suconnect 9876
ERROR: Can't connect
bandit20@bandit:~$ ./suconnect 9876
Read: GbKksEFF4yrVs6il55v6gwY5aVje5f0j
Password matches, sending next password
bandit20@bandit:~$
```

Level 21-22

navigate to the **/etc/cron.d** directory and look for the files, you will noticed the cronjob_bandit22 files. When opened to view the content of the cronjob, it seems that the job basically triggers a script located at **/usr/bin/cronjob_bandit22.sh**

```
bandit21@bandit:~$ cd /usr/bin
bandit21@bandit:/usr/bin$ cat cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
bandit21@bandit:/usr/bin$ cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
Yk7owGAcWjwMVRwrTesJEwB7WVOiILLI
bandit21@bandit:/usr/bin$
```

Level 22-23

1. **cat /etc/cron.d/cronjob_bandit23**
We get the output to a file location that runs at **/usr/bin/cronjob_bandit23.sh**

2. **cat /usr/bin/cronjob_bandit23.sh**

3. **cat /tmp/$mytarget**
This time we get **/tmp/: Permission denied**
Hmm looks like the $ may have blocked our full file location because it only shows **/tmp/** in the output.

4. **/usr/bin/cronjob_bandit23.sh**
We get the output: **Copying passwordfile /etc/bandit_pass/bandit22 to /tmp/8169b67bd894ddbb4412f91573b38db3**

5. **cat /tmp/8169b67bd894ddbb4412f91573b38db3**
We get our current password output. **Yk7owGAcWjwMVRwrTesJEwB7WVOiILLI**

6. so if we change from bandit22 to bandit23, we will get a file that have the password for bandit23. The long file name is a hash (md5) from mytarget. Let execute that same line but switch $myname to bandit23. We got another long string and looking at the content of this file in tmp folder gives us the next password.

```
bandit22@bandit:~$ ls -l /etc/cron.d/
total 16
-rw-r--r-- 1 root root 120 Dec 28 14:34 cronjob_bandit22
-rw-r--r-- 1 root root 122 Dec 28 14:34 cronjob_bandit23
-rw-r--r-- 1 root root 120 Dec 28 14:34 cronjob_bandit24
-rw-r--r-- 1 root root 190 Oct 31 13:21 popularity-contest
bandit22@bandit:~$ cat /etc/cron.d/cronjob_bandit23
@reboot bandit23 /usr/bin/cronjob_bandit23.sh  &> /dev/null
* * * * * bandit23 /usr/bin/cronjob_bandit23.sh  &> /dev/null
bandit22@bandit:~$ cat /usr/bin/cronjob_bandit23.sh
#!/bin/bash

myname=$(whoami)
mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)

echo "Copying passwordfile /etc/bandit_pass/$myname to /tmp/$mytarget"

cat /etc/bandit_pass/$myname > /tmp/$mytarget
bandit22@bandit:~$ cat /tmp/$mytarget
cat: /tmp/: Permission denied
bandit22@bandit:~$ /usr/bin/cronjob_bandit23.sh
Copying passwordfile /etc/bandit_pass/bandit22 to /tmp/8169b67bd894ddbb4412f9157
3b38db3
bandit22@bandit:~$ cat /tmp/8169b67bd894ddbb4412f91573b38db3
Yk7owGAcWjwMVRwrTesJEwB7WVOiILLI
bandit22@bandit:~$ echo I am user bandit23 | md5sum | cut -d ' ' -f 1
8ca319486bfbbc3663ea0fbe81326349
bandit22@bandit:~$ cat /tmp/8ca319486bfbbc3663ea0fbe81326349
jc1udXuA1tiHqjIsL8yaapX5XIAI6i0n
bandit22@bandit:~$
```

Level 23-24

This navigates to /var/spool/bandit24 directory and executes all scripts as bandit24. We need to make it write the content of /etc/bandit_pass/bandit24 somewhere in temporary file. Give the script and the folder the correct permissions.

```
bandit23@bandit:/etc/cron.d$ mkdir /tmp/sun1
bandit23@bandit:/etc/cron.d$ chmod 777 /tmp/sun1
bandit23@bandit:/etc/cron.d$ cd /tmp/sun1
bandit23@bandit:/tmp/sun1$ cat > sun.sh
#!/bin/bash
cat /etc/bandit_pass/bandit24 > /tmp/sun1/password
^C
bandit23@bandit:/tmp/sun1$ chmod 777 sun.sh
bandit23@bandit:/tmp/sun1$ cp sun.sh /var/spool/bandit24
bandit23@bandit:/tmp/sun1$ ls -al /var/spool/bandit24/
ls: cannot open directory '/var/spool/bandit24/': Permission denied
bandit23@bandit:/tmp/sun1$ ls
password  sun.sh
bandit23@bandit:/tmp/sun1$ cat password
UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ
bandit23@bandit:/tmp/sun1$
```

Level 24-25

There is no way to retrieve the pincode except by going through all of the 10000 combinaties, called brute-forcing. So create a bash script like before in \tmp\sun24\test.sh

Test.sh contains

```
for i in {1..10000}
do
  echo "UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ $i" >> ./f
done

# bash test.sh, you can see a new file named "f" created
# cat f | nc localhost 30002
```

```
Wrong! Please enter the correct pincode. Try again.
Correct!
The password of user bandit25 is uNG9O58gUE7snukf3bvZ0rxhtnjzSGzG

Exiting.
bandit24@bandit:/tmp/sun24$
```

Level 26-27

ssh to logged in bandit26, then it looked it logged out automatically.
1. before exec the command above the size of terminal should be small enough(util you can see 5 line)
2. ssh to localhost
3. type 'v' to enter vi
4. type ':r /etc/bandit_pass/bandit26' to show the password on screen.

```
bandit25@bandit:~$ cat /etc/passwd | grep bandit26
bandit26:x:11026:11026:bandit level 26:/home/bandit26:/usr/bin/showtext
bandit25@bandit:~$ cat /usr/bin/showtext
#!/bin/sh

export TERM=linux

more ~/text.txt
exit 0
bandit25@bandit:~$ ssh -i bandit26.sshkey bandit26@localhost
Could not create directory '/home/bandit25/.ssh'.
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:98UL0ZWr85496EtCRkKlo20X3OPnyPSB5tB5RPbhczc.
Are you sure you want to continue connecting (yes/no)? v
Please type 'yes' or 'no': yes
Failed to add the host to the list of known hosts (/home/bandit25/.ssh/known_hos
ts).
This is a OverTheWire game server. More information on http://www.overthewire.or
```

```
bandit25@bandit: ~

 _                     _ _ _   ___   __
5czgV9L3Xx8JPOyRbXh6lQbmIOWvPT6Z
 | |                   | (_) | |__ \ / /
 | |__   __ _ _ __   __| |_| |_   ) / /_
 | '_ \ / _` | '_ \ / _` | | __| / / '_ \
"/etc/bandit_pass/bandit26" [readonly] 1L, 33C                2,1           Top
```

Analyzing malicious pdf file

1. Report the number of objects

```
remnux@remnux: ~/Downloads
File  Edit  Tabs  Help
remnux@remnux:~/Downloads$ pdfid hvd.pdf
PDFiD 0.2.1 hvd.pdf
 PDF Header: %PDF-1.7
 obj                    6
 endobj                 6
 stream                 2
 endstream              2
 xref                   1
 trailer                1
 startxref              1
 /Page                  1
 /Encrypt               0
 /ObjStm                0
 /JS                    0
 /JavaScript            0
 /AA                    0
 /OpenAction            0
 /AcroForm              1
 /JBIG2Decode           0
 /RichMedia             0
remnux@remnu...   23:41
```

/Acroform elements might contain obfuscated javascript.

Pdfid shows that there are 6 objects in “hvd.pdf”.

2. Determine whether the file is compressed or not.

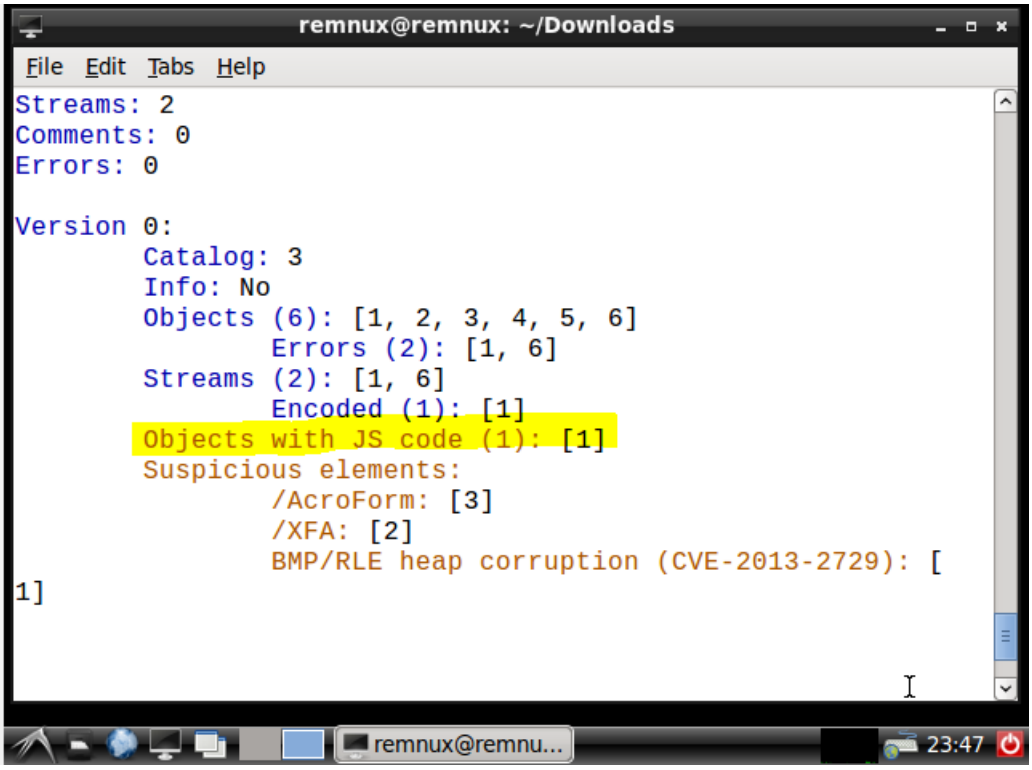
```
remnux@remnux:~/Downloads/hvd$ pdf-parser hvd.pdf
PDF Comment '%PDF-1.7\n'

PDF Comment '%\xc0\xff\xee\xfa\xba\xda\n'

obj 1 0
 Type:
 Referencing:
 Contains stream

  <<
    /Filter [ /Fl /Fl ]
    /L 544
  >>


obj 2 0
 Type:
 Referencing: 1 0 R

  <<
```

/Filter /F1 means the file is encoded with Flat Encode compression algorithm.

Object 2 contains /XFA elements referencing to object 1.

3. Determine whether file is obfuscated or not.

```
remnux@remnux: ~/Downloads
File  Edit  Tabs  Help
remnux@remnux:~/Downloads$ peepdf.py -f hvd.pdf




File: hvd.pdf
MD5: aaf8534120b88423f042b9d19f1c59ab
SHA1: ed0c7ab19d689554b5e112b3c45b68718908de4c
Size: 50717 bytes
Version: 1.7
Binary: True
Linearized: False
Encrypted: False
Updates: 0
Objects: 6
Streams: 2
Comments: 0
Errors: 0

Version 0:
```



Above picture shows there is javascript code with respect to object 1.

```
remnux@remnux:~/Downloads/hvd$ pdf-parser.py -c hvd.pdf --obje
ct 1 --filter --raw - object.raw
Usage: pdf-parser.py [options] pdf-file|zip-file|url
```

Object 1 when decompressed produces a 87 Mb file. This file contains 4 java script elements between <script> and 3 suspicious code between <image> tags. Code in between one of <image> tag is a base64 format code. I tried to decode the base64 data, it then produced 67Mb bmp file.

4. Extract the java script code into script.js file

Using peepdf, and js-code object_number , the java script code is extracted. This java script code is same as the 4 java script elements in the object.raw file.

```
remnux@remnux: ~/Downloads/hvd
File  Edit  Tabs  Help
Streams: 2
Comments: 0
Errors: 0

Version 0:
        Catalog: 3
        Info: No
        Objects (6): [1, 2, 3, 4, 5, 6]
                Errors (2): [1, 6]
        Streams (2): [1, 6]
                Encoded (1): [1]
        Objects with JS code (1): [1]
        Suspicious elements:
                /AcroForm: [3]
                /XFA: [2]
                BMP/RLE heap corruption (CVE-2013-2729): [1]


PPDF> js_code 1 > script.js
PPDF>
```

5. Deobfuscate java script

The js-didier tool, just like SpiderMonkey, will execute the java script code. But then it returned form2 variable is not defined. I tried to remove all the variables that were returned as undefined, but then it created me two log files. One log file contains just ‘unescape’.

```
remnux@remnux:~/Downloads/hvd$ js-didier script
script:62: ReferenceError: form2 is not defined
remnux@remnux:~/Downloads/hvd$
```